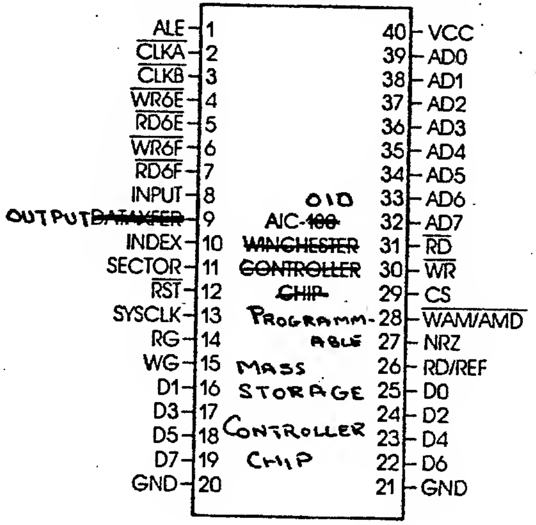
AIC-010-10
AIC-010-5



- * Controls ST506/412, ESDI, ST412HP, SA1000 and SMD, Interface Hard Disk Drives
- * Controls QIC-36 Interface Tape Drives
- * User Modifiable RAM Based Control Store
- * User Programmable Internal 32-bit ECC Polynomial
- * 5 Mbits/sec and 10 Mbits/sec Transfer Rate Versions
- * Soft or Hard Sector Drives
- * Multiple Sector Transfer
- * Sector Level Defect Handling
- * Non-interleaved Operation
- * User Programmable Sector Length upto a Full Track
- * High Speed Search Capability
- * Single +5V Power

DESCRIPTION

The Adaptec AIC-010 Programmable Mass Storage Controller is a LSI component that provides the major portion of hardware necessary to build a Winchester disk controller. The chip is capable of supporting most drive interfaces including, but not limited to, ST506/412, ESDI, ST412-HP, SA1000, and SMD. In addition, the chip can also be setup to control QIC-36 interface tape drives, and other such peripherals.

The Adaptec AIC-010 is a fully user programmable RAM based sequencer which can be loaded, in order to change its control store. The sequencer RAM can be easily modified to achieve full compatibility with a variety of drive interfaces. In addition, this allows the user to define specialized track formats. The chip also has a user programmable 32-bit ECC polynomial, thus offerring full flexibility in implenting error detection and correction.

The AIC-010 forms the nucleus of a three chip set comprised of the AIC-010, AIC-250 (or AIC-270) and the AIC-300. These chips along with a data seperator, driver/receivers, and a microprocessor chip provide all that is required to implement a full-featured, high performance disk controller. Typically, most implementations are able to read or write a full track in one revolution, run commonly available drives at their performance limits and, in fact, tend to put the performance bottleneck within the limitations of the system, as opposed to those of the controller.

The AIC-010 performs basic read/write functions for a mass storage device. For this purpose, the chip provides the necessary serialization/deserialization, formatting, ECC generation and correction functions. In addition, the AIC-010 also has search and verify capabilities.

The AIC-250 provides the write precompensation, write address mark/address mark detect and NRZ to MFM conversion functions required in ST506/412 type of drive interface applications. The AIC-270 has the ability to convert NRZ format data to/from 2/7 RLL code. This method of encoding can increase the effective capacity on a drive by upto 50%.

The AIC-300 provides a dual-ported buffer controller function in systems whose available bus bandwidth requires the use of a buffer between the host bus and the controller.

Figure 1 shows a simplified block diagram of an ST412/506 controller using the three chip set.

The AIC-010 is designed to work with either a local processor or the host processor. This choice is up to the designer and is a function of the host system's available bus bandwidth and board space design considerations. Accordingly the microcode for the control of the AIC-010 will be present in the system ram or a local (ep)rom.

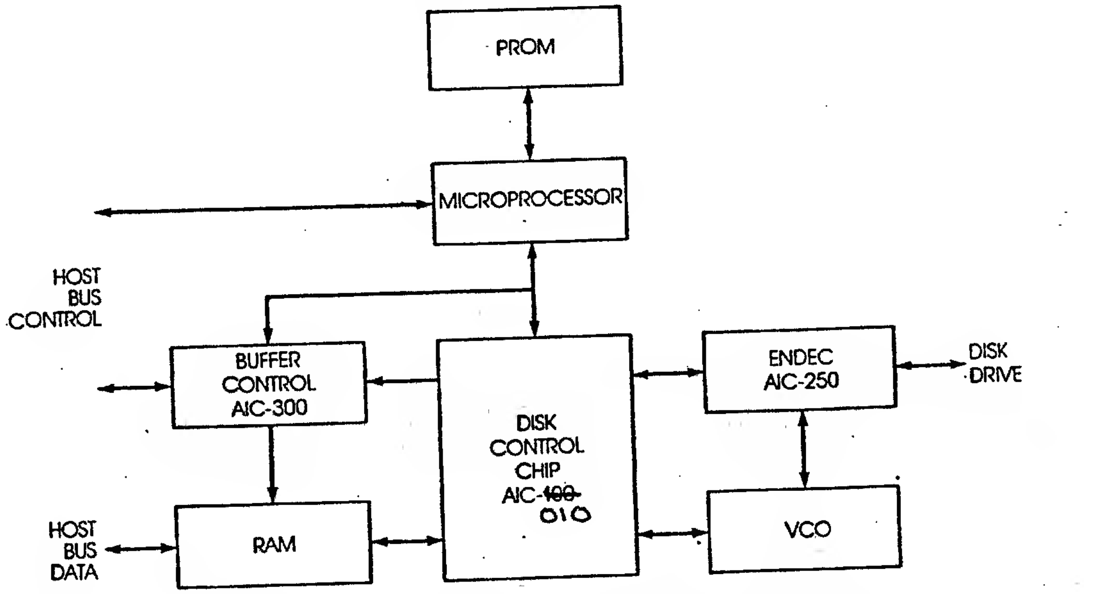


FIGURE 1. SIMPLIFIED WINCHESTER DISK CONTROLLER BLOCK DIAGRAM

AIC-010 PIN DESCRIPTION

SYMBOL	PIN	TYPE	NAME AND FUNCTION
ALE	1	IN	ADDRESS LATCH ENABLE: Signal used to latch the address from the multiplexed address/data bus.
$\overline{\text{CLKA}}$	2	OUT	CLOCK A: During a read or write operation, the output of this signal is derived from the input RD/REF CLK. Otherwise, it is derived from the input SYSCLK. The relationship between the input clock and CLKA is controlled by the contents of Register 7F.
$\overline{\text{CLKB}}$	3	OUT	CLOCK B: A pulse which overlaps the negative edge of Clock A, and occurs whenever a byte is transferred to/from Data Bus pins D0-D7.
GPIO0 or $\overline{\text{W6E}}$	4	IN/OUT	GENERAL PURPOSE I/O LINE 0: An user programmable I/O line for use as an input or an output. This pin can also be programmed as a decoded output for a write to address 6E.
GPIO1 or $\overline{\text{R6E}}$	5	IN/OUT	GENERAL PURPOSE I/O LINE 1: An user programmable I/O line for use as an input or an output. This pin can also be programmed as a decoded output for a read from address 6E.
GPIO2 or $\overline{\text{W6F}}$	6	IN/OUT	GENERAL PURPOSE I/O LINE 2: An user programmable I/O line for use as an input or an output. This pin can also be programmed as a decoded output for a write to address 6F.

GPIO3 or $\overline{R6F}$	7	IN/OUT	GENERAL PURPOSE I/O LINE 3: An user programmable I/O line for use as an input or an output. This pin can also be programmed as a decoded output for a read from address 6F.
INPUT	8	IN	INPUT PIN: The state of this pin is sampled by reading Register 7E, bit 4.
OUTPUT	9	OUT	OUTPUT PIN: Controlled by the bit 2 of the control block (AO thru B7) of the sequencer ram.
INDEX	10	IN	INDEX: Input for the index pulse received from the disk drive. Must be a minimum of one byte time.
SECTOR	11	IN	SECTOR: Input for the sector pulse received from drives that are hard-sectored. Must be a minimum of one byte time.
$\overline{RST}$	12	IN	RESET: A low input sets an internal reset latch that stops all operations within the chip and drops RG, WG, WAM & NRZ outputs. Registers 71 through 7E are reset.
SYSCLK	13	IN	SYSCLK: A 1.5 to 3.0 MHz clock input which is used to derive the Clock A output when not reading or writing data.
RG	14	OUT	READ GATE: Enables the external phaselock loop to lock onto the read data stream coming from the storage device
WG	15	OUT	WRITE GATE: Is used to enable or gate the writing of NRZ data out to the storage device.

DO-D7	16-19 22-25	IN/OUT	DATA BUS: Byte parallel data lines to/from the buffer.
GND	20-21		GROUND
RD/REF CLK	26	IN	READ REFERENCE CLOCK: A multiplexed input sourced from the VFO oscillator during read gate, otherwise from the write oscillator. This is the primary clock for the AIC-010.
NRZ	27	IN/OUT	NRZ: Read data input from the device when RG is active; write data to the device when WG is active.
$\overline{\text{WAM}}/\overline{\text{AMD}}$	28	IN/OUT	WRITE ADDRESS MARK/ADDRESS MARK DETECT: A one bit wide pulse is output when write gate is active and an address mark is to be written. When read gate is active, a low level input to indicate address mark detect.
CS	29	IN	CHIP SELECT: Active during processor bus cycles to/from the chip.
$\overline{\text{WR}}$	30	IN	WRITE: Signal from the microprocessor to latch data into a specified register.
$\overline{\text{RD}}$	31	IN	READ: Signal from the microprocessor to enable data from a specified register out onto the bus.
ADO-AD7	32-39	IN/OUT	Multiplexed address/data lines interfacing to the control processor.
VCC	40		+5 Volts.

FUNCTIONAL DESCRIPTION

Internal to the controller chip are three functional blocks:

* Microprocessor Interface Decoder
* Sector Format Sequencer
* Dataflow

MICROPROCESSOR INTERFACE DECODER: The microprocessor interface is an eight-bit multiplexed bus such as is found on the Intel 8085 family of processors. Other microprocessors such as the Z80 can be utilized by multiplexing their address and data lines, and generating the necessary control lines.

The controller chip decodes addresses from 6EH to FFH, and addresses 50H and 51H. While all addresses are not used, to prevent erroneous operation, addresses from 6EH to FFH must not be decoded elsewhere on the controller board. The device architecture is structured to allow the firmware of an processor to determine what functions are to be incorporated in the control unit design.

SECTOR FORMAT SEQUENCER: The sector format sequencer performs the basic sequencing function for a mass storage device which include:

* Read ID
* Read ID and Read Data
* Read ID and Write Data
* Write ID and Write Data

These functions can be modified to perform the search data and verify data functions.

The sequencer consists of 96 bytes of RAM, organised as a 24 x 4 matrix. These locations have to be set up at initialization time, for the proper operation of the chip. Under firmware control, the AIC-010 can be made to sequence through different types of operations. The user can control the timing relationships between various ouput signals, and can monitor the different input lines to branch to various sequencer locations.

The controller chip also has other registers which can be used to control the definition of the track format. Using these registers, such things like gap lengths, sync characters, ECC length can be controlled. The track layout (sector size and sector data fill character) can also be flexibly defined.

The Winchester controller chip interfaces with the bidirectional data bus which is connected to an external RAM buffer. The CS, WE, and address increment signals required for the sector buffer are derived from the Clock A and Clock B outputs.

DATA FLOW: The dataflow portion of the controller chip is composed of a 32 bit ECC and a serializer/deserializer. Data to be written to the disk enters the device in 8-bit parallel format. It is serialized, and run through a 32-bit ECC generator. The controller chip outputs NRZ serial data followed by 4bytes of ECC check burst.

The user has the ability to select the ECC polynomial that is optimum for the media and the encoding scheme that is being used. The ECC length can also be controlled, and in fact can be fully suppressed in the case of a tape controller. In this case, 2 bytes of CRC would have to be supplied externally.

Figure 2 is a block diagram of the AIC-010 controller chip, and identifies the different blocks.

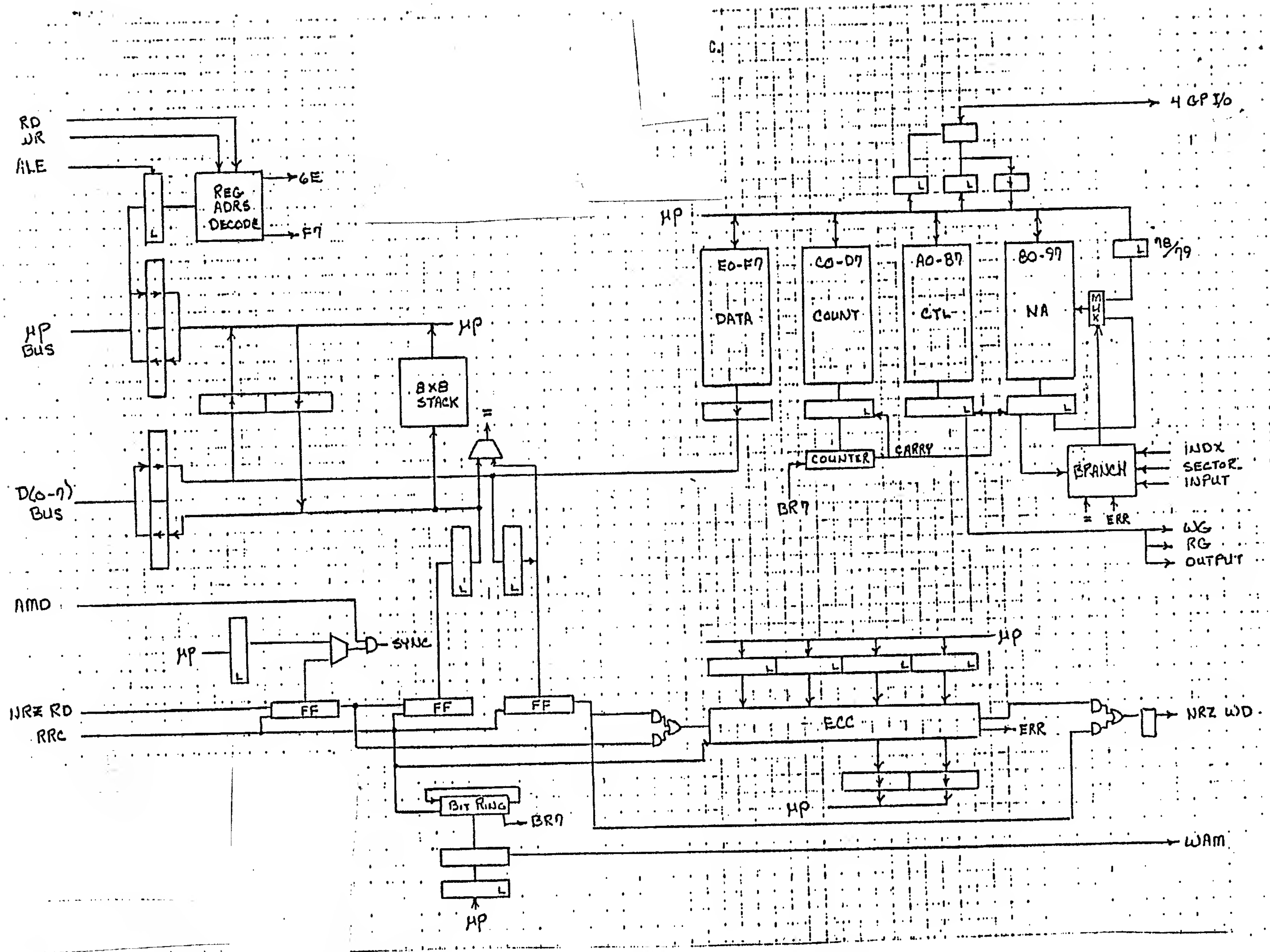


FIGURE 2. AIC-010 BLOCK DIAGRAM

FUNCTIONAL OPERATION

The Programmable Mass Storage controller chip is designed to be used with a low-cost micro processor rather than the high-speed-bit slice designs required for controllers in the past. This processor is used to maintain "loose" synchronization with the real time on the device. The Winchester controller chip in return maintains the "close" synchronization of data to and from the disk and provides the signals necessary to control this path. With this device, a lower total part count can be achieved with the same or greater performance than that of a bit slice processor design.

Because the controller chip controls primarily the high speed signals associated with the Winchester disk, the designer is free to choose which type of drive to interface, e.g., ST506, SMD, etc. Each of these interfaces can be accommodated with the 4 general purpose I/O lines. These lines can be defined as input, output or external register decode signals. The AIC-010 chip simplifies the external logic needed, by internally decoding addresses 6EH and 6FH, and providing signals to read and write ports at these locations.

An example of an ST412 application is shown in Figure 3. These 4 signals are used to read device status or write device control lines.

The basic operation of the controller chip are controlled by the contents of the sequencer ram. The sequencer ram consists of 96 bytes organized as 24 addresses which are 4 bytes wide. Each of these 4 bytes represent a field at that address, and is broken down as follows; Data, Count, Control and Next Address. The operation of the chip revolves around the branch register, Register 78; and the status/start register, Register 79. Register 79 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 register 78 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 conditions, which are based on external or internal events, the chip can be made to sequence throuvh different operations. An example of the use of the AIC-010 as a ST-506 controller is shown in the Appendix.

The controller chip also has a stack which is 8 bytes deep. During a read process, by enabling the stack, information read from the drive such as the ID field, can be pushed on to the stack. These can then be popped, to look for any relevant information.

During a read process from the device, the chip also has the ability to compare the data being received, on a byte for byte basis with information found in other locations. The controller chip can be set up to compare with information found in the data field in the sequencer ram (such as when looking for the ID field), or with information stored in the external buffer (such as during a data field search operation).

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 OP is started. By employing Registers 71, 72 and 73 the microprocessor can determine if the error is correctable, and if so, the error pattern and displacement from the beginning of the sector. The ECC polynomial is a computer selected code that will correct 8 bit single burst errors. After the error pattern is determined, it is EXORed with the data byte (bytes) in the RAM buffer.

CLKA and CLKB outputs are used to control the external RAM buffer address counter. CLKB should be interpreted as the beginning of a controller chip memory access with a Clock A period equal to the RAM access time. The D(0-7) pins will contain valid data during that time of the cycle when CLKA is high. This is shown in the reference timing diagram.

The table shows the register present in the AIC-010, which are used to control its operation. This is in addition to the sequencer ram which is located from address 80H to FFH. A more detailed graphical breakdown of the registers follows.

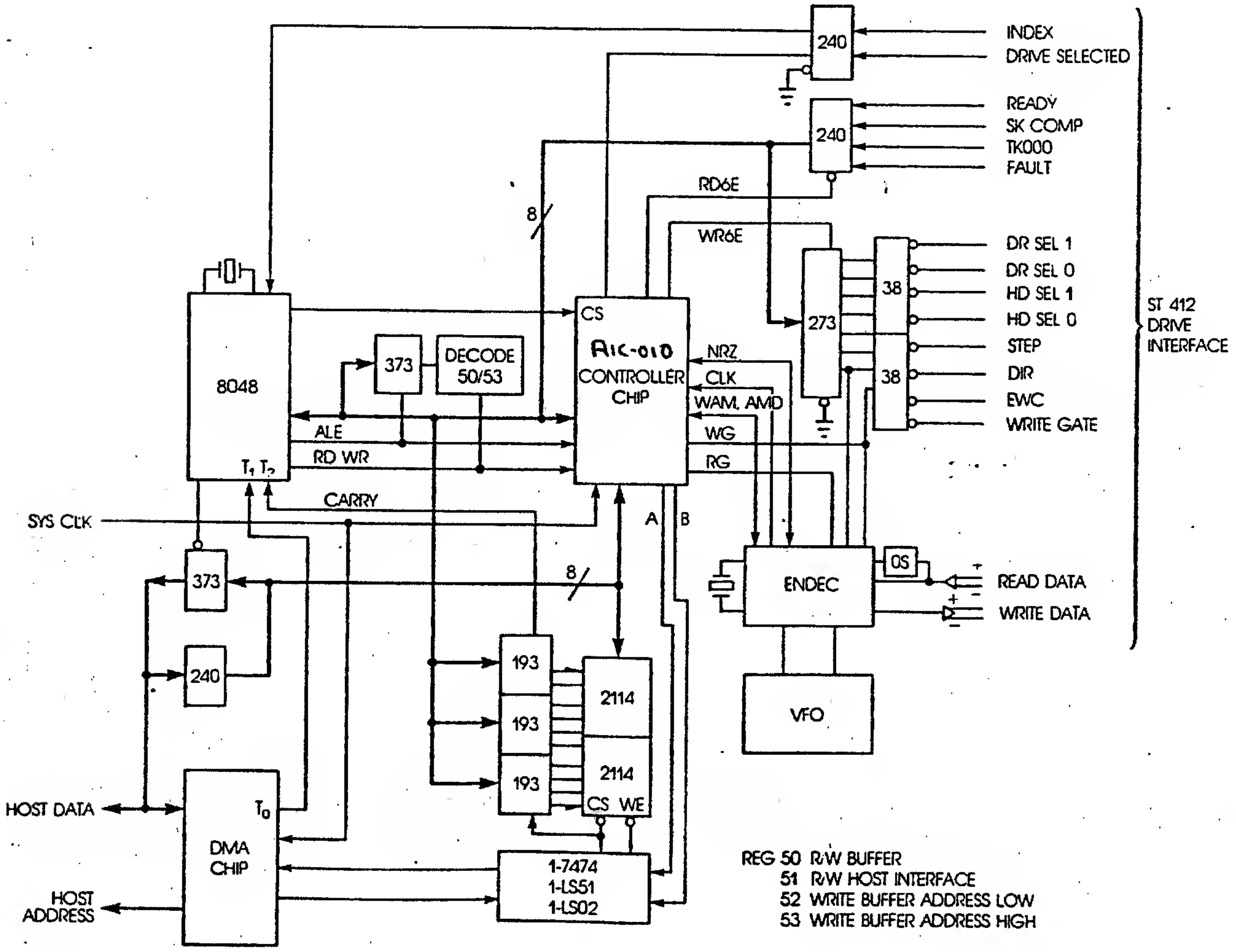


FIGURE 3. AIC-~~400~~ ST412 APPLICATION
010

AIC-010

~~SERDES~~ REGISTER SUMMARY

REV B 11/13/81

Ø	BUFFER DATA	71	ECC CTL	72.	ECC (16-23)	73	ECC (24-31)
/W	7	W	7 DISABL 8-15	R	7 ECC 23	R	7 ECC 31
	6		6 DISABL Ø-7		6 22		6 30
	5		5 CHIP RESET		5 21		5 29
	4 BUFFER		4 EN SECTOR BRCH		4 2Ø		4 28
	3		3 CLEAR ECC		3 19		3 27
	2		2 DISABL FEEDBACK		2 18		2 26
	1		1 SHIFT ECC		1 17		1 25
	Ø		Ø SERIAL ECC IN		Ø ECC Ø-16		Ø ECC 24

'4	POLY (Ø-7)	75	POLY (8-15)	76	POLY (16-23)	77	POLY (24-31)
W	7 POLY 8	W	7 POLY 16	W	7 POLY 24	W	7 —
	6 7		6 15		6 23		6 POLY 31
	5 6		5 14		5 22		5 3Ø
	4 5		4 13		4 21		4 29
	3 4		3 12		3 2Ø		3 28
	2 3		2 11		2 19		2 27
	1 2		1 1Ø		1 18		1 26
	Ø POLY 1		Ø POLY 9		Ø POLY 17		Ø POLY 25

'8	BRANCH/MA	79	W START ADR/	7A	OP CTL	7B	WAM CTL	
	7 —TEST BR7	R	7 AM ACTIVE	W/R	7 INHIBIT	W	7 WAM AT BR7	
					6 CARRY			
	6 —TEST ECC ERR		6 DATA XFER		6 —		6	6
	5 —TEST CARRY		5 BRCH ACTIVE	W/R	5 SUPRES XFER		5	5
I/R	4 BRCH/MA 4	W/R	4 STOPPED	W/R	4 SRCH OP		4	4
I/R	3 3	W/R	3 —		3 —		3	3
I/R	2 2	W/R	2 ECC ERR	R	3 NRZ DATA IN		2	2
I/R	1 1	W/R	1 COMPARE LOW	R	1 SECTOR PAST		1	1
I/R	Ø BRCH/MA Ø	W/R	Ø COMPARE EQUAL	R	Ø INDEX PAST		Ø WAM AT BR	Ø

'C	AMD CTL	7D	GP I/O CTL	7E	GP I/O	7F	CLK CTL/POP
W	7	W	7 ENAB RD R6F		7 —	R/W	7 / ØØ = SC/4
							}Ø1 = SC/2
	6		6 ENAB SET R6F		6 —		6 / 1Ø = SC
	5		5 ENAB RD R6E	R	5 OUT PUT		5 —
	4 AM		4 ENAB SET R6E	R	4 BRCH IN		4 / Ø = RRC/2 (5MHZ
	MATCH						/1 = RR4 4 (1ØMHZ)
	3		3 ENAB GP3 OUT	R/W	3 GP I/O 3		3 CLK A&B Hi Z
	2		2 ENAB GP2 OUT	R/W	2 2		2 / NRZ SHIFT
	1		1 ENAB GP1 OUT	R/W	1 1		1 } IN CODE
	Ø		Ø ENAB GPØ OUT	R/W	Ø GP I/O Ø		Ø /

SEQUENCE MEMORY BIT MAP

	EØ thru F7		CØ thru D7		AØ thru B7		8Ø thru 97				
	DATA		COUNT		CONTROL		NEXT ADRS.		ADRS. R79	BRANCH R78	COMMENTS
	7 6 5 4	3 2 1 Ø	7 6 5 4	3 2 1 Ø	7 6 5 4	3 2 1 Ø	7 6 5 4	3 2 1 Ø			
Ø									Ø		
1									1		
2									2		
3									3		
4									4		
5									5		
6									6		
7									7		
8									8		
9									9		
A									A		
B									B		
C									C		
D									D		
E									E		
F									F		
1Ø									1Ø		
11									11		
12									12		
13									13		
14									14		
15									15		
16									16		
17									17		
STOP 1F	7 6 5 4	3 2 1 Ø	7 6 5 4	3 2 1 Ø	7 6 5 4	3 2 1 Ø	7 6 5 4	3 2 1 Ø	1F STOP		
	DATA		DATA TYPE	FIELD CNT			N.A.				

DATA TYPE: AM, ECC, SEDCA

CONTROL: WG ON, RG ON, WG OFF, STACK ENABL, INVALID NRZ, OUTPUT, COMPARE EN, DATA XFER

RG - ECC = 1	RG - ECC = Ø
ØØØ - No Branch	ØØØ - No Branch
ØØ1 - ECC ERR Stop	ØØ1 - Stop on Input
Ø1Ø - No Compare Stop	Ø1Ø - Stop on Index or Sector
Ø11 - ERR or Comp. Stop	Ø11 - Stop on Not Equal
1ØØ - Good ECC & Comp.	1ØØ - Branch on Carry
1Ø1 - ECC ERR	1Ø1 - Branch on Input
11Ø - Not Equal	11Ø - Branch on Index or Sector
111 - ERR or Not Equal	111 - Branch on Not Equal

INTERNAL REGISTER DESCRIPTION

71 ECC CONTROL (WRITE ONLY)

7 6 5 4 3 2 1 0

SERIAL ECC INPUT: ECC bit 0 will be loaded with the contents of this bit when the shift control bit is set. Note that the Rd/Ref pin must be cycling,RG and WG must be inactive, and Feedback disabled.

ECC SHIFT CONTROL:Each time this bit is set a single shift pulse will be sent ot the 32-bit ECC register. This bit is automatically cleared after the shift pulse occurs

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

CLEAR ECC: While this bit is on the ECC shift register will be reset.

ENABLE SECTOR BRANCH: When set will cause the sector input to ORed with the index so that an operation may begin at index or sector.

CHIP RESET: When external RST occurs this bit will be set and a constant reset to the chip while it is set. The reset condition is cleared by setting this bit to a zero from the microprocessor.

POLYNOMIAL LENGTH: Controls the length of the ECC polynomial.

0 0 = 16 bit polynomial (16 - 31) enabled

0 1 = 24 bit polynomial (8 - 31) enabled

1 0 = Screwed up polynomial

1 1 = 32 bit polynomial (0 - 31) enabled

72 ECC (16 - 23) (READ ONLY)

7 6 5 4 3 2 1 0

This bit is an OR of ECC Bits 0 thru 16. Whenever one of those bits is set this bit will also be set. This OR is gated by the appropriate length selected.

ECC Bits 17 thru 23 - BIT 23 is in register Bit 7.

73 ECC (24 - 31) (READ ONLY)

7 6 5 4 3 2 1 0

ECC Bits 24 thru 31 - Bit 31 is in register Bit 7.

74 POLY (1 - 8) (WRITE ONLY)

7 6 5 4 3 2 1 0

Each bit correspondes to a feedback path being enabled. i.e., if Bit 1 in the registe is on, the output (ECC 31) will be XOR with data in and ECC 1 and become the input to ECC Bit 2.

75 POLY (9 - 16) (WRITE ONLY)

7 6 5 4 3 2 1 0

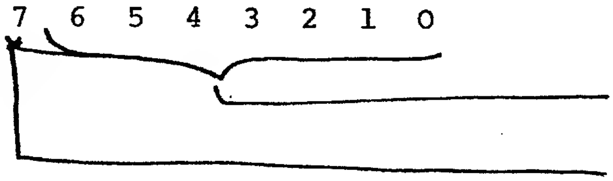
Same function for Bits 8 thru 15. Bit 7 enables feedback with Bit 15 of ECC.

76 POLY (17 - 24) (WRITE ONLY)

7 6 5 4 3 2 1 0

Same function for Bits 16 thru 23. Bit 7 enables feedback with Bit 23 of ECC.

77 POLY (25 - 31) (WRITE ONLY)

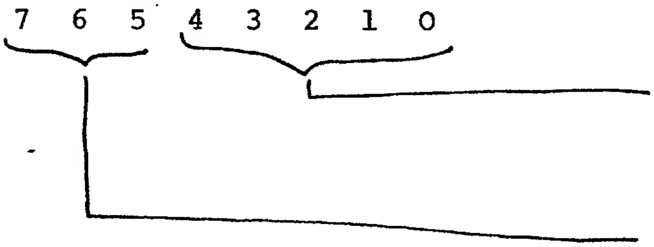


Same function for Bits 24 thru 30. Bit 6 enables feedback with Bit 30 of ECC.

Not used.

Note: Registers 74 thru 77 can not be reset.

78 BRANCH CONTROL (READ/WRITE)



BRANCH ADDRESS: Writing Bits 0 - 4 will cause the sequence RAM to jump to this address when a branch condition is met.

A read off this reg Bits 0 - 4 are the next address field. Bits 5, 6 and 7 are test points.

79 CONTROLLER STATUS (READ)   All bits reset by external reset.

7 6 5 4 3 2 1 0

COMPARE EQUAL: The state of the compare FF's as a result of all bytes, where comparison was enabled between the data buffer or micro RAM and the read data register at ECC time.

COMPARE LOW: Same as above except that the data buffer or micro RAM was greater than the read data register.

ECC ERROR:After the last bit off ECC data is read this bit is either set or reset depending on whether all bits in the ECC are zero.

NOT USED: will be zero.

STOPPED: The sequence RAM is at address 1F. The ECC contents have not been reset and RG & WG are reset. The bit ring is running.

BRANCH ACTIVE: This bit is set whenever a branch condition is met. The bit is reset by a read of this register.

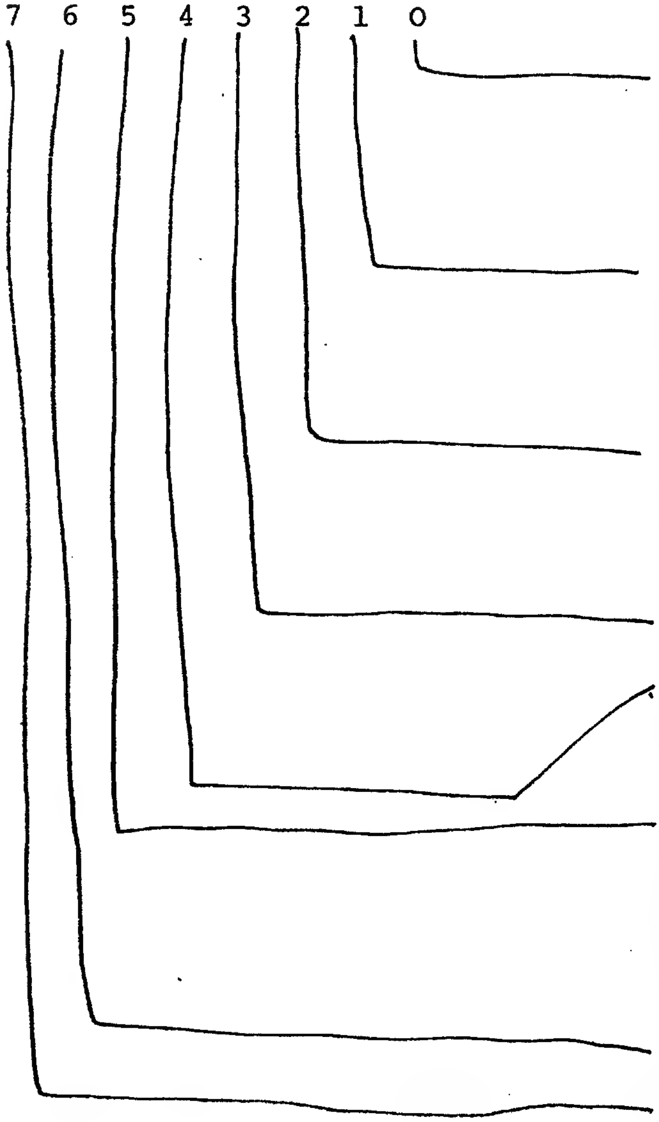
DATA TRANSFER:This bit is on whenever data is being transferred either to or from the buffer memory. i.e., it is the data XFER enable bit of the micro RAM.

AM ACTIVE: Is set by reading or writing an AM or SYNC byte and is reset by reading or writing the ECC bytes. The bit is also reset by a stopped condition.

79 SEQUENCER START (WRITE)

START ADDRESS:A write to bits 0-4 will start the sequencer at the appropriate address.

7A OPERATION CONTROL (READ/WRITE)



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 the bit.

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 the bit.

NRZ DATA IN: A input of "1" on the NRZ data pin while read gate was on, has occurred since the late time this register was accessed.

NOT USED: will be a one.

SEARCH OPERATION: This bit must be set whenever a data field compare is required.

SUPPRESS TRANSFER: Clk B will not be generated when this bit is on. Also during WG the data field will be written with the contents of the micro RAM data, or under RG compared with content of micro RAM data.

NOT USED.

INHIBIT DATA FIELD CARRY: When set the carry/load of the micro engine for the data field will be inhibited. After a carry has occurred this bit will bereset.

7B WAM CONTROL (WRITE ONLY)

7 6 5 4 3 2 1 0

WRITE ADDRESS MARK CONTROL:The WAM/AMD PIN will go active for each bit cell time corresponding to the bits set in this register during a write address mark operation.

7C AMD CONTROL (WRITE ONLY)

7 6 5 4 3 2 1 0

ADDRESS MARK DETECT CONTROL: A match between this register and the serial NRZ RD data input will cause a SYNC detect (if AMD input is active), the bit ring to start at zero, and data to be gated into the ECC. Only those bits enabled by reg 7F Bits 0 thru 7 can be set for comparison.

7D GP I/O CONTROL (WRITE ONLY)

7 6 5 4 3 2 1 0

GPIO DIRECTION CONTROL:When set these bits enable the corresponding bits of the GP I/O register to the output pins. When these bits are zero the pins are the source of the GP I/O register.

i.e., inputs are simply gated to the bus when a read of GP I/O is done.

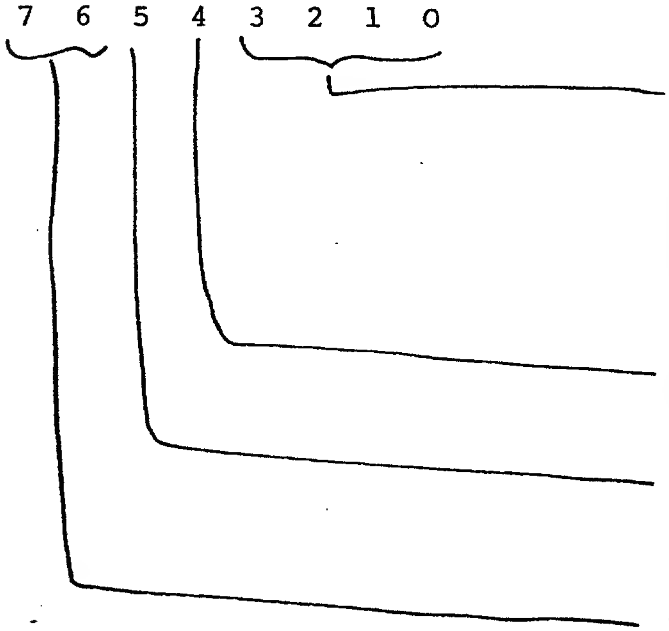
W6E CONTROL: When set along with Bit 0 this bit will disable GP I/O reg Bit 0 as an output and enable a set register 6E output pulse. When zero the GP I/O register is the output.

R6E CONTROL: Same function as above except a read pulse for register 6E will be output from pin 1.

W6F CONTROL: Same function as above except a write pulse for register 6F will be output from pin 2.

R6F CONTROL: Same function as above except a read pulse for register 6F will be output from pin 3.

7E GP I/O (READ/WRITE)



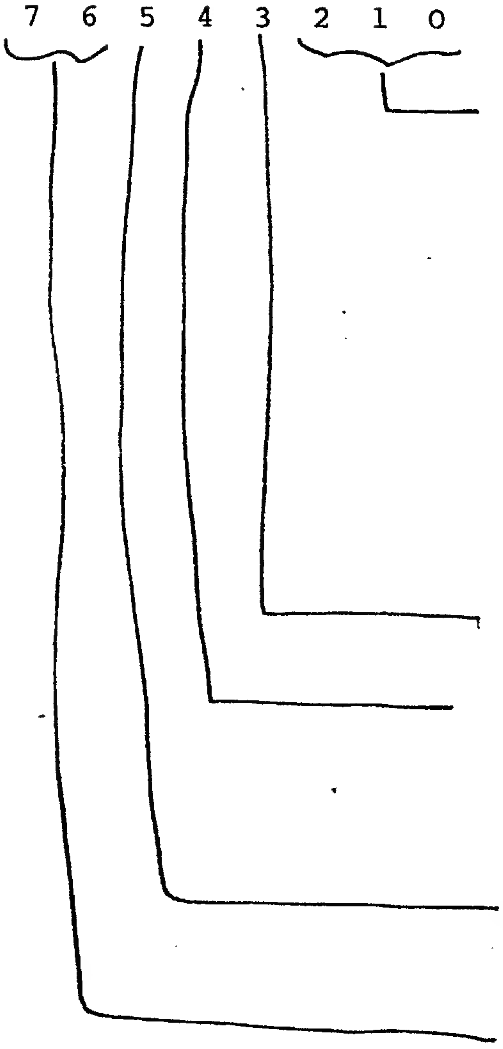
GPIO BITS 0-3: General purpose input/output Bits (0 - 3) are independently programable for direction of data on the 4 pins and each bit has a flip flop that holds data for output purposes only. A read of this register will gate the contents of the input pin.

BRANCH PIN:An external input that also is a branch condition for the micro engine.

OUTPUT PIN:This bit is a direct output of the micro RAM and is also bit in this register.

NOT USED

7F CLOCK CONTROL (WRITE)



0	0	0	NRZ Shift Reg Compare Starts At Bit 0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

When set Clk A & B outputs are high impedence.

0	Clk A during sequencer data XFER will be 2 cycles per byte.
1	Clk A during sequencer data XFER will be 4 cycles per byte.

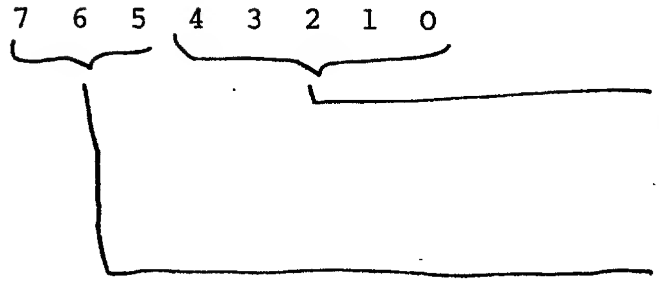
Not used

0	0	Clk A during non-Serdes XFER will be 1/4 of system clock period.
0	1	Clk A during non-Serdes XFER will be 1/2 of system clock period.
1	0	Clk A during non-Serdes XFER will be equal to system clock period.
1	1	Clk A will be messy.

7F STACK (READ)

STACK: A read of this reg will read the top of the 8 byte stack.

80 THRU 97 NEXT ADDRESS (READ/WRITE)



NEXT ADDRESS: This is the address the micro engine will go to after the down-counter has reached zero, if if a branch is not taken. There are 24 possible next address locations locations.(00 to 17)

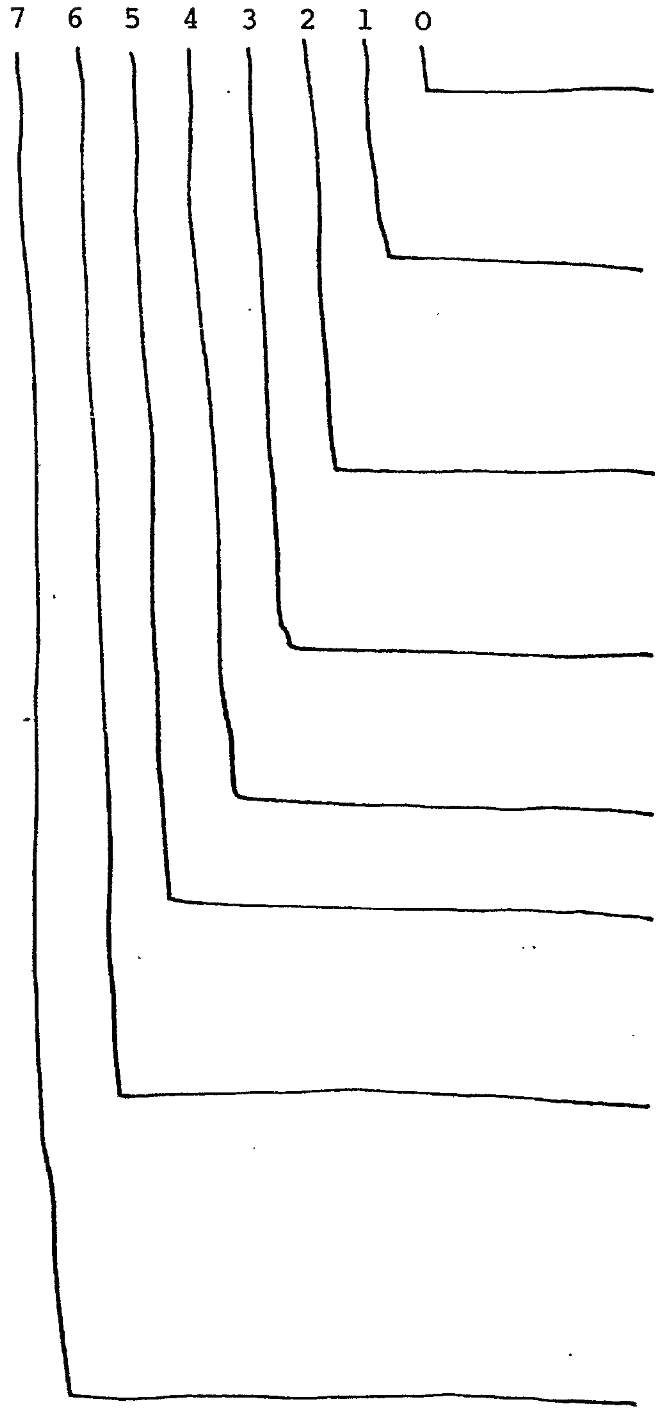
BRANCH CONDITIONS: Branch conditions when ECC and read gate are is active. These branches are taken at the end of ECC time.

0	0	0	Continue, next address used
0	0	1	Go to address 1F on ECC error
0	1	0	Go to address 1F on not compare equal
0	1	1	Go to address 1F on not compare equal or ECC error
1	0	0	Branch on good ECC and compare equal
1	0	1	Branch on ECC error
1	1	0	Branch on not compare equal
1	1	1	Branch on not compare equal or ECC error

Branch condition at all other times. These branches are taken immediately.

0	0	0	Continue, next address used
0	0	1	Go to address 1F on external input active
0	1	0	Go to address 1F on index or sector active
0	1	1	Go to address 1F on not compare equal
1	0	0	Branch on carry
1	0	1	Branch on external input active
1	1	0	Branch on index or sector active
1	1	1	Branch on not compare equal

AO THRU B7 CONTROL (READ/WRITE)



DATA TRANSFER:This bit when set will cause Clk B to be generated in SYNC with Clk A and data will be sourced or read on the data bus depending on RG or WG active respectively.

COMPARE ENABLE: When active along with RG will allow a comparison between read data and micro engine or data bus input (if SRCH enable is set).

OUTPUT: This bit is connected to output pin 9 and is used for external logic functions that must be synchronous with Read/Write functions.

INVALID NRZ CONTROL: When set with RG this bit will block the NRZ data input. This is used to allow VFO phase up.

STACK ENABLE: When on pushes read data onto 8 byte stack.

RESET WRITE GATE: WG latch will be cleared at carry and bit ring 4 for the field this bit is on for. Note that RG is always reset at the end of ECC.

SET READ GATE: RG latch will be set at the time this control is read from micro RAM. The latch will be reset at the end of ECC or when the micro engine goes to a stop. RG latch will not be set if WG is already on. The output of RG latch is connected to the output pin 14.

SET WRITE GATE: WG latch will be set at bit ring 4 time after this control will be reset at bit ring 4 after the ECC bytes (when NRZ control is off) or when the micro engine goes to a stop. WG latch will not set if RG is already on. The output of WG latch is connected to output pin 15.

CO THRU D7 COUNT (READ/WRITE)

$\underbrace{7\ \ 6\ \ 5}\ \underbrace{4\ \ 3\ \ 2\ \ 1\ \ 0}$

COUNT: These bits are the initial value of the micro engine counter when a new state is entered. Bits 0 thru 4 are set to bits 0 thru 4 of the counter respectively. The counter is down counted on bit ring 7 and when it reaches zero a new state will be accessed from thep micro RAM.

DATA TYPE: When data XFER bit of the micro RAM is off these bits are decoded for data type as indicated below:

0	0	0	Normal
0	0	1	AM
0	1	0	ECC
1	0	0	Set Enabl bit ring to Clk A (SEB (Reset at end of ECC)

Bits 5, 6 and 7 of the counter will be initialized to zero with data XFER Bitoff.

When the data XFER bit is set bits 5, 6 and 7 of the count will be initialized with bit 5, 6 and 7 respectively.

EO THRU F7 DATA (READ/WRITE)

7 6 5 4 3 2 1 0

DATA: This register is the source for all overhead bytes of data used by the device during write operations. During Read operations it is one of the operands to the comparison logic.

When data XFER is on with the WG source for write data will be the external data bus. However when "surpress XFER" is on with WG this register will again be the source for write data.

EXTERNAL REGISTERS

The Winchester controller chip has three registers decoded that do not exist within the device. Their purpose is to provide versatile control unit design capability.

Registers 50, 51, and 70, when decoded, provide for a bidirectional connection of the microprocessor data bus with the buffer data bus through the Winchester controller chip on read or write.

The general purposes I/O lines can also be set up as decodes for two external register addresses, at 6EH and 6FH. Internally, register 6E and 6F are decoded and a read or write to one of these addresses can be made to generate a negative pulse on one of four pins. These signals are then used to enable a 74LS244 onto the microprocessor bus or to latch the bus into a 74LS373. In this manner the drive interface to several types of drives can be accommodated.

An example of the usage of Registers 6E and 6F decode is shown below.

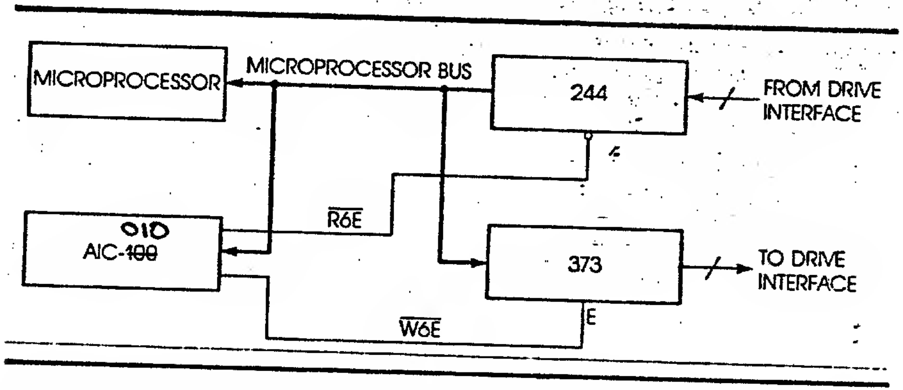


FIGURE 4. AIC-100 REGISTERS 6E AND 6F USAGE

APPENDIX

Fundamentally, any operation of the AIC-010 revolves around the following sequence of events:

* Initialize the chip by loading up the sequencer ram with the appropriate values.
* Setup the registers to handle the control of the data
* Start sequencing the chip at the appropriate ram location, by loading register 79 with the starting address.
* Monitor the status using register 79 and branching when appropriate, using the branch register, Register 78.

The following is a sequencer map of the AIC-010 programmed as a ST-506 controller. The contents of the RAM are loaded up by the external microprocessor after power up. In addition, the ECC polynomial has to also be set up by the support processor.

At this point, there are four fundamental operations that should be looked at. These are as follows:

* Soft sector format
* Soft sector read/write
* Hard sector format
* Hard sector read/write

The flow charts show the recommended steps to be followed in order to execute the above operations.

SEQUENCE MEMORY BIT MAP FOR ~~1001~~ ST-506 CONTROLLER:

	E0 THRU F7 DATA 76543210	C0 THRU D7 COUNT 76543210	A0 THRU B7 CONTROL 76543210	80 THRU 97 NEXT ADRS. 76543210		BRANCH R78	COMMENT
0	CYL	00	12	01	0		C
1	HD	00	12	02	1		H
2	SEC	00	12	03	2		S
3	FLAG	00	10	0C	3		F
4	6C	FF 7F	05 07	00	4		DATA
5	00	03	20	0F	5		RD ID
6	A1	80	02	07	6		IDAM
7	FE	00	02	00	7		ID SYNC
8	00	00	00	17	8		READ DATA DELAY
9	00	08	80	0A	9		WG↑ FOR DATA
A	A1	A0	02	0B	A		DATA AM
B	F8	00	02	04	B		DATA SYNC
C	00	43	00	94	C		ID ECC BRCH ON GOOD ID
D	00	43	00	94	D		DATA ECC BRCH ON CARRY
E	00	0B	80	06	E		WG↑ FMT ID
F	00	01	40	06	F		RG↑ FOR ID
10	4E	00—1F	80	0E	10		WG↑ GAP 1&4
11	00	05	00	09	11		FMT DATA FIELD
12	00	01	00	10	12		END DATA FIELD FOR FMT
13	00	01	00	16	13		" " " " LAST SEC
14	00	03	00	1F	14		" " " " RD/WR
15	00	00	00	D5	15		BRCH ON INDEX↑
16	4E	00	00	56	16		STOP ON INDEX↑
17	00	01	40	0A	17		RG↑ FOR DATA
STOP = 18	76543210 DATA	7654 DATA TYPE 3210 FIELD CNT. (-1)	76543210	76543210 N.A.	18 = STOP		

DATA TYPE: AM EN, ECC, SEBCA

CONTROL: WG ON, RG ON, WG OFF, STACK ENABL, INVALID IDRL, OUTPUT, COMPARE EN, DATA XFER

~~DATAS~~ OFF RG

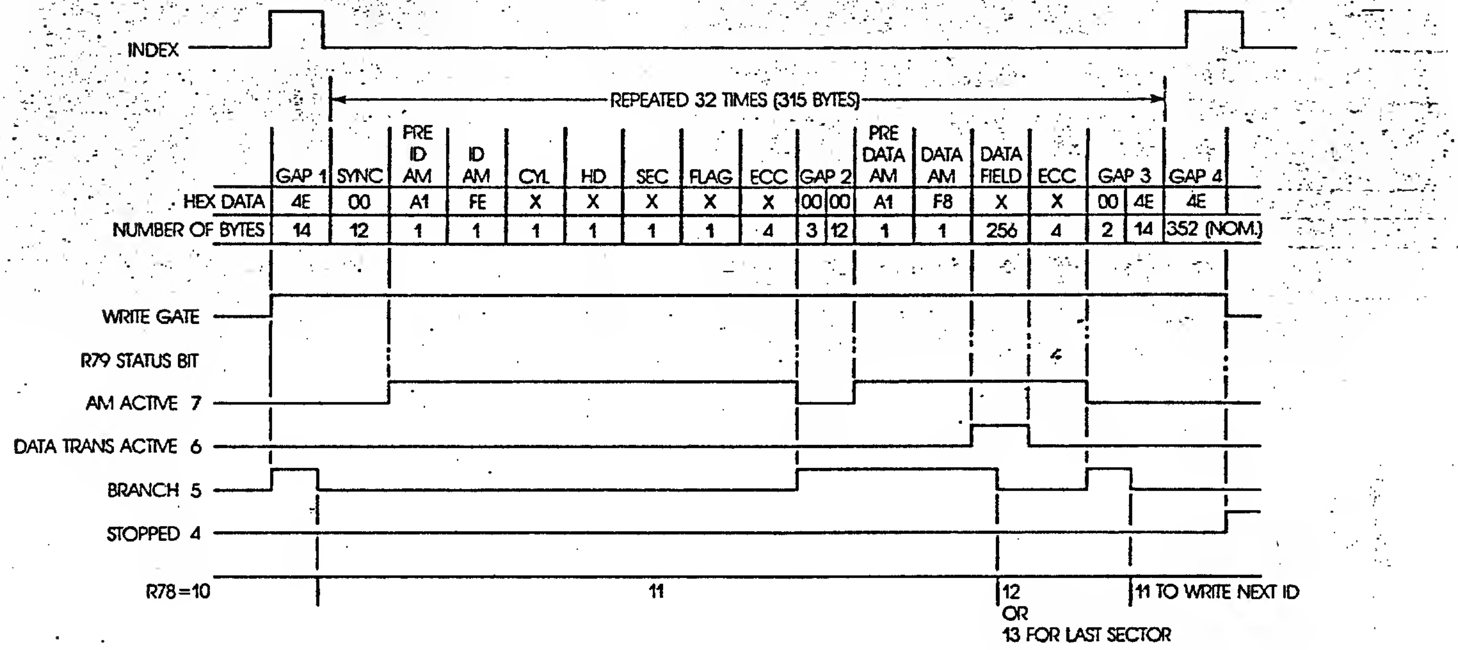
RG·ECC = 1		RG[redacted]=0	
000 –	NO BRANCH	000 –	NO BRANCH
001 –	ECC ERR STOP	001 –	STOP ON INPUT
010 –	NO COMPARE STOP	010 –	STOP ON INDX OR SECT
011 –	ERR OR COMP. STOP	011 –	STOP ON NOT EQUAL
100 –	GOOD ECC & COMP	100 –	BRANCH ON CARRY
101 –	ECC ERR	101 –	" " INPUT
110 –	NOT EQUAL	110 –	" " INDX OR SECT
111 –	ERR OR NOT EQ.	111 –	" " NOT EQUAL

Soft Sector Format

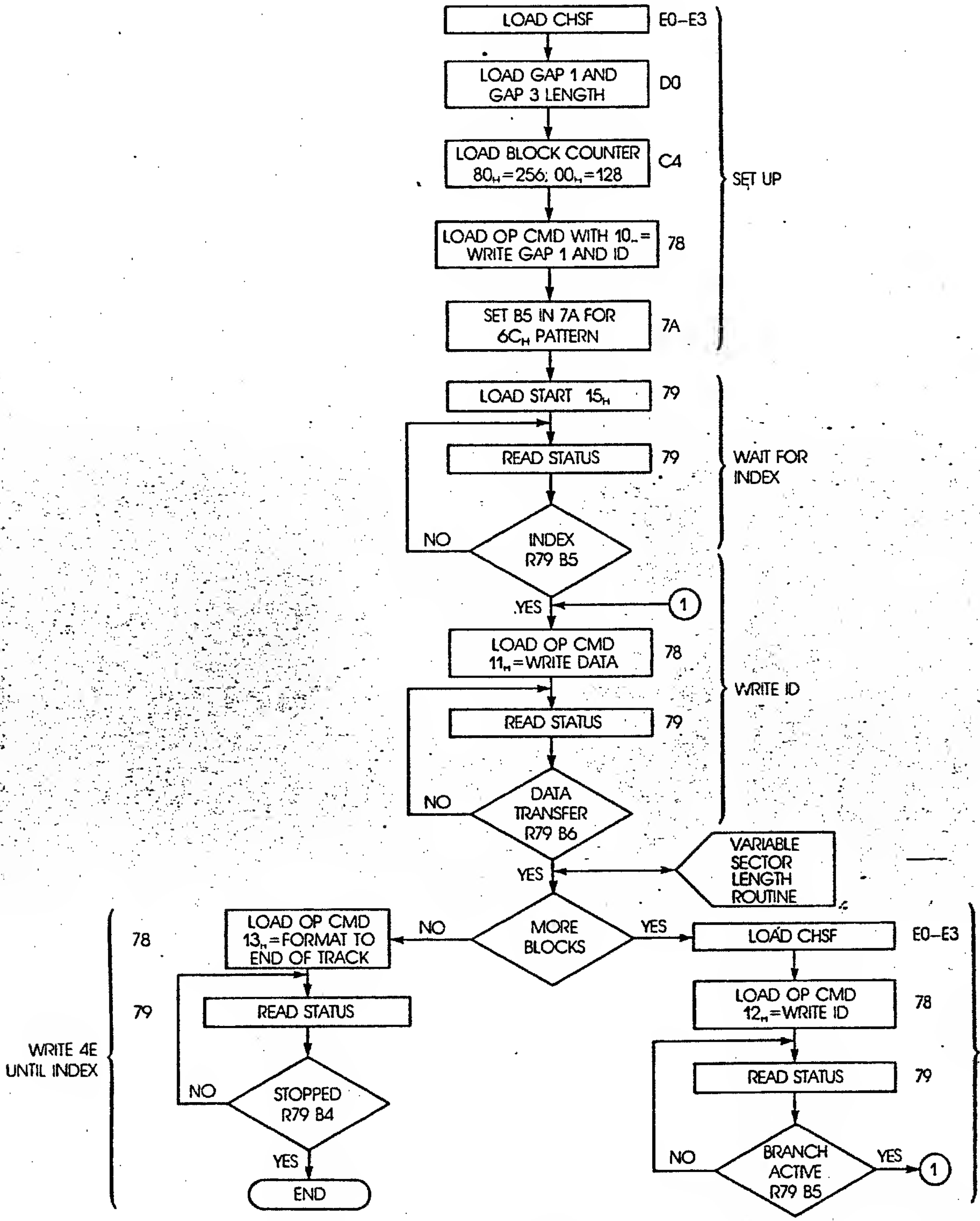
A standard Winchester soft sector format employs 256 byte sectors and MFM encoding. This format yields 32 sectors per track. Any sector size which is a multiple of 128 may be employed (discussed later).

The format track command sequence is as follows:

1. Set Registers E0, E1, E2, and E3 with the first sector ID (cylinder, head, sector and flag respectively).
2. Set Register D0 with Gap 1 and Gap 3 length.
3. Set Register C4 with either 80_H (256 byte count) or 00_H (128 byte count).
4. Set Register 78 (command register) with 10_H. This will format Gap 1 after index.
5. Set bit 5 in Register 7A for a $6C_H$ data pattern. Otherwise contents of the sector buffer will be used during write to the data field.
6. Set Register 79 with 15_H. This will start the format operation. The Winchester controller chip waits for index, after which Gap 1 will be written.
7. Read status from Register 79. If BRANCH ACTIVE (bit 5 is set) it means that index is past and Gap 1 is being written. After this the first ID will be written.
8. Set Register 78 with 11_H. This will cause the data to be written next.
9. Read status from Register 79. If bit 6 is set, the data field is now being written.
10. If there are no more sectors to be written, load Register 78 with 13_H, check Register 79 and wait for the controller to stop (bit 4 is set).
11. Otherwise set Register 78 with 12_H and update Register E0 through E3 with the next ID field to be written.
12. Wait for BRANCH ACTIVE by monitoring Register 79, bit 5. Gap 3 is being written.
13. Repeat steps 8, 9, 10, 11 and 12 for 31 times or the number of sectors to be formatted.



	GAP 1	SYNC	PRE ID AM	ID AM	CYL	HD	SEC	FLAG	ECC	GAP 2		PRE DATA AM	DATA AM	DATA FIELD	ECC	GAP 3		GAP 4
HEX DATA	4E	00	A1	FE	X	X	X	X	X	00	00	A1	F8	X	X	00	4E	4E
NUMBER OF BYTES	14	12	1	1	1	1	1	1	4	3	12	1	1	256	4	2	14	352 (NOM.)

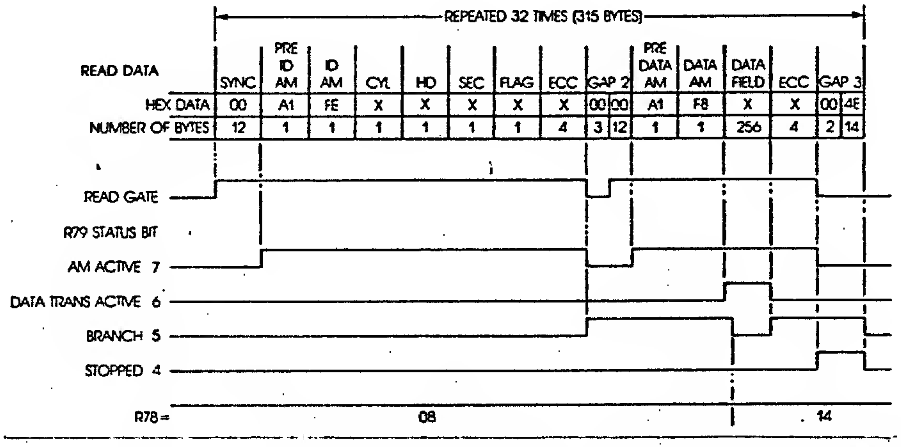

LOAD CHSF
E0–E3
LOAD GAP 1 AND GAP 3 LENGTH
D0
LOAD BLOCK COUNTER 80_H=256; 00_H=128
C4
LOAD OP CMD WITH 10_H= WRITE GAP 1 AND ID
78
SET B5 IN 7A FOR $6C_H$ PATTERN
7A
SET UP
LOAD START 15_H
79
READ STATUS
79
INDEX R79 B5
NO
WAIT FOR INDEX
YES
1
LOAD OP CMD 11_H=WRITE DATA
78
READ STATUS
79
DATA TRANSFER R79 B6
NO
WRITE ID
YES
VARIABLE SECTOR LENGTH ROUTINE
MORE BLOCKS
NO
YES
LOAD OP CMD 13_H=FORMAT TO END OF TRACK
78
READ STATUS
79
STOPPED R79 B4
NO
YES
END
WRITE 4E UNTIL INDEX
LOAD CHSF
E0–E3
LOAD OP CMD 12_H=WRITE ID
78
READ STATUS
79
BRANCH ACTIVE R79 B5
NO
YES
1
WRITE DATA

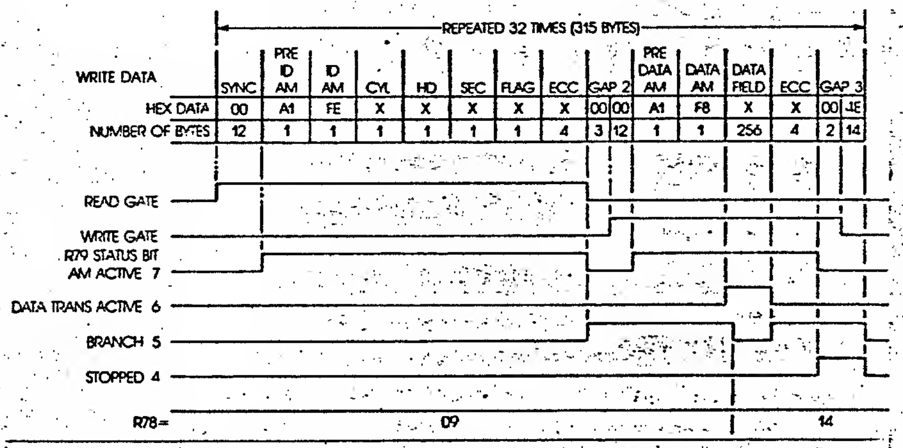
Soft Sector Read/Write

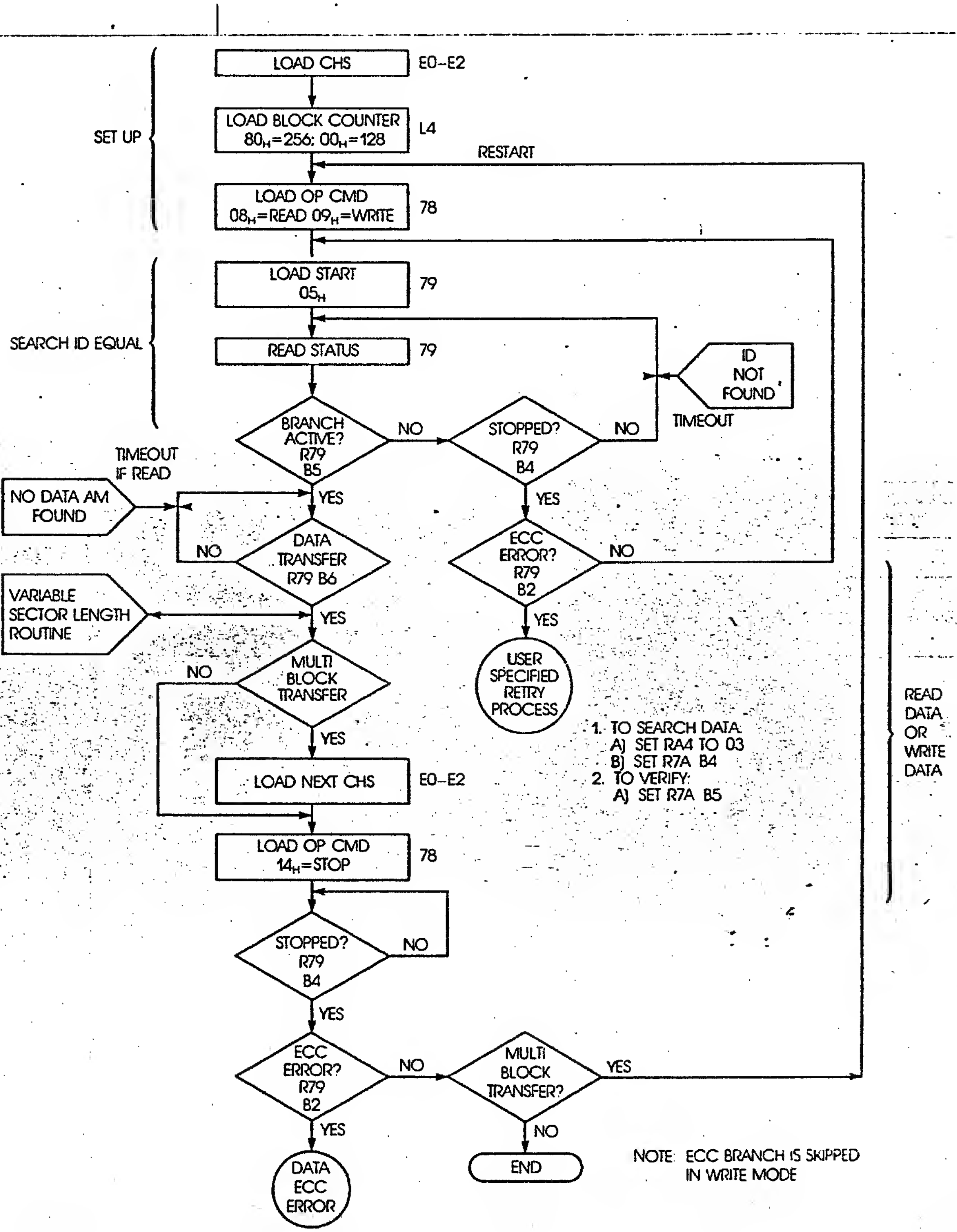
In order to read or write data, the heads have to first be positioned over the appropriate cylinder, and the relevant head must be selected. The following steps assume that the correct track has been reached.

The operation is performed as follows:

1. Set REO, E1 and E2 with the desired sector ID.
2. Set C4 with either OO (128 byte counter) or 8O (256 byte counter).
3. Set OP command (R78) with O8, the read data command, or with O9, the write command.
4. Set Start Reg (R79) with O5. This will turn on Read Gate and enable the VFO to look for an address mark.
5. Wait for BRANCH ACTIVE (R79, bit 5). If the correct ID field was read, the Winchester controller chip will continue on to read the data field. If an ID ECC error or incorrect sector was encountered, the stopped bit in R79 will be set. If so, go back to Step 4.
6. Wait for DATA TRANSFER (R79 bit 6). Read data is now being transferred to the sector buffer, or write data from the buffer.
7. If this is a multiblock transfer, update EO-E2 with next sector ID while data is being transferred.
8. Set OP command (R78) with 14. This will stop the Winchester controller chip at the end of the data field ECC.
9. Wait for STOPPED (R79 bit 4).
10. If it is a read command, test ECC ERROR (R79, bit 2). If it is set, go to the error correction routine. If not, continue on to read the next sector (Step 3) or end.






LOAD CHS
E0–E2
LOAD BLOCK COUNTER
80_H=256; 00_H=128
L4
SET UP
RESTART
LOAD OP CMD
08_H=READ 09_H=WRITE
78
LOAD START
05_H
79
SEARCH ID EQUAL
READ STATUS
79
ID NOT FOUND
TIMEOUT
BRANCH ACTIVE? R79 B5
NO
STOPPED? R79 B4
NO
YES
YES
TIMEOUT IF READ
NO DATA AM FOUND
DATA TRANSFER R79 B6
NO
ECC ERROR? R79 B2
NO
YES
YES
VARIABLE SECTOR LENGTH ROUTINE
USER SPECIFIED RETRY PROCESS
MULTI BLOCK TRANSFER
NO
YES
1. TO SEARCH DATA
A) SET RA4 TO 03
B) SET R7A B4
2. TO VERIFY:
A) SET R7A B5
READ DATA OR WRITE DATA
LOAD NEXT CHS
E0–E2
LOAD OP CMD
14_H=STOP
78
STOPPED? R79 B4
NO
YES
ECC ERROR? R79 B2
NO
MULTI BLOCK TRANSFER?
YES
YES
NO
DATA ECC ERROR
END
NOTE: ECC BRANCH IS SKIPPED IN WRITE MODE

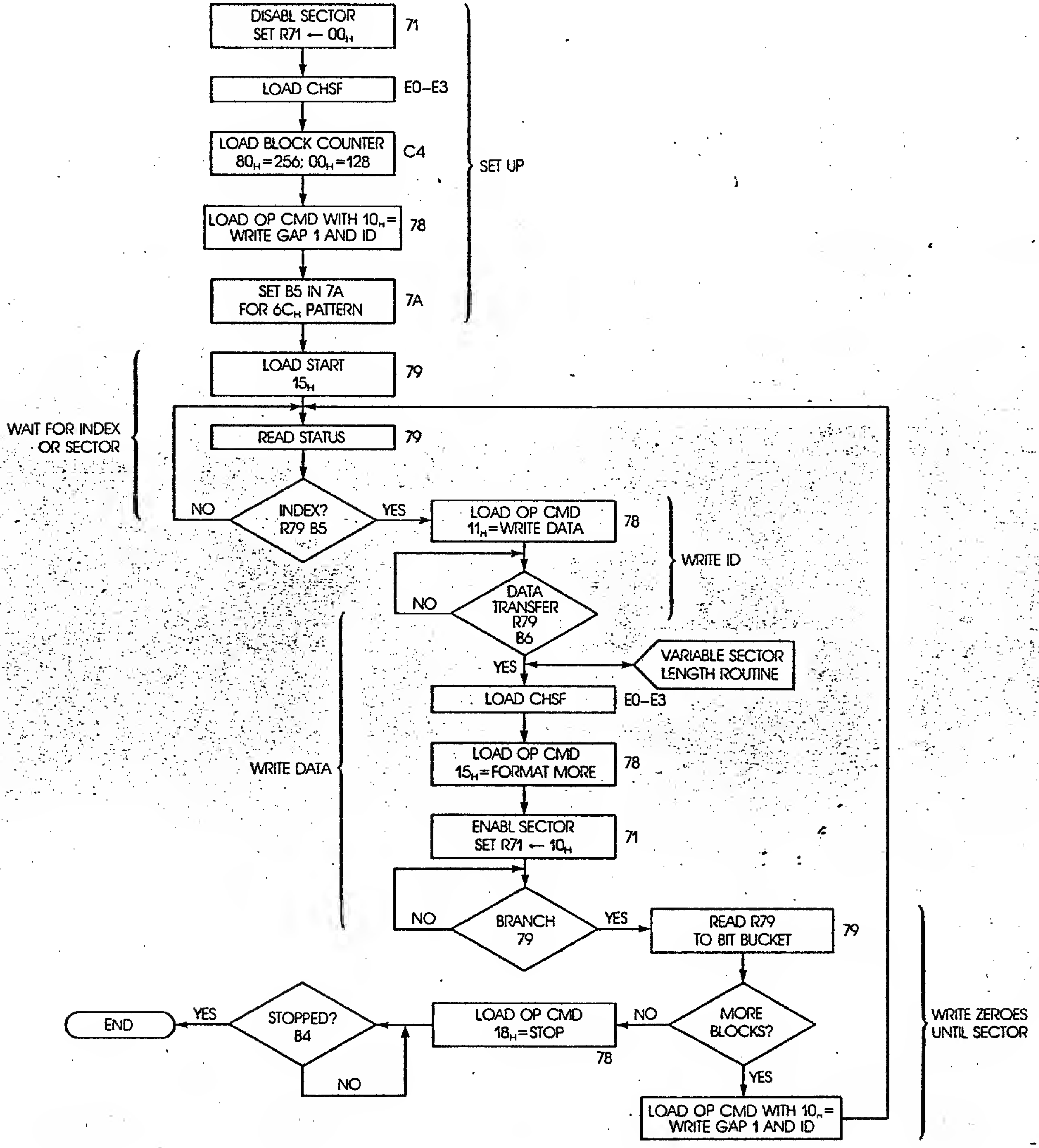
Hard Sector Format

As mentioned eorlier, the AIC-1OO controller chip is copable of supporting hard sectored drives. Hord sector drives differ from soft sector drives in thot, between every adjocent sector on a track, there is a sector mark, and this is used to identify the beginning of a sector.

A hord sectored format operation is performed os follows:

1. Disoble sector mork by setting Register 71 to OO_H. Thus the controller chip will wait for the index mark before writing out Gap 1.
2. Lood Reglsters EO-E3 with the sector ID (cylinder, head, sector and flog).
3. Lood Register C4 with the data length ($8O_H$ =256, OO_H =128).
4. Load Register 78 with $1O_H$. This will cause Gap 1 to be written after index.
5. Set bit 5 in Register 7A for $6C_H$ data pattern. Otherwise sector buffer is used.
6. Load Register 79 with 15_H to start formatting.
7. Read status from Register 79. If bit 5 is set, then Gap 1 is being written. After this the ID is written.
8. Load Register 78 with 11_H. This tells the controller to write the data next.
9. Read status from Register 79. If bit 6 is set, then data is being tronsferred.
10. Update Registers EO-E3 with the next sector ID. This has to be done before even checking if there are more blocks, since, on a hard sector drive, the timing is more critical.
11. Lood Register 78 with 15_H. This tells the controller to write zeros until the next sector mork is encountered.
12. Enoble sector branch by setting Register 71 to $1O_H$.
13. Read stotus from Register 79 and branch when active (bit 5 is set). This means that a sector mark was encountered.
14. Read status from Register 79 and discord contents. This guarantees a reset.
15. Check to see if any more blocks have to be written. If there is no more to be done, then lood Register 78 with 18_H. This tells the controller to stop. Monitor Register 79 bit 4 (stop bit) before leaving the routine.
16. If more blocks have to be written, then load Register 78 with $1O_H$ and repeat steps 7 through 15.

NOTE: It is suggested that Gap 1 length be kept to zero. Thus, during format, after the sector mark is encountered, the controller will write out the Sync for ID field. Inter record separation is provided by the controller writing OO from end of data field to next sector mark.

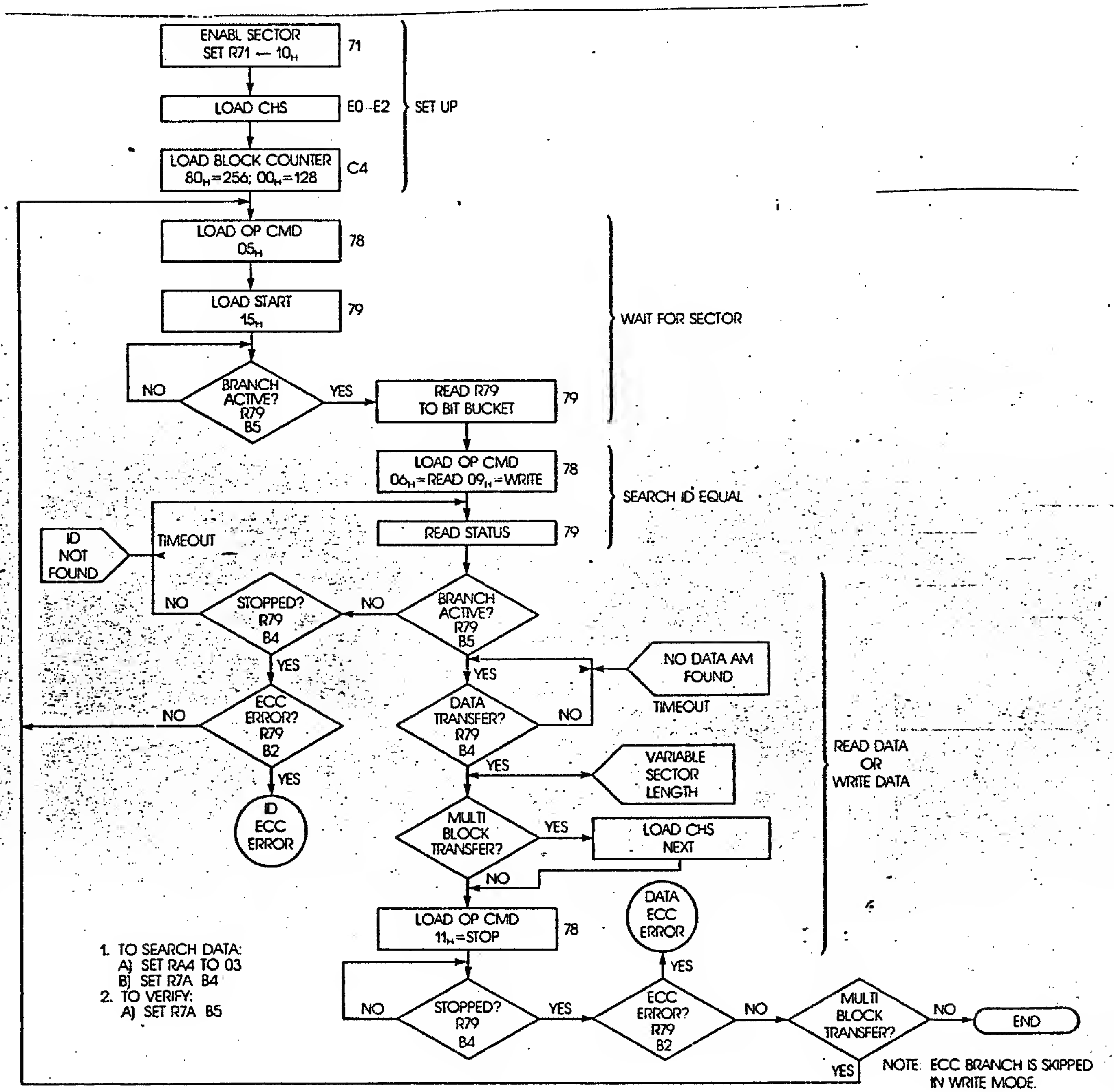

DISABL SECTOR
SET R71 ← 00$_H$
71
LOAD CHSF
E0–E3
LOAD BLOCK COUNTER
80$_H$=256; 00$_H$=128
C4
SET UP
LOAD OP CMD WITH 10$_H$=
WRITE GAP 1 AND ID
78
SET B5 IN 7A
FOR 6C$_H$ PATTERN
7A
LOAD START
15$_H$
79
WAIT FOR INDEX
OR SECTOR
READ STATUS
79
NO
INDEX?
R79 B5
YES
LOAD OP CMD
11$_H$=WRITE DATA
78
WRITE ID
NO
DATA
TRANSFER
R79
B6
YES
VARIABLE SECTOR
LENGTH ROUTINE
LOAD CHSF
E0–E3
WRITE DATA
LOAD OP CMD
15$_H$=FORMAT MORE
78
ENABL SECTOR
SET R71 ← 10$_H$
71
NO
BRANCH
79
YES
READ R79
TO BIT BUCKET
79
END
YES
STOPPED?
B4
NO
LOAD OP CMD
18$_H$=STOP
78
NO
MORE
BLOCKS?
YES
WRITE ZEROES
UNTIL SECTOR
LOAD OP CMD WITH 10$_H$=
WRITE GAP 1 AND ID

Hard Sector Read/Write

As mentioned earlier, a hard sector drive has a sector mark between adjacent sectors. Thus the controller starts reading the ID field after the next sector mark is encountered.

The read and write operations differ in that after a write operation, no ECC test is necessary. The read or write data operation is performed as follows:

1. Enable branch on sector mark by setting Register 71 to 10_H.
2. Set Registers EO, E1 and E2 with the desired sector ID.
3. Set C4 with either OO (128 byte counter) or 8O (256 byte counter).
4. Load Register 78 with 05_H, the read ID command. The VFO will look for a SYNC of A1 after read gate is turned on.
5. Load Register 79 with 15_H. The controller will wait for index or the next sector mark (since R71 was set to 10_H) and read the ID field.
6. Wait for BRANCH ACTIVE (R79, bit 5). This means that the next sector mark or index has been encountered. The read gate will not be turned on.
7. Read and discard Register 79 to ensure reset.
8. Load Register 78 with O8, the read command or a O9 for a write command.
9. Wait for BRANCH ACTIVE (R79 bit 5). If the correct ID field was read, the Winchester controller chip will continue on to read the data field. If an ID ECC error or incorrect sector was encountered, the stopped bit in Register 79 will be set. If so, go back to Step 4.
10. Wait for DATA TRANSFER (R79, bit 6). Read data is now being transferred to the sector buffer, or from the buffer in the case of a write.
11. If this is a multi-block transfer, update EO-E2 with next sector ID while data is being transferred.
12. Set Register 78 with 14_H (STOP command). This will stop the controller chip at the end of the data field ECC.
13. Wait for STOPPED (R79, bit 4).
14. Test ECC ERROR (R79, bit 2). If it is set, go to the error correction routine. If not, continue on to the next sector (Step 4) or end.



1. TO SEARCH DATA:
   A) SET RA4 TO 03
   B) SET R7A B4
2. TO VERIFY:
   A) SET R7A B5

NOTE: ECC BRANCH IS SKIPPED IN WRITE MODE.

AIC-010 SOFT SECTOR FORMAT SEQUENCING

The following example shows the operation of the AIC-010 (programmed as a ST-506 controller) during a soft sector format operation. The example is presented using "Snapshots" of the AIC-010 and system activity descriptions. The time references assigned to each snapshot or event represent the approximate times associated with a ST-506 disk drive. The drive is formatted for 256 byte sectors.

SYSTEM ACTIVITY (t = 0.0 to 0.5ms)

Host initializes AIC-010 registers

Branch register = 10H
RD0H = Gap 1 length (14 bytes)
RE0H = I.D. Cylinder Value
RE1H = I.D. Head Value
RE2H = I.D. Sector Value
RE3H = I.D. Flag Value

Host writes 15H to AIC-010 sequencer start register (R79H) at t = .5ms

AIC-010 STATE (t = 0.5ms)

Current Address:	15H
Next Address:	15H Loop until Branch
Branch Control:	110 & RG = 0 Branch on Index or Sector
Branch Register:	10H
Control:	N.A.
Data Field:	N.A.
Count:	N.A.

SYSTEM ACTIVITY (t = 0.5 to 10.0ms)

AIC-010 waits for Leading edge of Index to begin writing Gap 1. When Index is detected by the AIC-010 the Branch Active Flag is set.

AIC-010 STATE (t = 10.0ms)

Current Address:	10H
Next Address:	0EH
Branch Control:	000 No Branch
Branch Register:	10H
Control:	80H (WG on)
Data Field:	4EH Gap 1 Character
Count:	Set by host at initialization (14 bytes)

SYSTEM ACTIVITY (t = 10.0 to 10.022ms)

AIC-010 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 11H (write I.D.)

AIC-010 STATE (t = 10.022ms)

```
Current Address:  0EH
   Next Address:  06H
 Branch Control:  000
                  No Branch
Branch Register:  11H
        Control:  80H (WG on)
     Data Field:  00H
          Count:  0BH (Write 12 bytes of 00 sync code)
```

SYSTEM ACTIVITY (t = 10.022 to 10.040ms)

AIC-010 NRZ output is 00H Sync characters. The host is waiting for data transfer.

AIC-010 STATE (t = 10.040ms)

```
Current Address:  06H
   Next Address:  07H
 Branch Control:  000
                  No Branch
Branch Register:  11H
        Control:  02H (Not valid with RG =0)
     Data Field:  A1H
          Count:  1000 0000 (Strobe AM output bit as defined by
                             WAM Control register 7BH)
```

SYSTEM ACTIVITY (t = 10.040 to 10.041ms)

AIC-010 NRZ output is A1H Address mark character. The AM output will be strobed at the bit(s) time defined by the WAM control register.

AIC-010 STATE (t = 10.041ms)

```
Current Address:  07H
   Next Address:  00H
 Branch Control:  000
                  No Branch
Branch Register:  11H
        Control:  02H (Not valid with RG = 0)
     Data Field:  FEH
          Count:  00H (Write 1 byte of I.D field sync byte)
```

SYSTEM ACTIVITY (t = 10.041 to 10.043ms)

AIC-010 NRZ output is FEH I.D. field sync character.

AIC-010 STATE (t = 10.043ms)

Current Address:	00H
Next Address:	01H
Branch Control:	000 No Branch
Branch Register:	11H
Control:	02H (Not valid with RG =0)
Data Field:	Cylinder value established by host
Count:	00H (Write 1 byte)

SYSTEM ACTIVITY (t = 10.043 to 10.050ms)

AIC-010 NRZ output is I.D. cylinder, head, sector, flag bytes as micro-sequencer executes locations 00, 01, 02 and 03. The next location after 03H (Flag byte write) is 0CH (I.D. ECC write).

AIC-010 STATE (t = 10.050ms)

Current Address:	0CH
Next Address:	14H
Branch Control:	100 Branch at end of ECC
Branch Register:	11H
Control:	00H
Data Field:	N.A.
Count:	0100 0011 (Write 4 bytes of ECC)

SYSTEM ACTIVITY (t = 10.050 to 10.056ms)

AIC-010 NRZ output is I.D. ECC characters (4 bytes)

AIC-010 STATE (t = 10.056ms)

Current Address:	11H
Next Address:	09H
Branch Control:	000 No Branch
Branch Register:	11H
Control:	00H
Data Field:	00
Count:	05H (Write 6 bytes of 00H)

AIC-010 STATE (t = 10.060ms)

Current Address:	09H
Next Address:	0AH
Branch Control:	000 No Branch
Branch Register:	11H
Control:	80H
Data Field:	00
Count:	08H (Write 9 bytes of 00H)

SYSTEM ACTIVITY (t = 10.056 to 10.080ms)

AIC-010 NRZ output is 15 bytes of 00H (Data field sync bytes)

AIC-010 STATE (t = 10.080ms)

Current Address:	0AH
Next Address:	0BH
Branch Control:	000 No Branch
Branch Register:	11H
Control:	02H (Not valid with RG = 0)
Data Field:	A1H
Count:	1010 0000 (Write 1 byte Data field AM AM output will be strobed as defined by WAM control register. CLKA and bit ring will be synchronized)

AIC-010 STATE (t = 10.080ms)

Current Address:	0BH
Next Address:	04H
Branch Control:	000 No Branch
Branch Register:	11H
Control:	02H (Not valid with RG = 0)
Data Field:	F8H
Count:	00H (Write 1 byte Data field syn)

SYSTEM ACTIVITY (t = 10.080 to 10.083ms)

AIC-010 NRZ output is A1 (with AM output strobe) followed by F8H sync byte.

AIC-010 STATE (t = 10.083ms)

Current Address:	04H
Next Address:	0DH
Branch Control:	000 No Branch
Branch Register:	11H
Control:	05H (Output & Data Transfer bits set)
Data Field:	6CH
Count:	FFH (256 byte multiples sent by host)

SYSTEM ACTIVITY (t = 10.083 to 10.492ms)

AIC-010 NRZ output is the sector data from the host buffer or a fill value of 6CH. Host buffer data will be transferred if the "Supress Transfer" bit (R7A bit 5) is off. If this bit is set then the 6CH fill byte will be output. At this point the Data Transfer flag (R79 bit 6) will be set and the host will load the branch register with 12H (end of data field format branch). Setting "Inhibit Data Field Carry" will allow a second 256 byte transfer (for 512 byte sectors).

AIC-010 STATE (t = 10.492ms)

Current Address:	0DH
Next Address:	14H
Branch Control:	100
	Branch after ECC complete
Branch Register:	12H
Control:	00H
Data Field:	00H
Count:	43H (Write 4 ECC bytes)

SYSTEM ACTIVITY (t = 10.492 to 10.498ms)

AIC-010 NRZ output is Data Field ECC. At the end of the ECC, branch is active. The host has been waiting for branch active to set the branch register to 12H (Write I.D. for the next sector).

AIC-010 STATE (t = 10.498ms)

Current Address:	12H
Next Address:	10H
Branch Control:	000
	No Branch
Branch Register:	12H
Control:	00H
Data Field:	00H
Count:	01H (Write 2 bytes of 00H bytes)

SYSTEM ACTIVITY (t = 10.498 to 10.502ms)

AIC-010 NRZ output is 2 bytes of 00H. Host has sensed Branch Active and set Register 78 to 12H, setting up the next I.D. field write.

AIC-010 STATE (t = 10.502ms)

Current Address:	10H
Next Address:	0EH
Branch Control:	000
	No Branch
Branch Register:	11H
Control:	80H (WG on)
Data Field:	4EH
Count:	Set by host at initialization (14 bytes)

SYSTEM ACTIVITY (t = 10.502 to 10.524ms)

At this point the AIC-010 is writing Gap 3, which is the same as when it was writing Gap 1. Following this, 12 bytes of 00H, the I.D. AM (A1 FE) and the 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-010 STATE (t =   ms)

Current Address:	13H
Next Address:	16H
Branch Control:	000 No Branch
Branch Register:	13H
Control:	00H
Data Field:	00H
Count:	01H (Write 2 bytes Of 00H bytes)

AIC-010 STATE (t =   ms)

Current Address:	16H
Next Address:	16H
Branch Control:	010 Stop on Index
Branch Register:	13H
Control:	00H
Data Field:	4EH
Count:	00H (Write 1 byte of 4EH then loop to 16H)

SYSTEM ACTIVITY (t =   ms)

The AIC-010 continues to write 4EH bytes until Indexis encountered. At index the AIC-010 stops and the track format is complete.

MULTI-SECTOR READ OR WRITE

Multi-sector reads or writes are accomplished by loading the next sector address to be found while DATA TRANSFER is active (Reg 79, bit 6) for the present sector and restarting the read or write at Step 3 immediately after the stopped bit is set.

VERIFY SECTOR

A Verify Sector is accomplished by setting the SUPPRESS TRANSFER in the OP Modifier Register (Reg 7A, bit 5) and then performing the read data command sequence. This will verify that the ECC is good for the data field without generating a CLKB.

SEARCH SECTOR DATA

A search of the data field is performed by setting OP Modifier Register, (Reg 7A, bit 4) and the Search Enable Register (Reg 4A, bit 2) then entering the read data sequence. The contents of the sector buffer will be compared, byte for byte, with the data read from the disk. The result of this comparison is latched into the Status Register (Reg 79, bit 0 and 1). Be sure to reset both Register 7A, bit 4 and Register A4, bit 2 after completion of search.

VARIABLE SECTOR SIZE

The Winchester controller chip has an 8 bit data field length counter. This field is programmable, and by setting this field to any value from OOH to FFH, any sector length up to 256 bytes can be written to the drive. For sector sizes greater than 256, bit 7 of the OP Modifier Register (Reg 7A) must be employed.

By setting this bit during DATA TRANSFER before the first count has expired, the Winchester controller chip will be inhibited from going on to ECC and another 256 bytes of data will be transferred. OP Modifier bit 7 (Reg 7A, bit 7) will be automatically reset whenever the counter overflows. By testing this bit, a count of 256 byte segments may be accomplished.

PROGRAMMING 8-BIT ECC CORRECTION

After each read data operation a read error may have occurred. This may be determined by reading Register 79. If bit 2 is set, an error did occur and the following procedure is employed to determine if the error is correctable. Note that the majority of read errors are soft (i.e., caused by noise) and that the correction algorithm is time consuming. It is recommended that the record be re-read before attempting correction.

The general flow of the algorithm for 8 bit correction is as follows:

1. Off-load the 32-bit syndrome into local RAM.
2. Shift the syndrome back into the ECC register in reverse order, swapping the syndrome end for end.
3. Change the ECC polynomial from forward to reciprocal.
4. Shift the ECC until all bits except the high order (24-31) bits are zero (correctable) or the number of shifts are greater than the number of bits in the record (uncorrectable).
5. If correctable, the number of shifts represent 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 exclusive ORed with the appropriate bits in memory to correct the error.

Detailed Programming Steps

1. After a read error is detected, disable feedback by setting R71=04_H.
2. Store contents of R73 in RAM (*x*).
3. Shift ECC 8 times by setting R71=06_H eight times.
4. Store contents of R73 in RAM (*x*+1).
5. Shift ECC 8 times by setting R71=06_H eight times.
6. Store contents of R73 in RAM (*x*+2).
7. Shift ECC 8 times by setting R71=06_H eight times.
8. Store contents of R73 in RAM (*x*+3).
9. Clear ECC and disable feedback by setting R71 to 08 and then 04.
10. Right rotate location RAM (*x*+3) and test if carry is set: (i.e., test bit 0) if set, then load R71=07_H if not set, then load R71=06_H repeat operation 7 more times to load entire byte.
11. Repeat step 10 for RAM locations *x*+2, *x*+1, and *x* until all 32 bits of the syndrome are loaded into the ECC in reverse order.
12. Load R74=00_H and R77=01_H to enable the reciprocal polynomial and disable the forward polynomial.
13. Compute record length in bits: # of bits per data field=ECC+Data+AM and SYNC for a 256 byte record length in bits =4•8+256•8+2•8=2096.
14. Enable feedback by setting R71=00_H.
15. Shift ECC once by setting R71=02_H and increment a software counter.
16. Test to see if the software counter is greater than the record length; if yes, the error is uncorrectable, re-enable the forward polynomial and end operation.
17. Test to see if R72=00_H; if yes, go to Step 18 if no, go to Step 15.
18. Subtract hardware offset of 7 from the shift count. If a correctable error is located within the ECC or the SYNC & AM bytes (the shift count <=32), the data field is good and no further action is required. Subtract 32 from the shift count.
19. The bit displacement (shift count) must now be converted to a byte offset by right shifting the count 3 times. The value of the shift count equals the bit displacement from end of the record.
20. R73 is the mirror image of the error pattern. Form the error mask data (2 bytes) by concatenating R73 with a zero byte.
21. Get the shift count (E) for error mask data by extracting the lower 3 bits from the shift count obtained in Step 18.
22. Right shift the error mask data with MSB (bit 15) set to zero. Repeat E-1 times more.
23. Mirror the error mask data byte by byte.
24. The 2 byte error mask data may now be EXORed with the data in memory to correct the error. The byte offset obtained in Step 19 is low order byte offset.

NOTES:

1) For 5-bit ECC correction, the following modification is necessary.
   Step 17: Test to see if R72=OO_H and R73 bits O, 1, 2 are zero;
   if yes, go to Step 18.
   if no, go to Step 15.
   Step 18: R73 bits 3-7 are the mirror image of the error pattern. (O-7 for 8 bit ECC)
2) In Step 23, say, if the original error mask data is 5C 9A, after mirroring the data is 3A 59.

ABSOLUTE MAXIMUM RATINGS

Ambient Temperature Under Bias	0°C to 70°C
Storage Temperature	−65°C to 150°C
Voltage on Any Pin with Respect to Ground	−0.5 to 7 volts
Power Dissipation	1 watt
Power Supply Voltage	4.75 to 5.25 volts

NOTE: Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

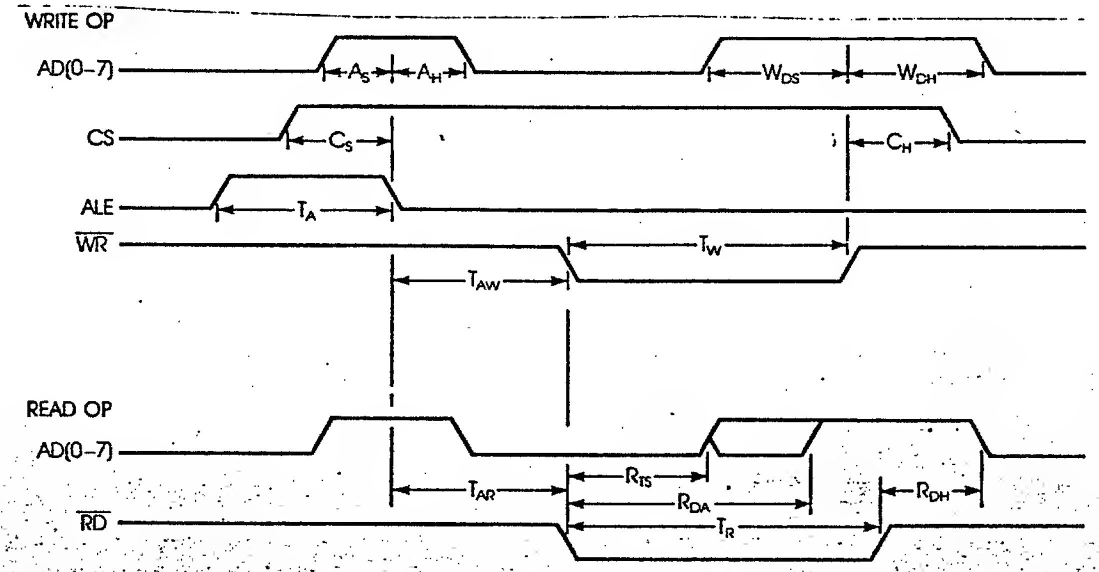
D.C. CHARACTERISTICS

SYMBOL	PARAMETER	MIN	MAX	UNITS	CONDITIONS
V_{IL}	Input Low Voltage (NRZ, RD/REF, WAM/AMD)	−0.5	0.5	V	
V_{IL}	Input Low Voltage (All Other)	−0.5	0.3	V	
V_{IH}	Input High Voltage	3.0	V_{CC}+0.5	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2 mA*
V_{OH}	Output High Voltage	2.2			I_{OH} = 400 µA
I_{CC}	Supply Current		200	mA	
I_{IL}	Input Leakage	−10	10	µA	$0 < V_{IN} \leq V_{CC}$
I_{OL}	Output Leakage Off State	−100	100	µA	$0.45 < V_{OUT} < V_{CC}$
C_{IN}	Input Capacitance		10	pF	
C_{OUT}	Output Capacitance		30	pF	

NOTE: For RG and WG, I_{OL} = 5 mA.

010

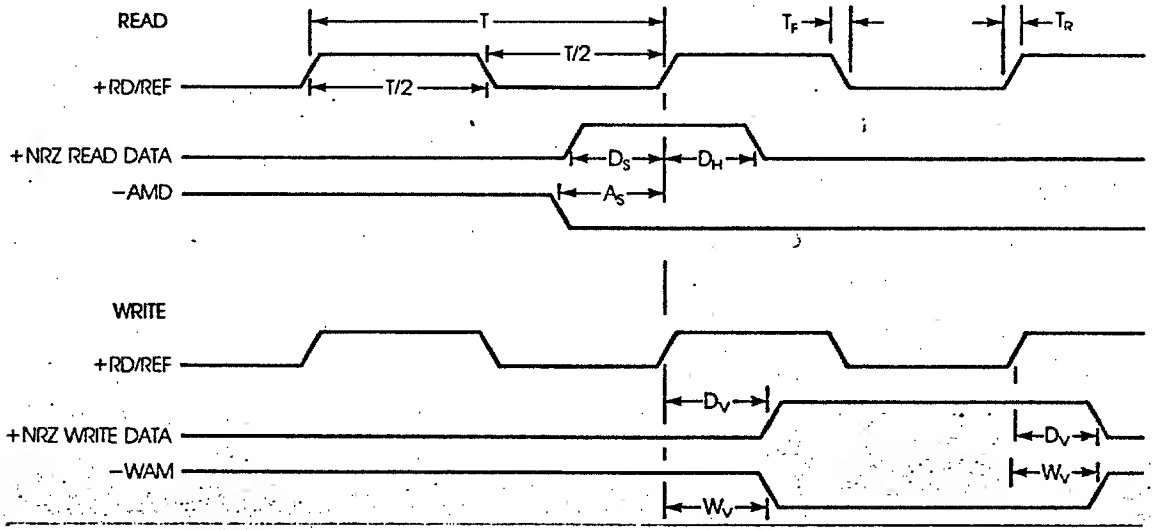
AIC-~~100~~ MICROPROCESSOR INTERFACE TIMING



SYMBOL	PARAMETER	MIN	MAX	UNITS
T_A	ALE Width	45		ns
T_{AW}	ALE ↓ to $\overline{WR}$ ↓	60		ns
T_{AR}	ALE ↓ to $\overline{RD}$ ↓	60		ns
T_W	$\overline{WR}$ Width	200		ns
T_R	$\overline{RD}$ Width	230		ns
A_S	ADRS Valid to ALE ↓	25		ns
A_H	ALE ↓ to ADRS $\overline{\text{Valid}}$	20		ns
C_S	CS Valid to ALE ↓	25		ns
C_H	$\overline{RD}$ or $\overline{WR}$ ↑ to CS ↓	0		ns
W_{DS}	Write Data Valid to $\overline{WR}$ ↑	40		ns
W_{DH}	$\overline{WR}$ ↑ to Write Data $\overline{\text{Valid}}$	0		ns
R_{TS}	$\overline{RD}$ ↓ to AD (0-7) Active	40		ns
R_{DA1}	$\overline{RD}$ ↓ to Data Valid (Regs 71-7F)		150	
R_{DS2}	$\overline{RD}$ ↓ to Data Valid (All Other Regs)		230	ns
R_{DH}	$\overline{RD}$ ↑ to Data $\overline{\text{Valid}}$	80	130	ns

010

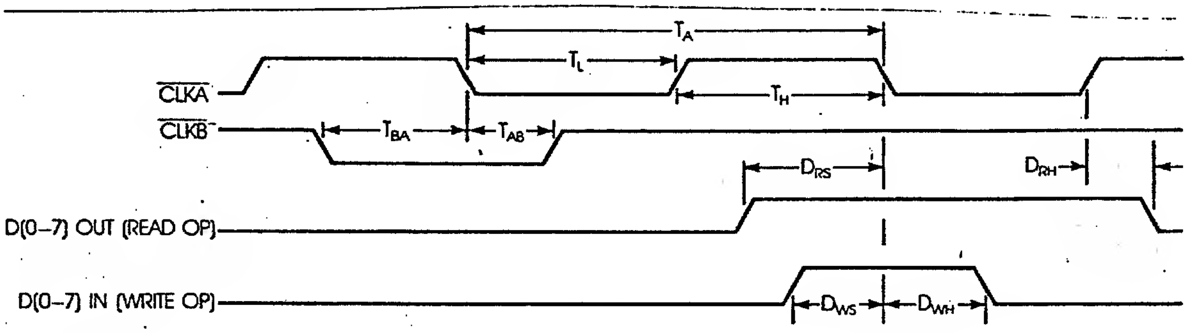
AIC-~~100~~ DISK READ/WRITE TIMING



SYMBOL	PARAMETER	AIC-100		AIC-100-10		UNITS
		MIN	MAX	MIN	MAX	
T	RD/REF Period	180	5000	100	5000	ns
T/2	RD/REF Period ÷ 2	85		45		ns
T_R	RD/REF Rise Time		20		10	ns
T_F	RD/REF Fall Time		20		10	ns
D_S	Data In Valid to RD/REF ↑	50		20		ns
D_H	RD/REF ↑ to Data In $\overline{\text{Valid}}$	10		50		ns
A_S	AMD Valid to RD/REF ↑	50		20		ns
D_V	RD/REF ↑ to Data Out	15	100	10	60	ns
W_V	RD/REF ↑ to W_{AM} Out	15	100	10	60	ns

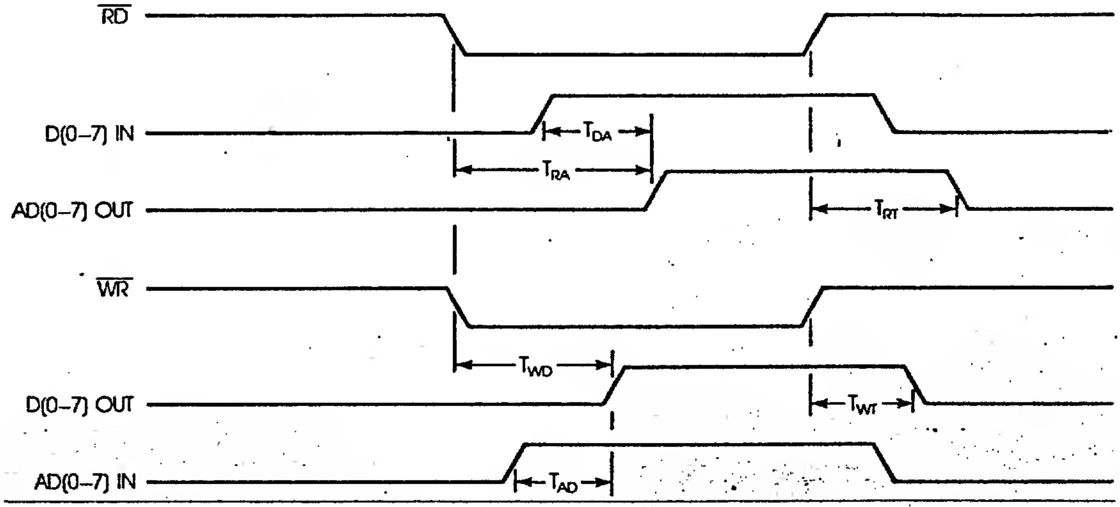
010

AIC-~~400~~ READ/WRITE DATA BUS TIMING



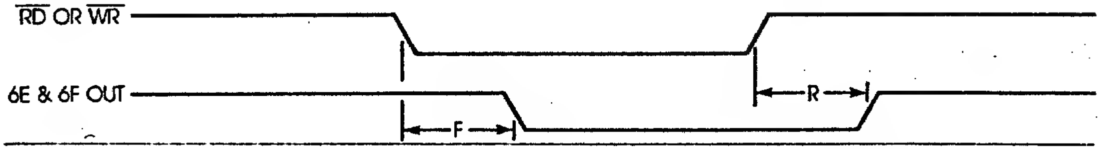
SYMBOL	PARAMETER	MIN	MAX	UNITS
T_A	$\overline{CLKA}$ Period	200		ns
$T_H=T_L$	$\overline{CLKA}$ Low or High Time	95		ns
T_{BA}	$\overline{CLKB}$ ↓ to CLKA ↓	90		ns
T_{AB}	$\overline{CLKA}$ ↓ to $\overline{CLKB}$ ↑	100		ns
D_{RS}	D(0-7) In Valid to $\overline{CLKA}$ ↓	60		ns
D_{RH}	$\overline{CLKA}$ ↑ to D(0-7) Out $\overline{Valid}$	20	80	ns
D_{WS}	D(0-7) In Valid to $\overline{CLKA}$ ↓	50		ns
D_{WH}	$\overline{CLKA}$ ↓ to D(0-7) In $\overline{Valid}$	20		ns

AIC-~~100~~010 REG 50, 51 AND 70 TIMING



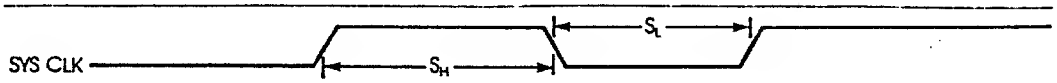
SYMBOL	PARAMETER	MIN	MAX	UNITS
T_{DA}	D(0-7) In Valid to AD(0-7) Out Valid		115	ns
T_{RA}	$\overline{RD}$ ↓ to AD(0-7) Out Valid		160	ns
T_{RT}	$\overline{RD}$ ↑ to AD(0-7) Out Tri State	35	100	ns
T_{WD}	$\overline{WR}$ ↓ to D(0-7) Out Valid		120	ns
T_{WT}	$\overline{WR}$ ↑ to D(0-7) Out Tri State	40	130	ns
T_{AD}	AD(0-7) In Valid to D(0-7) Out Valid		95	ns

AIC-~~100~~010 REG 6E AND 6F TIMING



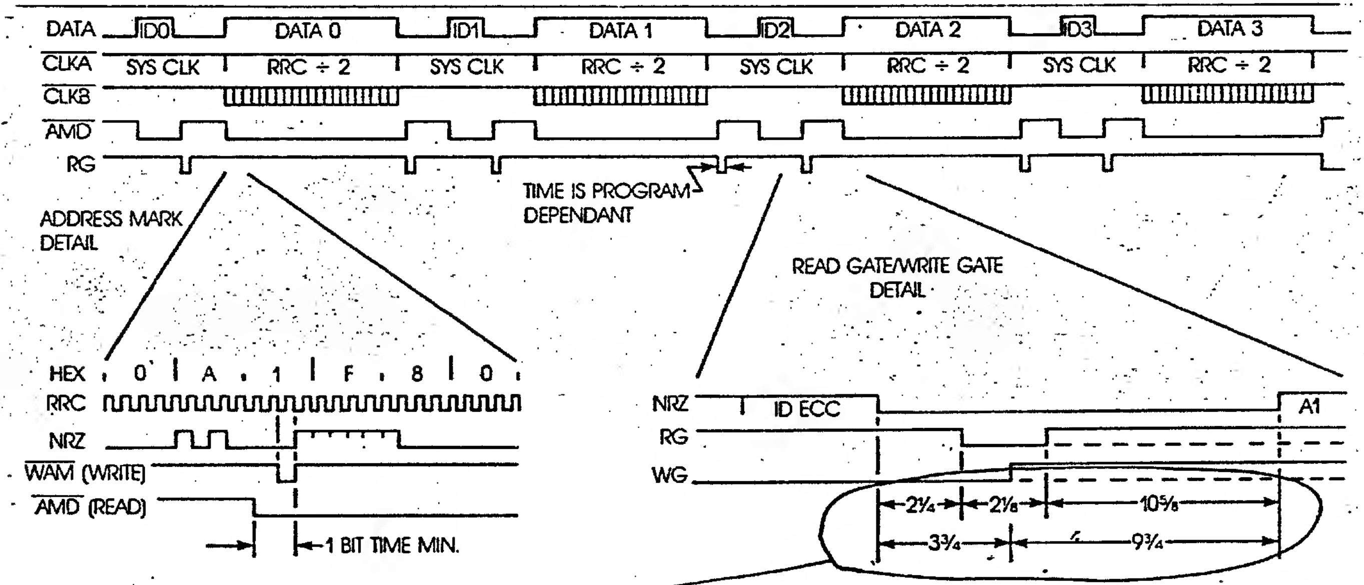
SYMBOL	PARAMETER	MIN	MAX	UNITS
F	$\overline{RD}$ or $\overline{WR}$ ↓ to 6E or 6F ↓		130	ns
R	$\overline{RD}$ or $\overline{WR}$ ↑ to 6E or 6F ↑	45	140	ns

010 .
AIC-~~100~~ SYSCLK TIMING



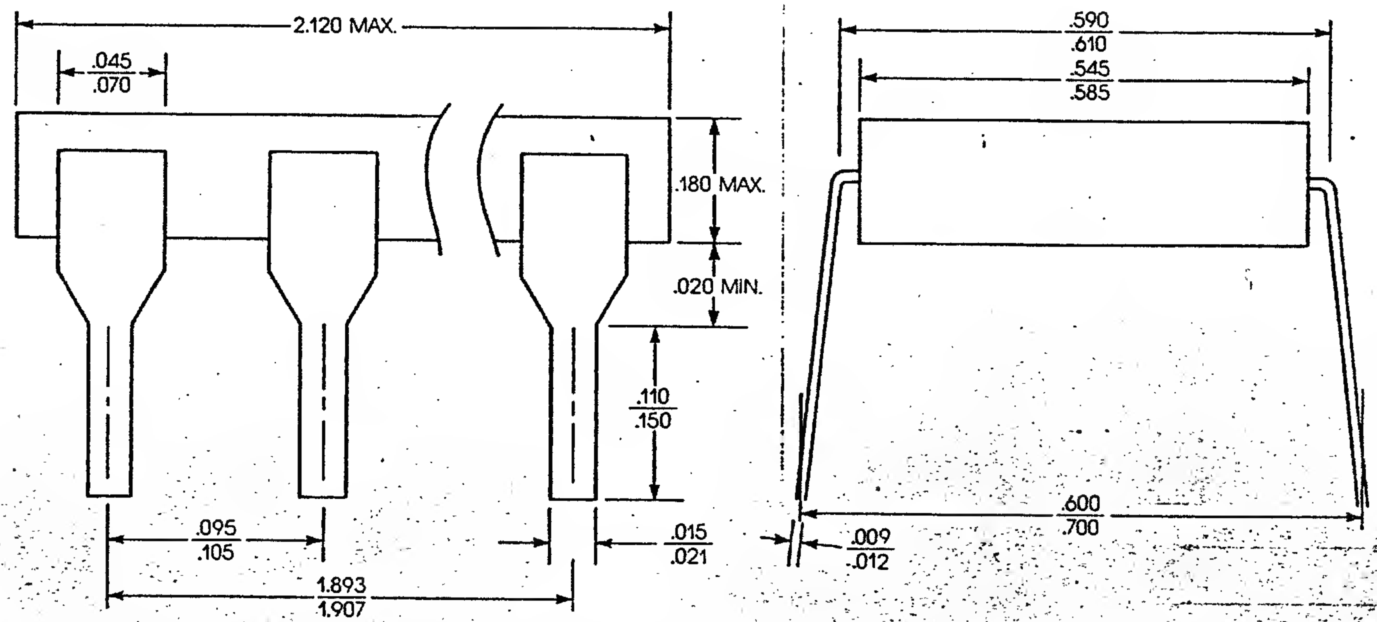
SYMBOL	PARAMETER	MIN	MAX	UNITS
S_H	SYSCLK High	60		ns
S_L	SYSCLK Low	60		ns

REFERENCE TIMING DIAGRAM



NOTE: NUMBERS ARE GIVEN IN BYTE TIMES. NOTICE THAT THE SUM OF RG TIMES IS 1.5 BYTES LONGER THAN THE SUM OF WG TIMES DUE TO 1 BYTE DATA DELAY IN THE AIC-100 AND ½ BYTE DELAY IN THE ENCODER.

PACKAGING INFORMATION



40-Lead Plastic Dual-in-line Package