iSBX 218™
FLEXIBLE DISK CONTROLLER
HARDWARE REFERENCE MANUAL

Manual Order Number: 121583-001

Copyright © 1980 Intel Corporation
Intel Corporation, 3065 Bowers Avenue, Santa Clara, California 95051

Additional copies of this manual or other Intel literature may be obtained from:

Literature Department
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

The information in this document is subject to change without notice.

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update nor to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

Intel software products are copyrighted by and shall remain the property of Intel Corporation. Use, duplication or disclosure is subject to restrictions stated in Intel's software license, or as defined in ASPR 7-104.9(a)(9).

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of Intel Corporation.

The following are trademarks of Intel Corporation and may be used only to describe Intel products:

BXP	Intellec	Multibus
i	iSBC	Multimodule
ICE	iSBX	PROMPT
iCS	Library Manager	Promware
Insite	MCS	RMX
Intel	Megachassis	UPI
Intelevision	Micromap	μScope

and the combination of ICE, iCS, iSBC, iSBX, MCS, or RMX and a numerical suffix.

PREFACE

This manual provides general information, preparation for use instructions, programming information, principles of operation and service information for the iSBX 218 Flexible Disk Controller. Additional related information is provided in the following Intel documents:

- *iSBX™ Bus Specification,* Order Number 142686
- *Intel MCS-80 User's Manual,* Order Number 9800153
- *Intel Multibus Specification,* Order Number 9800683
- *Intel 8080/8085 Assembly Language Programming Manual,* Order Number 9800301
- *Intel 8272 Single/Double Density Floppy Disk Controller,* Data Sheet

CONTENTS

CONTENTS (Continued)

TABLES

ILLUSTRATIONS

FIGURES (Continued)

CHAPTER 1
GENERAL INFORMATION

1-1. INTRODUCTION

The iSBX 218 Flexible Disk Controller Multimodule Board (hereinafter called the iSBX 218 board or controller) is a member of Intel's growing line of expansion Multimodule boards designed to augment the iSBC microcomputers, and other iSBC boards equipped with an iSBX connector.

The iSBX 218 board provides for integration of system mass storage by interfacing with up to four 8″ double sided, double density floppy diskette drives. Each drive has a storage capacity of 1.6 Megabytes of unformatted data, providing a total capacity of 6.4 Megabytes of unformatted data. This chapter contains a description of the iSBX 218 board and lists the equipment supplied, compatible equipment, and specifications.

1-2. DESCRIPTION

The iSBX 218 Controller is a double-wide Multimodule printed circuit board that interfaces with Intel iSBC boards equipped with iSBX connectors conforming to the iSBX™ Bus Specifications. The iSBX 218 board also interfaces with the diskette drives and issues appropriate commands for accessing, reading, and writing data on the diskette. Design of the board is based around the Intel 8272 Single/Double Density Floppy Disk Controller (FDC) chip. Included in the design is the logic to write pre-compensate the Modified Frequency Modulation (MFM) data, interface to the disk drives, and separate the MFM (or FM) data and clocks read from the diskettes. The controller operates exclusively on +5 Vdc power supplied via the iSBX connector. The 8272 FDC chip utilizes fifteen commands for communication between the host iSBC board and the disk drives. Other TTL logic is required to interface to the disk drives. When operating in the MFM mode, logic is provided to pre-compensate the write data before it is sent to the drive. Additionally, data separation logic takes the raw read data from the diskette drive and converts it to NRZ data.

The iSBX 218 board may be wired to generate an interrupt on request from the 8272 FDC chip, or operate in the DMA mode. Interrupt priority is determined on the host iSBC board. Jumpers are also provided to wire the iSBX 218 board for 8″ or 5¼″ diskette drives. Due to the wide variety of available 5¼″ diskette drive interfaces, the user is responsible for insuring that the specifications of a particular 5¼″ disk drive are compatible with the iSBX 218 board.

1-3. EQUIPMENT SUPPLIED

The following equipment is supplied with the iSBX 218 board:

a. Schematic Diagram, dwg. no. 162308
b. Assembly Diagram, dwg. no. 162057.
c. 6 Screws, ¼″ 6-32 nylon.
d. 3 Spacers, ½″ 6-32 nylon

1-4. COMPATIBLE EQUIPMENT

The iSBX 218 board must be used with an iSBC host board that is equipped with an iSBX connector.

The controller board cannot directly access the MULTIBUS bus structure. Multibus interfacing is provided through the P1 Multibus connector on the iSBC host board.

1-5. SPECIFICATIONS

Specifications for the iSBX 218 Flexible Disk Controller Multimodule Board are provided in Table 1-1.

Table 1-1. Specifications

I/O ADDRESSING

I/O addressing is dependent on the iSBX connector used on the iSBC host board (I/O Port Address)

POWER REQUIREMENT

+5 Volts, ±0.25 Volts, @ 895 mA (max).

INTERFACE CONNECTORS:

Interface	No. of Pins	Pin Centers		Mating Connectors
		in	mm	
P1 host board	36	0.1	2.54	Intel 103059-001
J1 diskette drive	50	0.1	2.54	3M 3425-0000

ENVIRONMENTAL REQUIREMENTS

Operation Temperature: 0°C to 55°C (32°F to 131°F)

Relative Humidity: To 90% without condensation

PHYSICAL CHARACTERISTICS

Width: 7.24 cm (2.85 inches).

Length: 19.1 cm (7.50 inches).

Height: 1.40 cm (0.56 inch) iSBX 218 board only.
2.82 cm (1.13 inches) iSBX 218 board and iSBC board.

Weight: 91 gm (3.20 ounces).

CHAPTER 2
PREPARATION FOR USE

2-1. INTRODUCTION

This chapter provides instructions for installing the iSBX 218 Controller Board onto a host microcomputer or other iSBC board equipped with an iSBX connector. Included are instructions on unpacking and inspection, installation considerations such as power, cooling, mounting, and size requirements, dc characteristics, connector pin assignments, jumper configurations, and installation procedures.

2-2. UNPACKING AND INSPECTION

Inspect the shipping carton immediately upon receipt for evidence of mishandling during transit. If the shipping carton is severely damaged or water-stained, request that the carrier's agent be present when the carton is opened. If the carrier's agent is not present when the carton is opened, and the contents are damaged, keep the carton and packing materials for the agent's inspection.

For repairs to a product damaged in shipment, contact the Intel Product Service Hotline to obtain a Return Authorization Number and further instructions. A purchase order will be required to complete the repair. A copy of the purchase order should be submitted to the carrier with your claim.

It is suggested that all salvageable shipping cartons and packing materials be retained for future use in the event the product must be shipped.

2-3. INSTALLATION CONSIDERATIONS

The iSBX 218 board is designed to interface with all Intel Single Board Computers that support the I/O interface connector required for the Multimodule boards. Other installation considerations such as power, cooling, mounting, and physical size requirements, are provided in the following paragraphs.

2-4. POWER REQUIREMENT

The power requirement for the Controller board is +5V (±0.25V) at 895 mA maximum. All power is provided to the iSBX 218 board from the host iSBC board via the P1 connector on the iSBX 218 board.

2-5. COOLING REQUIREMENT, iSBX 218

The iSBX 218 Controller dissipates 4.75 watts. Adequate circulation of air must be provided to prevent a temperature rise above +55°C (+131°F).

2-6. MOUNTING REQUIREMENT

Figure 2-1 shows the iSBX connector and spacer locations. The iSBX 218 board mounts onto an iSBC board supporting an iSBX connector and the proper mounting holes. The mounting hardware supplied as part of the iSBX board includes:

a. 3 nylon spacers, ½" threaded, separate from the board.

b. 6 nylon screws, ¼" 6-32, separate from the board.

c. One 36-pin connector (P1), factory-installed onto the board.

d. One 50-pin connector (J1), factory-installed onto the board.

NOTE

The iSBX 218 board, when installed onto a host iSBC board, occupies an additional card slot adjacent to the host iSBC board in the 604/614 card cage.

2-7. PHYSICAL DIMENSIONS

Physical dimensions of the iSBX 218 board are as follows:

a. Width: 7.24 cm (2.85 inches).

b. Length: 19.1 cm (7.50 inches).

c. Height: 1.40 cm (0.56 inches) iSBX 218 board only.
2.82 cm (1.13 inches) iSBX 218 board and host iSBC board.

Figure 2-2 shows the clearances for a Multimodule board mounted onto a host iSBC board. Height dimensions shown are maximum.

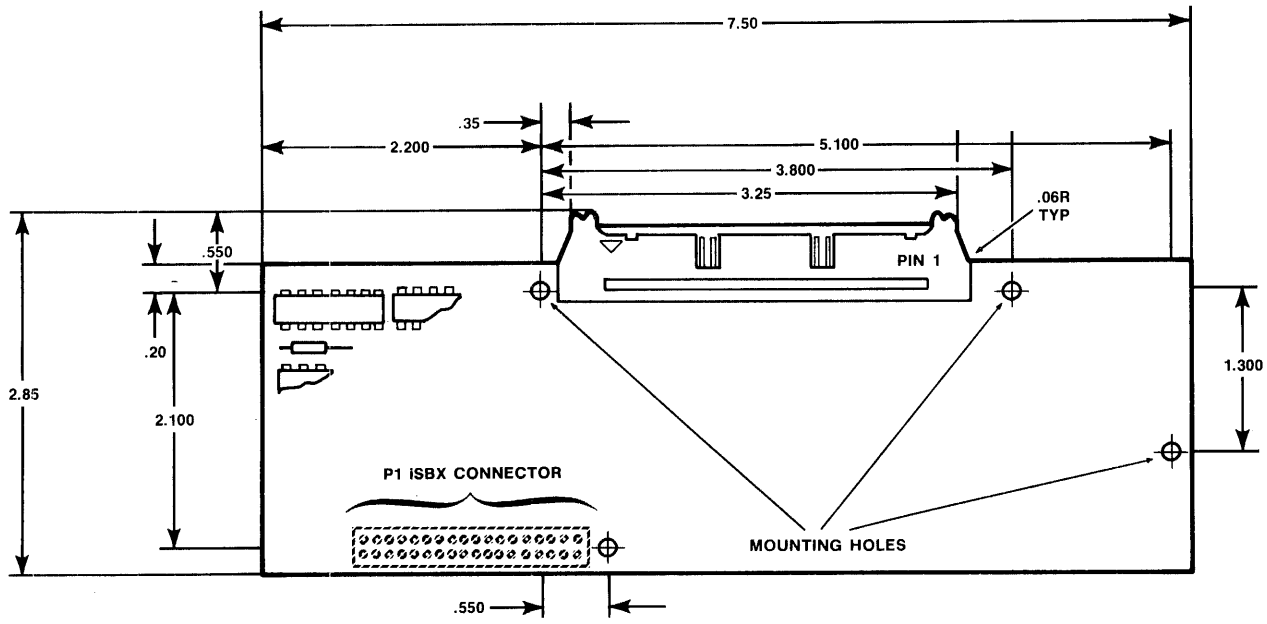


Figure 2-1. Board Dimensions (Inches)

2-8. ELECTRICAL CONSIDERATIONS

The following paragraphs define all the electrical requirements for the iSBX 218 board. First the dc requirements are listed and then the ac timing is described.

2-9. GENERAL iSBX BUS CONSIDERATIONS

Table 2-1 shows the relationship between logical and electrical states. The sample signals shown in table 2-1 are IORD/ and IORD. IORD/ is an active low signal and IORD is an active high signal.

2-10. ENVIRONMENTAL

All iSBX bus requirements should be met while the environment is within the following ranges:

Temperature: 0°C to +55°C (+32°F to 131°F) Free moving air across the host iSBC board and the iSBX 218 board.

Humidity: 90% maximum relative (non-condensing).

Shock: 30 g's of force for an 11 msec duration, 3 times in 3 planes, both sides (total of 18 drops).

Vibration: Sweeping from 10 Hz to 55 Hz and back to 10 Hz, at a distance of 0.010 inches peak-to-peak, lasting 15 minutes in each of the three planes.

2-11. CONNECTOR CONFIGURATIONS

Connectors P1 and J1 are shown in Figure 2-3. Connector P1 on the iSBX 218 board interfaces to the host iSBC board. The pin assignments for connector P1 are listed in Table 2-2. Connector J1 on the iSBX 218 board interfaces to the diskette drives, and the pin assignments for this connector are listed in Table 2-3. The signal names provided in Table 2-3, for the diskette drive interface, may vary somewhat, depending on the drive manufacturer's signal name definitions. A typical drive connector pin configuration is shown in Table 2-4.

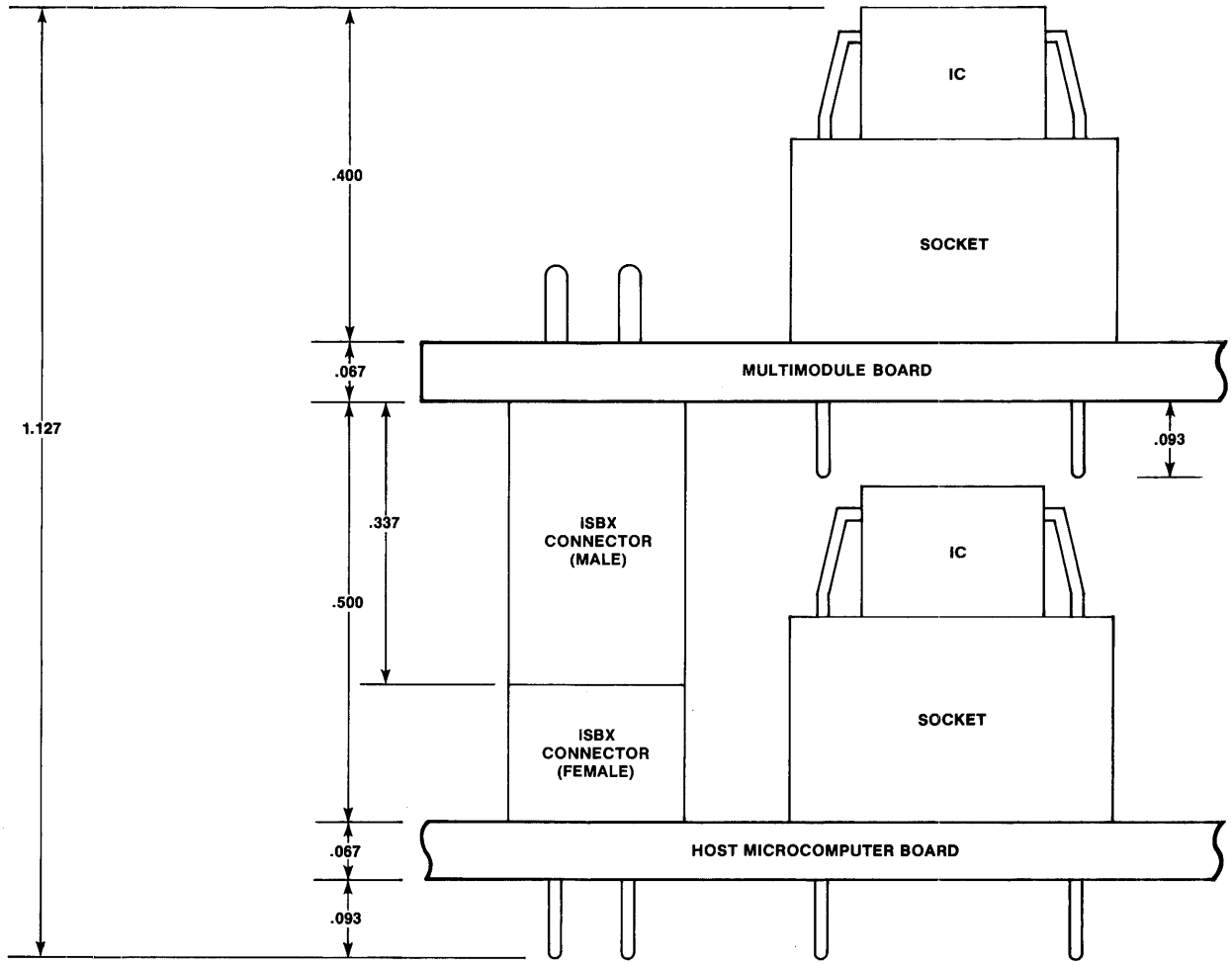


Figure 2-2. Mounting Clearances (Inches)

Table 2-1. Logical and Electrical States

Signal Name	Logical State	Electrical Signal Level	At Receiver	At Driver
IORD/	0	H = TTL High State	5.25 ≥ H ≥ 2.0V	5.25 ≥ H ≥ 2.4V
IORD/	1	L = TTL Low State	0.8 ≥ L ≥ −0.5V	0.5 ≥ L ≥ 0V
IORD	0	L = TTL Low State	0.8 ≥ L ≥ −0.5V	0.5 ≥ L ≥ 0V
IORD	1	H = TTL High State	5.25 ≥ H ≥ 2.0V	5.25 ≥ H ≥ 2.4V
V_{CC} = 5 volts ±5% referenced to logical ground. V = volts.				

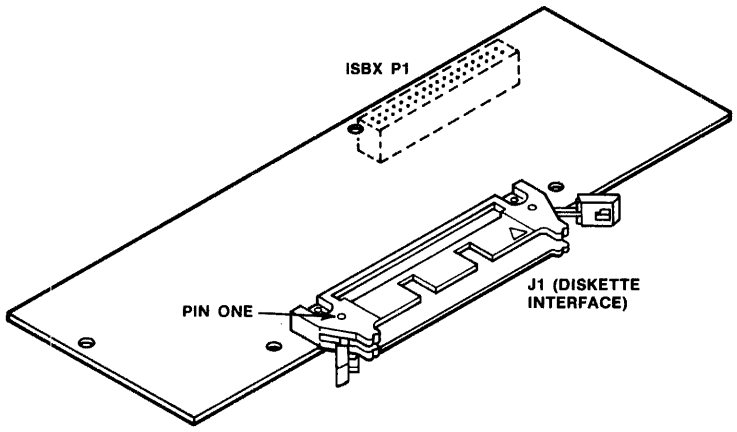


Figure 2-3. iSBX 218 Connector Locations

Table 2-2. Connector P1 Assignment

Pin	Mnemonic	Description	Pin	Mnemonic	Description
35	GND	Signal Ground	36	+5V	+5 Volts
33	MD0	MDATA Bit 0	34	MDRQT	M DMA Request
31	MD1	MDATA Bit 1	32	MDACK/	M DMA Acknowledge
29	MD2	MDATA Bit 2	30	TC	Terminal Count
27	MD3	MDATA Bit 3	28	—	Reserved
25	MD4	MDATA Bit 4	26	—	Reserved
23	MD5	MDATA Bit 5	24	—	Reserved
21	MD6	MDATA Bit 6	22	MCS0/	M Chip Select 0
19	MD7	MDATA Bit 7	20	—	Reserved
17	GND	Signal Gnd	18	+5V	+5 Volts
15	IORD/	I/O Read Cmd	16	—	Not Used
13	IOWRT/	I/O Write Cmd	14	—	Not Used
11	MA0	M Address 0	12	MINTR1	M Interrupt 1
9	—	Reserved	10	—	Reserved
7	—	Reserved	8	MPST/	M Present
5	RESET	Reset	6	—	Not Used
3	GND	Signal Gnd	4	+5V	+5 Volts
1	—	Reserved	2	—	Reserved

- MUST BE SUPPLIED BY HOST

Table 2-3. Connector J1 Assignment

Pin	Description	Pin	Description
1	Signal Ground	2	Low Current/
3	Signal Ground	4	Fault Reset/
5	Signal Ground	6	Fault/
7	Signal Ground	8	Reserved
9	Signal Ground	10	Two-Sided/
11	Signal Ground	12	Reserved
13	Signal Ground	14	Side Select
15	Signal Ground	16	Reserved
17	Signal Ground	18	Head Load/
19	Signal Ground	20	Index/
21	Signal Ground	22	Ready/
23	Signal Ground	24	Reserved
25	Signal Ground	26	Drive 0 Sel/
27	Signal Ground	28	Drive 1 Sel/
29	Signal Ground	30	Drive 2 Sel/
31	Signal Ground	32	Drive 3 Sel/
33	Signal Ground	34	Direction
35	Signal Ground	36	Step/
37	Signal Ground	38	Write Data/
39	Signal Ground	40	Write Enable/
41	Signal Ground	42	Track 00/
43	Signal Ground	44	Write Protect/
45	Signal Ground	46	Read Data/
47	Signal Ground	48	Reserved
49	Signal Ground	50	Reserved

NOTE: See Chapter 4 for detailed signal description.

CAUTION

J1 pin numbers refer to the iSBX 218 board connector only. The pin numbers on the mating connector at the diskette drive may not be the same.

2-12. DC SPECIFICATIONS

The dc specifications for the iSBX bus interface (P1) are summarized in Tables 2-5 and 2-6. Table 2-5 is the output specifications; Table 2-6 is the input specifications. The output specifications are the requirements on the output drivers of the iSBX 218 board, (i.e., the data bus output drivers must guarantee at least 1.6 mA @ 0.5 volts). The output specifications in Table 2-5 are the minimum drive requirements. The input specifications are the requirements of the receivers on the iSBX 218 board. (e.g., the loading of the address lines (MA0-MA2) can be no greater than 0.5 mA @ 0.8 volts). These tables also summarize the maximum loading permitted on an iSBX Multimodule board interface at any one time.

2-13. TIMING

Table 2-7 summarizes all the ac timing requirements for an iSBX Multimodule board I/O interface. The timing diagrams are shown in Figures 2-4 through 2-7. The MWAIT/ signal is not used on the iSBX 218 board.

Table 2-4. Typical Diskette Drive Interface Signal Pinout

Drive Connector Pin Number	Signal Name
1-49 odd	Ground
2	Write Current Switch
NC	
NC	
NC	
10	Two-sided
12	Disk Change
14	Side Select
16	In Use
18	Head Load
20	Index
22	Ready
24	Sector
26	Drive Select 1
28	Drive Select 2
30	Drive Select 3
32	Drive Select 4
34	Direction Select
36	Step
38	Write Data
40	Write Gate
42	Track 00
44	Write Protect
46	Read Data
48	FM Sep Data
50	FM Sep Clock

NOTE: The above pin numbers are for a typical OEM diskette drives including Shugart Associates SA850, Pertec FDX50, MFE700, Memorex 552, and other industry drives.

Table 2-5. Output Specifications (P1)

Bus Signal Name	Type[1] Drive	I_{OL} Max −Min (mA)	@ Volts (V_{OL} Max)	I_{OH} Max −Min (μA)	@ Volts (V_{OH} Min)	C_O (Min) (pf)
MD0-MD7	TRI	1.6	0.5	−200	2.4	130
MINTR0-1	TTL	2.0	0.5	−100	2.4	40
MDRQT	TTL	1.6	0.5	− 50	2.4	40
OPT1-2	TTL	1.6	0.5	− 50	2.4	40
MPST/	TTL	Note 2				

1. TTL = standard totem pole output. TRI = Tri-state.
2. iSBX MMIO board must connect this signal to ground.

Table 2-6. Input Specifications (P1)

Bus Signal Name	Type[1] Receiver	I_{IL} Max (mA)	@ V_{IN} Max (volts)	I_{IH} Max (μA)	@ V_{IN} Max (volts)	C_I Max (pf)
MD0-MD7	TRI	−0.5	0.8	70	2.0	40
MA0-MA2	TTL	−0.5	0.8	70	2.0	40
MCS0/-MCS1/	TTL	−4.0	0.8	100	2.0	40
MRESET	TTL	−2.1	0.8	100	2.0	40
MDACK/	TTL	−1.0	0.8	100	2.0	40
IORD/ IOWRT/	TTL	−1.0	0.8	100	2.0	40
OPT1-OPT2	TTL	−2.0	0.8	100	2.0	40

NOTE: 1. TTL = standard totem pole output. TRI = Three-state.

NOTE

The input waveforms for the ac timing specifications are as follows:

2-14. JUMPER CONFIGURATION

The iSBX 218 board contains several jumpers that permit the user to select the board operation for a particular diskette drive or data transfer option. Two jumpers (W2 and W8) are installed for testing only and should never be removed. Jumper W1 can be connected between "A" and "C" for non-DMA operation, or between "A" and "B" for DMA operation.

Jumpers W3 through W7 must be configured for the type of diskette drive to be used, i.e., 8" or 5¼". Table 2-8 shows how to connect these jumpers to enable the iSBX 218 board to operate with the desired drive type. The iSBX 218 board cannot be configured to operate with two different types of diskette drives at the same time. For actual physical location of jumpers, see Figure 5-1.

2-15. I/O CABLING

The user-supplied cabling for the iSBX 218 board provides the interface between the Controller and the diskette drives. This cable is connected to iSBX 218 connector J1 with a mating connector 3M Part Number 3425-0000. This connector is a 50-pin ribbon cable, mass-terminated connector, whose connection configuration is shown in Figure 2-8. The drive end of this interface cable must be configured to match the specific diskette drive being used. The interface cables between the drives are configured as described in each of the manufacturer's user manuals.

Table 2-7. iSBX Bus™ Multimodule Board I/O AC Requirements

Symbol	Parameter	Min (ns)	Max (ns)	Figure Reference
t_1	Address stable before read	50	—	2-4
t_2	Address stable after read	30	—	2-4
t_3	Read pulse width	300	—	2-4
t_4[2]	Data valid from read	0	250	2-4
t_5[2]	Data float after read	0	150	2-4
t_6[3]	Time between RD and/or WRT	—	—	—
t_7	CS stable before CMD	25	—	2-4
t_8	CS stable after CMD	30	—	2-4
t_9	Power up reset pulse width	50 Msec	—	2-6
t_{10}	Address stable before WRT	50	—	2-3
t_{11}	Address stable after WRT	30	—	2-3
t_{12}[1]	Write pulse width	300	—	2-3
t_{13}[1]	Data valid to write	250	—	2-3
t_{14}	Data valid after write	30	—	2-3
t_{15}	MCLK cycle	100	110	2-6
t_{16}	MCLK width	35	65	2-6
t_{17}[1]	MWAIT/ pulse width	0	4 msec	2-3, 2-4
t_{18}	Reset pulse width	50 Msec	—	2-6
t_{19}	MCS/ to MWAIT/ valid	0	75	2-3, 2-4
t_{20}	DACK set up to I/O CMD	100	—	2-5
t_{21}	DACK hold	30	—	2-5
t_{22}	CMD to DMA RQT removed to end of DMA cycle	—	200	2-5
t_{23}	TDMA pulse width	500	—	2-5
t_{24}[1]	MWAIT/ to valid read data	—	0	2-4
t_{25}[1]	MWAIT/ to WRT CMD	0	—	2-3

NOTES:
1. Required only if WAIT is activated.
2. If MWAIT/ not activated.
3. Time dependent on the host iSBC board to which the Multimodule board is connected.

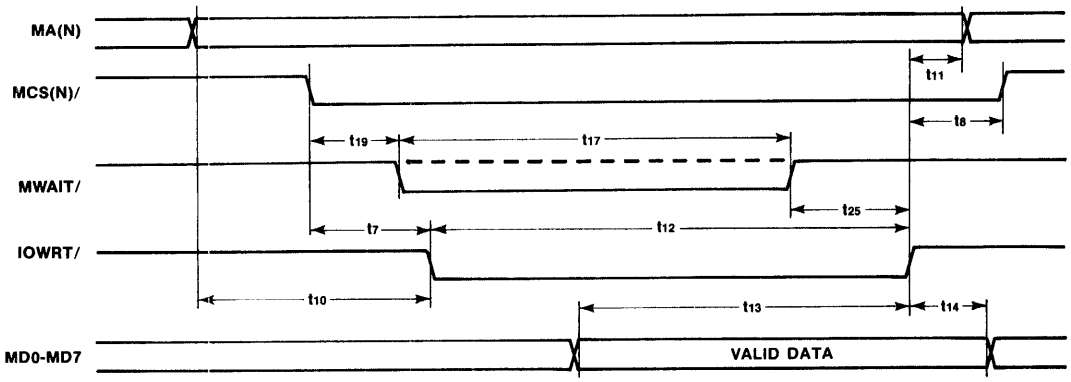


Figure 2-4. iSBX Multimodule Board I/O Write Timing

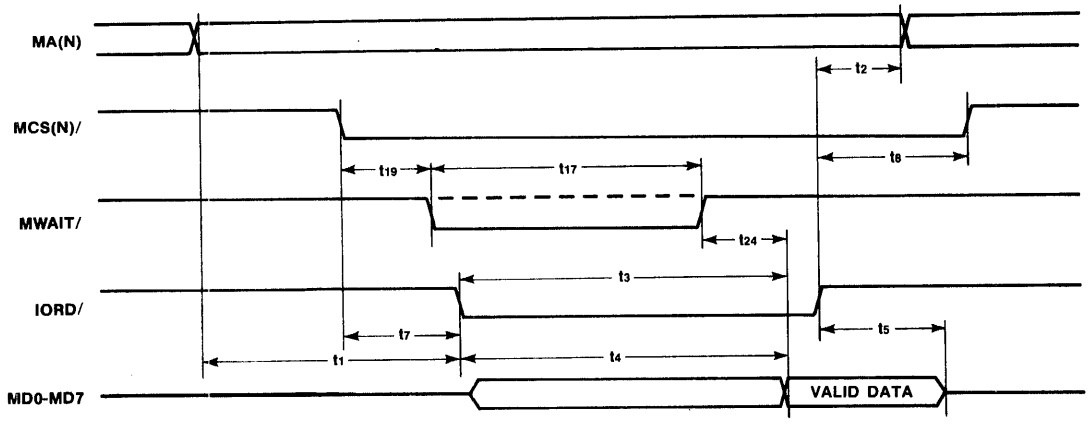


Figure 2-5. iSBX Multimodule Board I/O Read Timing

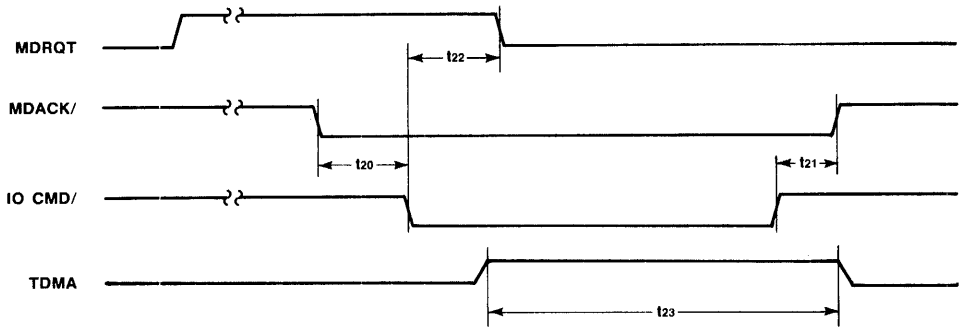


Figure 2-6. iSBX Multimodule Board I/O DMA Timing

Table 2-8. Diskette Select Jumpers

Jumper	For 8" Diskette	*For 5¼" Diskette
W3	A-C	A-B
W4	A-C	A-B
W5	A-C	A-B
W6	A-C	A-B
W7	A-C	A-B

*The user is responsible for insuring that the specifications of a particular 5¼" disk drive are compatible with the iSBX 218 board.

2-16. TERMINATION

All line terminations required on the iSBX 218 board are provided through the 150 ohm resistors in resistor pack RP2. This is a factory installed device and is not field removable. Termination requirements for the diskette drive are listed in the drive manufacturer's user manual. The last drive in the daisy chain string must be terminated. A typical four diskette drive installation, showing cabling and termination location is shown in Figure 2-9. More detailed information on drive installation is provided in the user manual for the specific drive employed.

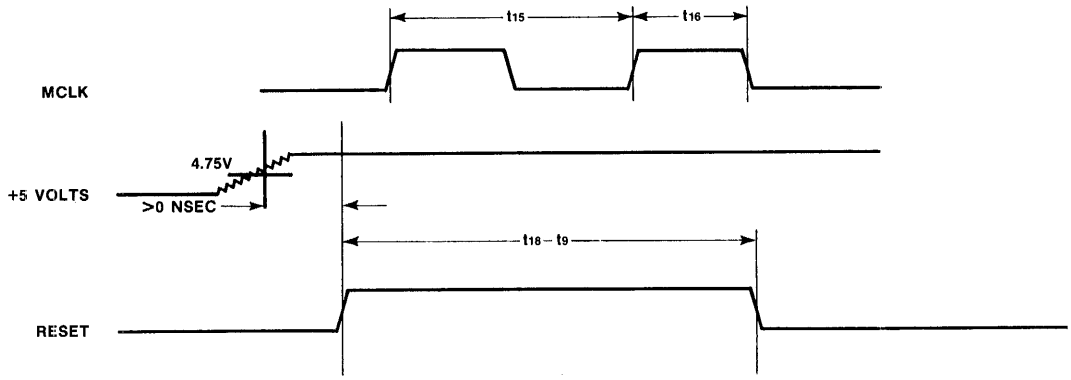


Figure 2-7. iSBX Multimodule Board Reset Timing

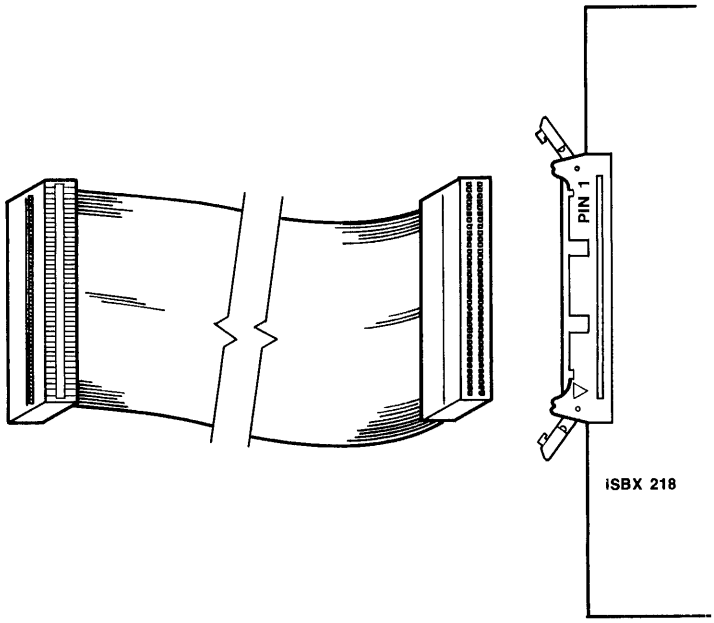


Figure 2-8. Diskette Drive Cable Configuration

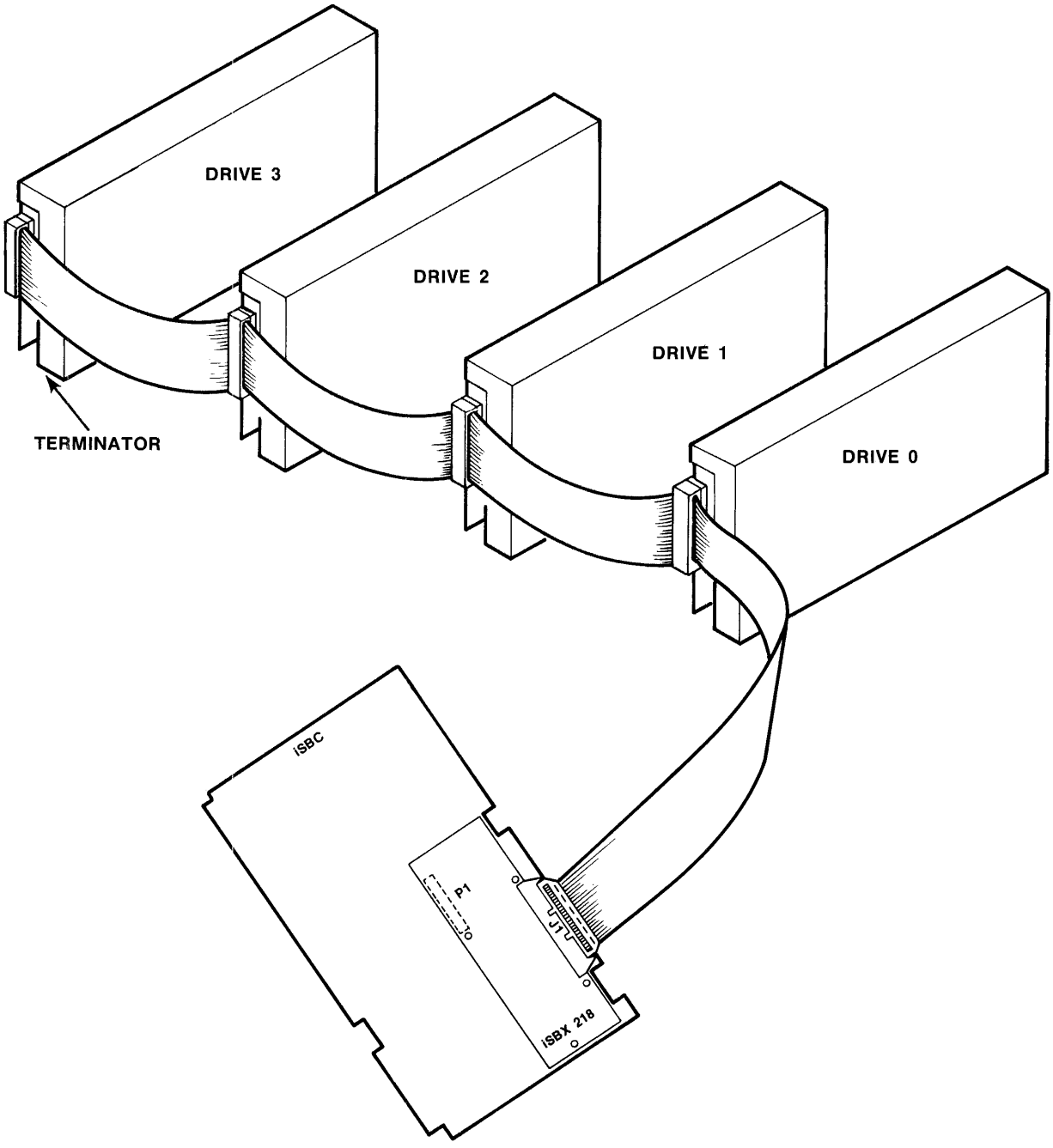


Figure 2-9. Typical Diskette Drive Installation

2-17. INSTALLATION PROCEDURE

The iSBX 218 board mounts onto the host iSBC board. Install the board as follows:

a. With the 6-32 screws, secure the three ½ inch plastic spacers to the host iSBC board as shown in Figure 2-10.

b. Locate pin 1 on the iSBX connector (P1) of the Multimodule board and align it with pin 1 of the iSBX connector on the host iSBC board.

c. Align the iSBX 218 board mounting holes with the spacers on the host iSBC board. See Figure 2-1 for hole location.

d. Gently press the two boards together until the connector seats.

e. Fasten the iSBX 218 board to the spacer with the three remaining 6-32 nylon screws.

NOTE

The position of an installed Multimodule board and the host board connector number may vary according to the type of host iSBC board that is used.

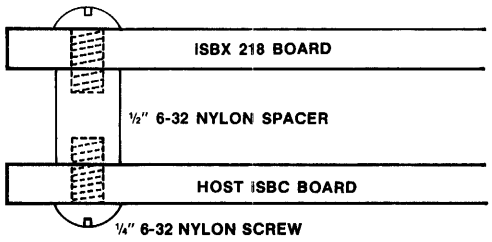


Figure 2-10. Mounting Technique

CHAPTER 3
PROGRAMMING INFORMATION

3-1. INTRODUCTION

This chapter provides programming information for the iSBX 218 Flexible Disk controller board. Included is information on I/O addressing, system initialization, and 8272 FDC programming. The logic and design of the iSBX 218 board is based around the 8272 FDC chip. The 8272 FDC chip is the only programmable device on the controller.

This chapter provides information to invoke the total command set from the base board to the iSBX 218 board permitting the use of 8″ single/double sided, single/double density floppy disk drives or 5¼″ mini-flexible disk drives. Included in the programming information is the complete status register bit definitions, DMA operation, and interrupt handling.

The 8272 FDC chip command set and its implementation are explained in the following text. Each host iSBC board must meet the requirements to implement this command set. It is therefore necessary that each iSBC board be configured with the appropriate PROMs containing the required firmware to meet these requirements.

3-2. 8272 FDC OPERATION

The 8272 is an LSI Floppy Disk Controller (FDC) chip, which contains the circuitry and control functions for interfacing the host iSBC microprocessor board with up to four floppy disk drives. It is capable of supporting either IBM 3740 single density format (FM), or IBM System 34 double density format (MFM) including double sided recording.

Address mark detection circuitry is internal to the FDC, thus simplifying the Phase Locked Loop (PLL) and read electronics. The track stepping rate, head load time, and head unload time may be programmed by the user. The FDC also permits multiple sector transfers in both read and write modes with a single instruction.

Signals provided by the 8272 FDC permit the iSBX 218 board to operate in the Direct Memory Access (DMA) mode or the Non-DMA mode. In the Non-DMA mode, the FDC generates interrupts to the processor to transfer each data byte. In the DMA mode, the processor initializes the DMA controller on the host board to perform all data transfers and then loads a command into the FDC. Figure 3-1 shows a typical configuration using the host iSBC board with an Intel 8085 microprocessor, 8237 DMA controller, and the associated busses. The iSBX 218 board must be configured with jumpers for either the DMA or Non-DMA mode operation, refer to paragraph 2-14. When using DMA mode the iSBX 218 requires the Terminal Count (TC) input signal.

3-3. I/O ADDRESSING

The microprocessor on the host iSBC board communicates with the programmable 8272 FDC chip through a sequence of I/O Read and I/O Write Commands, via the iSBX connector (P1). The address of the iSBX I/O connector is determined by the host iSBC board configuration. There are three basic communication functions that occur between the host and the 8272 FDC chip and they are: read main status register; read from data register; and write into data register. Each function is specified by the state of A_0, $\overline{RD}$, and $\overline{WR}$ shown in table 3-1.

Table 3-1. Status/Data Register Selection

A_0	$\overline{RD}$	$\overline{WR}$	FUNCTION
0	0	1	Read Main Status Register
0	1	0	Illegal
0	0	0	Illegal
1	0	0	Illegal
1	0	1	Read from Data Register
1	1	0	Write into Data Register

3-4. 8272 COMMAND SET

There are 15 separate commands that comprise the command set for the Intel 8272 FDC. These commands are loaded into the iSBX 218 board through connector P1. Each of these commands require multiple bytes to fully specify the operation that the FDC is to implement. The command set is as follows:

Read Data
Read ID
Read Deleted Data
Read a Track
Scan Equal
Scan High or Equal
Scan Low or Equal
Specify
Write Data
Format a Track
Write Deleted Data
Seek
Recalibrate (Restore to Track 0)
Sense Interrupt Status
Sense Drive Status

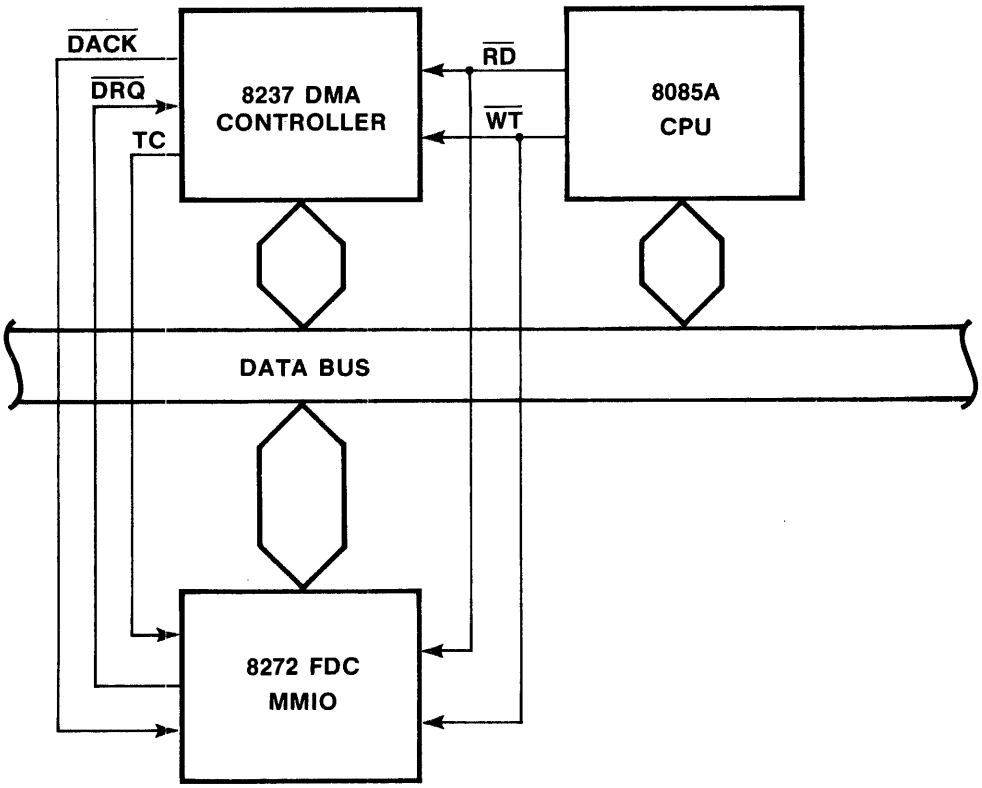


Figure 3-1. 8272/DMA Mode Block Diagram

Typically, these commands are issued in short sequences of seek (to the desired track), followed by either a read, write, scan or format command, and finished by a sense command. Scan commands allow searching to take place in parallel with the host processor activity. Sense commands provide status of command operations. Specify commands provide initial 8272 timer information. Recalibrate moves the read/write head to track 0. Details of these commands are provided in following paragraphs of this chapter.

3-5. 8272 REGISTERS — MAIN STATUS/ DATA REGISTERS

The 8272 contains two registers that are accessed by the host iSBC processor; a Status Register and a Data Register. The main status register can only be read by the host iSBC processor and is used to facilitate the transfer of data between the 8272 and processor. The 8-bit Main Status Register contains the status information of the FDC, and can be accessed at any time. Data bytes are written into the Data Register to program commands into the 8272. Data bytes are also read out of the data register to obtain results after execution of a command. The 8-bit Data Register consists of several registers in a stack. Only one of these registers is presented to the data bus at a time. The data registers store data, commands, parameters, and Floppy Disk Drive (FDD) status information. The relationship between the main Status/Data registers and the signals $\overline{RD}$, $\overline{WR}$, and A_0 is shown in table 3-1.

3-6. STATUS REGISTER

The Main Status Register bits are defined and listed in table 3-2. The DIO and RQM bits in the Status Register indicate when data is ready and in which direction it will be transferred on the data bus. Figure 3-2 shows the Status Register timing. Status Register bits for ST0 through ST3 are defined and listed in table 3-3. ST0-ST3 are status data registers which can only be accessed in sequence after a command completion.

Each of the 15 different commands executed by the 8272 FDC chip is initiated by multi-byte transfer

from the host processor, and the result, after execution, may also be a multi-byte transfer to the host processor. Each command is broken down into the following three phases:

a. Command Phase: The FDC receives all information required to perform a particular operation from the host iSBC processor.

b. Execution Phase: The FDC performs the operation it was instructed to do.

c. Result Phase: After completion of the operation, status and other housekeeping information are made available to the host iSBC processor.

During the command and result phases, the main status register must be read by the host CPU until DB7, Request for Master, is in a 1 state. This ensures that the 8272 is ready to accept or provide the next command or result byte before each byte of information is written into or read from the data register.

Bits DB6 and DB7 are in a 0 and 1 state, respectively, before each command word byte is written into the 8272. Many of the commands require multiple bytes, the main status register is read prior to each byte transfer to the 8272 FDC chip. During the result phase, DB6 and DB7 in the main status register are both 1's before reading each byte from the data register. Reading the main status register before each data byte transfer is required only in the command and result phases.

During the execution phase, in the Non-DMA mode, the receipt of each data byte (when reading data from the disk) is indicated by the interrupt signal (INT). Activation of the read signal $\overline{RD}$ by the host CPU, resets the interrupt and loads the data register contents onto the data bus. When performing a write command, the $\overline{WR}$ signal resets the interrupt signal. If the processor cannot handle interrupts fast enough (every 13μs for M/M), it may poll the main status register where bit 7 (RQM) functions as the interrupt signal.

Table 3-2. Main Status Register Bits

Bit Number	Name	Symbol	Description
DB_0	FDD 0 Busy	D_0B	FDD number 0 is in the Seek mode.
DB_1	FDD 1 Busy	D_1B	FDD number 1 is in the Seek mode.
DB_2	FDD 2 Busy	D_2B	FDD number 2 is in the Seek mode.
DB_3	FDD 3 Busy	D_3B	FDD number 3 is in the Seek mode.
DB_4	FDC Busy	CB	A read or write * command is in process.
DB_5	Non-DMA mode	NDM	The FDC is in the non-DMA mode. This bit is set only during the execution phase in non-DMA mode. Transition to "0" state indicates execution phase has ended.
DB_6	Data Input/Output 0 = OUTPUT 1 = INPUT	DIO	Indicates direction of data transfer between FDC and Data Register. If DIO = "1" then transfer is from Data Register to the Processor. If DIO = "0", then transfer is from the Processor to Data Register.
DB_7	Request for Master	RQM	Indicates Data Register is ready to send or receive data to or from the Processor. Both bits DIO and RQM should be used to perform the handshaking functions of "ready" and "direction" to the processor.

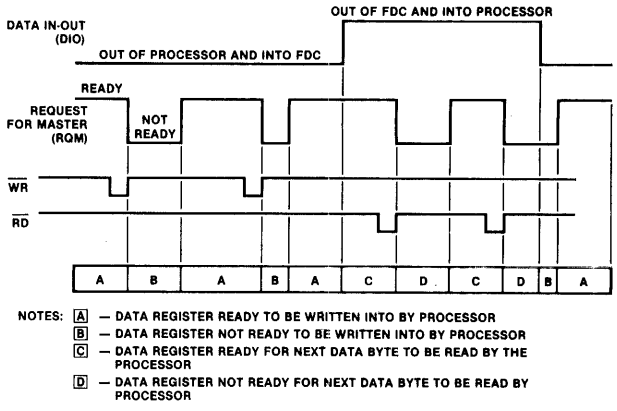


Figure 3-2. Status Register Timing

BITS 0→3 WILL STAY HIGH UNTIL SENSE INTERRUPT COMMAND IS EXECUTED.

* IN THE PROCESS OF READING RESULT BYTES OR WRITING PARAMETERS.

Table 3-3. Status Register

BIT No.	BIT Name	BIT Symbol	DESCRIPTION
STATUS REGISTER 0			
D_7	Interrupt Code	IC	D_7 = 0 and D_6 = 0 Normal Termination of Command. (NT). Command was completed and properly executed.
D_6			D_7 = 0 and D_6 = 1 Abnormal Termination of Command, (AT). Execution of Command was started, but was not successfully completed.
			D_7 = 1 and D_6 =0 Invalid Command issue. (IC). Command which was issued was never started.
			D_7 = 1 and D_6 =1 Abnormal Termination because during command execution the ready signal from FDD changed state.
D_5	Seek End	SE	When the FDC completes the SEEK Command, this flag is set to 1 (high).
D_4	Equipment Check	EC	If a fault Signal is received from the FDD, or if the Track 0 Signal fails to occur after 77 Step Pulses (Recalibrate Command) then this flag is set.
D_3	Not Ready	NR	When the FDD is in the not-ready state and a read or write command is issued, this flag is set. If a read or write command is issued to Side 1 of a single sided drive, then this flag is set.
D_2	Head Address	HD	This flag is used to indicate the state of the head at Interrupt.
D_1	Unit Select 1	US 1	These flags are used to indicate a Drive
D_0	Unit Select 0	US 0	Unit Number at Interrupt
STATUS REGISTER 1			
D_7	End of Cylinder	EN	When the FDC tries to access a Sector beyond the final Sector of a Cylinder, this flag is set.
D_6			Not used. This bit it always 0 (low).
D_5	Data Error	DE	When the FDC detects a CRC error in either the ID field or the data field, this flag is set.
D_4	Over Run	OR	If the FDC is not serviced by the main-systems during data transfers, within a certain time interval, this flag is set.
D_3			Not used. This bit always 0 (low).
D_2	No Data	ND	During execution of READ DATA, WRITE DELETED DATA or SCAN Command, if the FDC cannot find the Sector specified in the IDR Register, this flag is set.
			During executing the READ ID Command, if the FDC cannot read the ID field without an error, then this flag is set.
			During the execution of the READ A Cylinder Command, if the starting sector cannot be found, then this flag is set.
D_1	Not Writable	NW	During execution of WRITE DATA, WRITE DELETED DATA or Format A Cylinder Command, if the FDC detects a write protect signal from the FDD, then this flag is set.
D_0	Missing Address Mark	MA	If the FDC cannot detect the ID Address Mark after encountering the index hole twice, then this flag is set.
			If the FDC cannot detect the Data Address Mark or Deleted Data Address Mark, this flag is set. Also at the same time, the MD (Missing Address Mark in Data Field) of Status Register 2 is set.
STATUS REGISTER 2			
D_7			Not used. This bit is always 0 (low).
D_6	Control Mark	CM	During executing the READ DATA or SCAN Command, if the FDC encounters a Sector which contains a Deleted Data Address Mark, this flag is set.
D_5	Data Error in Data Field	DD	If the FDC detects a CRC error in the data field then this flag is set.
D_4	Wrong Cylinder	WC	This bit is related with the ND bit, and when the contents of C on the medium is different from that stored in the IDR, this flag is set.
D_3	Scan Equal Hit	SH	During execution, the SCAN Command, if the condition of "equal" is satisfied, this flag is set.
D_2	Scan Not Satisfied	SN	During executing the SCAN Command, if the FDC cannot find a Sector on the cylinder which meets the condition, then this flag is set.
D_1	Bad Cylinder	BC	This bit is related with the ND bit, and when the content of C on the medium is different from that stored in the IDR and the content of C is FF, then this flag is set.
D_0	Missing Address Mark in Data Field	MD	When data is read from the medium, if the FDC cannot find a Data Address Mark, or Deleted Data Address Mark, then this flag is set.
STATUS REGISTER 3			
D_7	Fault	FT	This bit is used to indicate the status of the Fault signal from the FDD.
D_6	Write Protected	WP	This bit is used to indicate the status of the Write Protected signal from the FDD.
D_5	Ready	RDY	This bit is used to indicate the status of the Ready signal from the FDD.
D_4	Track 0	T0	This bit is used to indicate the status of the Track 0 signal from the FDD.
D_3	Two Sided	TS	This bit is used to indicate the status of the Two Side signal from the FDD.
D_2	Head Address	HD	This bit is used to indicate the status of Side Select signal to the FDD.
D_1	Unit Select 1	US 1	This bit is used to indicate the status of the Unit Select 1 signal to the FDD.
D_0	Unit Select 0	US 0	This bit is used to indicate the status of the Unit Select 0 signal to the FDD.

During the execution phase, when performing a read in the DMA mode, interrupts are not generated for each data byte transfer. The 8272 generates DMA requests when each byte of data is available. The DMA controller responds with a DMA acknowledge and the $\overline{RD}$ signal. When the DMA acknowledge goes low, the DMA request is reset. When execution is completed (Terminal Count), the interrupt signal activates, indicating the beginning of the result phase. Reading of the first byte of data in the result phase (ST0) automatically resets the interrupt signal in the DMA mode.

During the result phase all bytes shown in the result part of each command must be read. The Read Data Command, for example, has seven bytes of data in the result phase. All seven bytes must be read to complete the Read Data Command. The 8272 FDC chip will not accept a new command until the previous command is completed.

The 8272 FDC chip contains five status registers. The main status register, table 3-2, mentioned previously, can be accessed by the host processor at any time. The other four status registers (ST0, ST1, ST2, ST3), table 3-3, are only available during the result phase, and are read as data from the Data Register only at the completion of the command. The number of status registers read is determined by the particular command executed.

The data bytes sent to the 8272 FDC chip comprising the command phase are read out during the result phase. The writing and reading of these data bytes is performed in the order shown for each command. After the last data byte in the command phase is written to the 8272 FDC chip, the execution phase starts automatically. When the last data byte is read out in the result phase, the command is automatically ended and the 8272 FDC chip is ready for a new command. A command may be aborted at any time by activating the Terminal Count signal. The host iSBC board *must* activate the Terminal Count (TC) signal when the proper number of drive data bytes have been written or read for a particular command.

3-7. COMMAND DESCRIPTIONS

During the command phase, the main status register must be polled by the host processor before each byte is written into the data register. The DIO (DB6) and RQM (DB7) bits must be in the 0 and 1 states respectively, before each byte of the command is written into the 8272 FDC chip. The start of the execution phase of any command causes DIO and RQM to switch to 1 and 0 states respectively. The mnemonics for the byte designations for both the command and result phase of the command set are listed in table 3-4. Figure 3-3 is provided as a reference for typical track format definitions and mnemonics used in the following command descriptions. The values of N, SC, and GPL for the various sector sizes are defined in Table 3-7. Paragraph 3-16 covers the specification of Head Load Time (HLT), Head Unload Time (HUT), and Step Rate Time (SRT) used in the execution of other commands.

3-8. READ DATA

The Read Data command shown in figure 3-4 requires nine bytes to load the command and the command data. After execution, the host reads seven bytes to complete the result phase of the command. After the Read Data command is issued the FDC loads the heads (if they are not loaded), waits the specified head settling time and begins reading ID Address Marks and ID fields. When the current sector number (R=byte four of the command) stored in the ID register compares with the sector number read from the diskette, the FDC transfers data (from the data field) byte-by-byte to the data bus. The Read/Write head must have been previously positioned to the desired track by a Seek command.

After completion of the read operation from the current sector, the Sector Number is incremented by one, and the data from the next sector is read and set onto the data bus. This action constitutes a Multi-sector read operation. The read command is terminated by receipt of the Terminal Count signal. Activation of Terminal Count causes the FDC to stop outputting data onto the data bus, but it continues to read data from the current sector, check CRC bytes (Cyclic Redundancy Count), and terminates the command at the end of the current sector read.

SEE P. 4-3

The amount of data transferred with a single command to the FDC depends on the MT-bit, MFM-bit (byte 0 of the read command), and N-byte (byte five of the read command). The 'N' byte determines the number of bytes per sector. Table 3-5 lists the different transfer capacities, persuant to the different bit configurations.

The Multi-track function (MT) allows the FDC to read data from both sides of the diskette. For a particular cylinder, the data transfer starts at Sector R, Side 0 and ends at the last sector on Side 1. This function pertains to only one cylinder (the same track on each side of the diskette).

STARTS AT "R" ON 2ND SIDE ALSO.

When N=0, the Data length (DTL), (byte nine of the command), defines the number of bytes the FDC considers a sector. If the DTL is smaller than the actual data length in the sector, the extra data in the sector is not sent to the data bus. The FDC reads (internally) the complete sector and performs the CRC check. Depending on the manner of command

termination, a Multi-sector read operation may be performed. When N=1, the DTL has no meaning and should be set to 0FFH.

At the completion of the read data command, the read/write head remains loaded until the specified Head Unload Time (HUT) has elapsed. If the host processor issues another command before the head unloads, the head settling time can be ignored between subsequent read operations.

If the FDC detects the Index Hole twice without finding the right sector, the FDC sets the No Data (ND) flag in status register 1 and terminates the read data command.

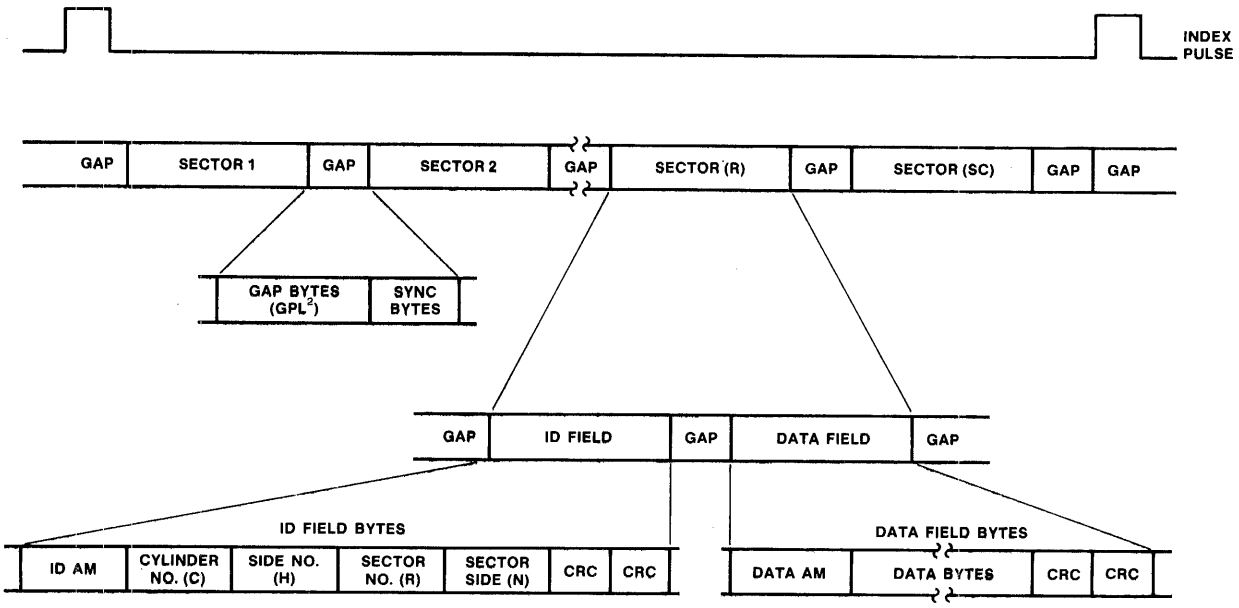
After reading the ID and Data fields in each sector, the FDC checks the CRC bytes. If a read error is detected in the ID field, the FDC sets the Data Error (DE) flag in status register 1. If a CRC error occurs in the data field, the FDC also sets the Data Error in the Data Field (DD) flag in status register 2, and terminates the command.

If the FDC reads a Deleted Data Address Mark from the diskette, and SK=0, the FDC sets the Control Mark (CM) flag in status register 2. The command is terminated after the complete sector is read. If SK=1, the FDC skips the sector with the Deleted Data Address Mark and reads the next sector.

Table 3-4. Command Mnemonics

SYMBOL	NAME	DESCRIPTION
A_0	Address Line 0	A_0 controls selection of Main Status Register (A_0 = 0) or Data Register (A_0 = 1).
C	Cylinder Number	C stands for the current selected Cylinder track number 0 through 76 of the medium.
D	Data	D stands for the data pattern which is going to be written into a Sector.
D_7-D_0	Data Bus	8-bit Data Bus where D_7 is the most significant bit, and D_0 is the least significant bit.
DS0, DS1	Drive Select	DS stands for a selected drive number 0 or 1.
DTL	Data Length	When N is defined as 00, DTL stands for the data length which users are going to read out or write into the Sector.
EOT	End of Track	EOT stands for the final Sector number of a ~~Cylinder~~ TRACK.
GPL	Gap Length	GPL stands for the length of Gap 3 (spacing between Sectors excluding VCO Sync Field).
H	Head Address	H stands for head number 0 or 1, as specified in ID field.
HDS	Head Select	HDS stands for a selected head number 0 or 1 (H = HDS in all command words).
HLT	Head Load Time	HLT stands for the head load time in the FDD (2 to 254 ms in 2 ms increments).
HUT	Head Unload Time	HUT stands for the head unload time after a read or write operation has occurred (16 to 240 ms in 16 ms increments).
MFM	FM or MFM Mode	If MF is low, FM mode is selected and if it is high, MFM mode is selected.
MT	Multi-Track	If MT is high, a multi-track operation is to be performed (a cylinder under both HD0 and HD1 will be read or written).

SYMBOL	NAME	DESCRIPTION
N	Number	N stands for the number of data bytes written in a Sector.
NCN	New Cylinder Number	NCN stands for a new Cylinder number, which is going to be reached as a result of the Seek operation. Desired position of Head.
ND	Non-DMA Mode	ND stands for operation in the Non-DMA Mode.
PCN	Present Cylinder Number	PCN stands for the Cylinder number at the completion of SENSE INTERRUPT STATUS Command. Position of Head at present time.
R	Record	R stands for the Sector number, which will be read or written.
R/W	Read/Write	R/W stands for either Read (R) or Write (W) signal.
SC	Sector	SC indicates the number of Sectors per Cylinder.
SK	Skip	SK stands for Skip Deleted Data Address Mark.
SRT	Step Rate Time	SRT stands for the Stepping Rate for the FDD (1 to 16 ms in 1 ms increments). Stepping Rate applies to all drives (F = 1 ms, E = 2 ms, etc.).
ST 0 ST 1 ST 2 ST 3	Status 0 Status 1 Status 2 Status 3	ST 0-3 stands for one of four registers which store the status information after a command has been executed. This information is available during the result phase after command execution. These registers should not be confused with the main status register (selected by A_0 = 0). ST 0-3 may be read only after a command has been executed and contain information relevant to that particular command.
STP		During a Scan operation, if STP = 1, the data in contiguous sectors is compared byte by byte with data sent from the processor (or DMA), and if STP = 2, then alternate sectors are read and compared.



NOTES: ID AM AND DATA AM BYTES ARE USED AS READ SYNC-UP MARKERS
CRC BYTES ARE CYCLIC REDUNDANCY COUNT BYTES USED FOR ERROR CHECKING

Figure 3-3. Generalized Diskette Track Format

During diskette data transfers between the FDC and the host processor, via the data bus, the FDC must be serviced by the processor every 27 μs in the FM Mode, or every 13 μs in the MFM Mode. If the host processor fails to service the FDC as prescribed, the FDC sets the Over Run (OR) flag in status register 1 and terminates the command.

If the host processor terminates a read or write operation in the FDC, the ID information in the result phase is dependent upon the status of the MT bit and EOT byte. Table 3-6 shows the values for C, H, R, and N, when the processor terminates the command.

3-9. WRITE DATA

The Write Data Command, shown in figure 3-4, requires nine bytes to load the command and the command data. After execution, the host processor reads seven bytes to complete the result phase of the command. After the Write Data command is issued, the FDC loads the heads (if they are not loaded), waits the specified head settling time, and begins reading ID Fields. The FDC reads the ID field of each sector and checks the CRC bytes. When the current sector number (R=byte four of the command) stored in the ID register compares with the sector number read off the diskette, the processor transfers data to the FDC processor byte-by-byte, via the data bus, and the FDC sends it to the FDD in a serial fashion. If the FDC detects a read error in one of the ID Fields, it sets the DE flag of Status Register 1 and terminates the write data command.

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	REMARKS
READ DATA										
Command	W	MT	MFM	SK	0	0	1	1	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	C								Sector ID information prior to Command execution
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	DTL								
Execution										Data transfer between the FDD and main-system
Result	R	ST 0								Status information after Command execution
	R	ST 1								
	R	ST 2								
	R	C								Sector ID information after Command execution
	R	H								
	R	R								
	R	N								

Figure 3-4. Read Data Command

Table 3-5. Transfer Capacity

Multi-Track MT	MFM/FM MFM	Bytes/Sector N	Maximum Transfer Capacity (Bytes/Sector) (Number of Sectors)	Final Sector Read from Diskette
0	0	00	(128) (26) = 3,328	26 at Side 0
0	1	01	(256) (26) = 6,656	or 26 at Side 1
1	0	00	(128) (52) = 6,656	26 at Side 1
1	1	01	(256) (52) = 13,312	
0	0	01	(256) (15) = 3,840	15 at Side 0
0	1	02	(512) (15) = 7,680	or 15 at Side 1
1	0	01	(256) (30) = 7,680	15 at Side 1
1	1	02	(512) (30) = 15,360	
0	0	02	(512) (8) = 4,096	8 at Side 0
0	1	03	(1024) (8) = 8,192	or 8 at Side 1
1	0	02	(512) (16) = 8,192	8 at Side 1
1	1	03	(1024) (16) = 16,384	

NOTE: The user is responsible for determining 5¼" diskette track capacities per the drive vendors specifications.

Table 3-6. ID Information When Processor Terminates Command

MT	EOT	Final Sector Transferred to Processor	ID Information at Result Phase C	H	R	N
0	1A	Sector 1 to 25 at Side 0	NC	NC	R + 1	NC
	0F	Sector 1 to 14 at Side 0				
	08	Sector 1 to 7 at Side 0				
	1A	Sector 26 at Side 0	C + 1	NC	R = 01	NC
	0F	Sector 15 at Side 0				
	08	Sector 8 at Side 0				
	1A	Sector 1 to 25 at Side 1	NC	NC	R + 1	NC
	0F	Sector 1 to 14 at Side 1				
	08	Sector 1 to 7 at Side 1				
	1A	Sector 26 at Side 1	C + 1	NC	R = 01	NC
	0F	Sector 15 at Side 1				
	08	Sector 8 at Side 1				
1	1A	Sector 1 to 25 at Side 0	NC	NC	R + 1	NC
	0F	Sector 1 to 14 at Side 0				
	08	Sector 1 to 7 at Side 0				
	1A	Sector 26 at Side 0	NC	LSB	R = 01	NC
	0F	Sector 15 at Side 0				
	08	Sector 8 at Side 0				
	1A	Sector 1 to 25 at Side 1	NC	NC	R + 1	NC
	0F	Sector 1 to 14 at Side 1				
	08	Sector 1 to 7 at Side 1				
	1A	Sector 26 at Side 1	C + 1	LSB	R = 01	NC
	0F	Sector 15 at Side 1				
	08	Sector 8 at Side 1				

Notes: 1. NC (No Change): The same value as the one at the beginning of command execution.

2. LSB (Least Significant Bit): The least significant bit of H is complemented.

After writing data into the current sector, the Sector Number stored is incremented by one, and the next data field is written. The FDC continues this Multi-Sector Write Operation until Terminal Count occurs. When the Terminal Count signal activates while writing is in process, the FDC continues writing zeroes into the current sector to complete the data field. If the Terminal Count signal is received while data is being written in the data field and the data is terminated, the FDC fills the remainder of the data field with zeros.

The Write Data command implementation is similar to the Read Data command with the following items the same. Refer to the Read Data command (paragraph 3-7) for a detailed description.

a. Transfer Capacity

b. EN (End of Cylinder) Flag

c. ND (No Data) Flag

d. Head Unload Time Interval

e. ID Information when the processor terminates command (Table 3-6)

f. Definition of DTL when N=0 and when N≠0

During the write data mode, data transfers between the host processor and the FDC must occur every 31 μs in the FM mode, and every 15 μs in the MFM mode. If these time requirements are not met, the FDC sets the OR flag in Status Register 1, and terminates the write data command after filling the remainder of the data field with indeterminate data bytes.

3-10. WRITE DELETED DATA

The Write Deleted Data Command, shown in figure 3-6, requires nine bytes to load the command and the command data. After execution, the host processor reads seven bytes to complete the result phase of the command. This command operates in the same manner as the Write Data command, except a Deleted Data Address Mark is written at the beginning of the Data Field instead of the normal Data Address Mark.

PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
WRITE DATA										
Command	W	MT	MFM	0	0	0	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	C								Sector ID information prior to Command execution
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	DTL								
Execution										Data transfer between the main system and FDD
Result	R	ST 0								Status information after Command execution
	R	ST 1								
	R	ST 2								
	R	C								Sector ID information after Command execution
	R	H								
	R	R								
	R	N								

Figure 3-5. Write Data Command

PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	REMARKS
WRITE DELETED DATA										
Command	W	MT	MFM	0	0	1	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	C								Sector ID information prior to Command execution
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	DTL								
Execution										Data transfer between the FDD and main-system
Result	R	ST 0								Status information after Command execution
	R	ST 1								
	R	ST 2								
	R	C								Sector ID information after Command execution
	R	H								
	R	R								
	R	N								

Figure 3-6. Write Deleted Data Command

3-11. READ DELETED DATA

The Read Deleted Data Command, shown in figure 3-7, requires nine bytes to load the command and the command data. After execution, the host processor reads seven bytes to complete the result phase of the command. This command operates in the same manner as the Read Data Command, with the exception of Data Address Mark detection. When the Data Address Mark is detected at the beginning of a data field and SK=0; the FDC reads all the data in the sector, and sets the CM flag in Status Register 2, and terminates the command. If SK=1 and the Data Address Mark is detected, the FDC skips the sector with the Data Address Mark and reads the next sector.

3-12. READ A TRACK

The Read A Track Command, shown in figure 3-8, requires nine bytes to load the command and the command data. After execution, the host reads seven bytes to complete the result phase of the command. This command operates similar to the Read Data command, with the exception that all of the data fields contained in each sector of the selected track are read contiguously. After detection of the Index Hole, the FDC reads all the data fields on the track as continuous blocks of data. If an ID or Data CRC check error occurs during the execution of this command, the FDC ignores the CRC check error and continues to read data from the track. The FDC compares the ID information read from each sector with the value stored in the IDR, and sets the ND flag of Status Register 1 if there is no comparison. Multi-track or skip operations are not permitted with this command.

3-13. READ ID

The Read ID Command, shown in figure 3-9, requires two bytes to load the command and the command data. After execution, the host processor reads seven bytes to complete the result phase of the command. The Read ID command is used to provide the present position of the read/write heads. The FDC stores the values from the first ID field read. If no proper ID Address Mark is found on the diskette before the Index Hole is sensed for the second time, the Missing Address Mark (MA) flag in Status Register 1 is set. If no data is found, the No Data (ND) flag is also set in Status Register 1, and the command is terminated.

3-14. FORMAT A TRACK

The Format A Track Command, shown in figure 3-10, requires six bytes to load the command and the command data. After execution, the host processor reads seven bytes to complete the result phase of the command. This command causes an entire track to be formatted. After the Index Hole is detected, Data is written on the Diskette. Gaps, Address Marks, ID Fields and Data Fields, all per the IBM System 34 (Double Density) or System 3740 (Single Density) format are recorded. The particular format to be written is controlled by the values programmed into N, Sectors/Cylinder (SC), Gap Length (GPL), and Data Pattern (D) which are supplied by the processor during the command phase. The ID Field for each sector is also supplied by the processor; that is four data requests per sector are made by the FDC for Cylinder Number (C), Head Number (H), Sector Number (R), and Number of bytes/sector (N). This enables the diskette to be formatted with non-sequential sector numbers, if desired. The data field is filled by continually writing the byte stored in D.

During formatting, after each sector, the processor must send new values for C, H, R, and N to the 8272 FDC chip for the next sector of the track. The value of the R register is incremented by one after each sector is formatted, thus, the R register contains a value of R+1 when it is read during the result phase. This incrementing and formatting continues for the whole track until the FDC encounters the Index Hole for the second time. This results in command termination. The read/write head must have been previously positioned to the desired track number by a seek command.

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	REMARKS
READ DELETED DATA										
Command	W	MT	MFM	SK	0	1	1	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	C								Sector ID information prior to Command execution
	W	H								
	W	R								
	W	N								
	W	EC			1					
	W	GPL								
	W	DTL								
Execution										Data transfer between the FDD and main-system
Result	R	ST 0								Status information after Command execution
	R	ST 1								
	R	ST 2								
	R	C								Sector ID information after Command execution
	R	H								
	R	R								
	R	N								

Figure 3-7. Read Deleted Data Command

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	REMARKS
READ A TRACK										
Command	W	0	MFM	SK	0	0	0	1	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	C								Sector ID information prior to Command execution
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	DTL								
Execution										Data transfer between the FDD and main-system. FDC reads all of cylinders contents from index hole to EOT
Result	R	ST 0								Status information after Command execution
	R	ST 1								
	R	ST 2								
	R	C								Sector ID information after Command execution
	R	H								
	R	R								
	R	N								

Figure 3-8. Read A Track Command

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	REMARKS
READ ID										
Command	W	0	MFM	0	0	1	0	1	0	Commands
	W	0	0	0	0	0	HDS	DS1	DS0	
Execution										The first correct ID information on the Cylinder is stored in Data Register
Result	R	ST 0								Status information after Command execution
	R	ST 1								
	R	ST 2								
	R	C								Sector ID information during Execution Phase
	R	H								
	R	R								
	R	N								

Figure 3-9. Read ID Command

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	REMARKS
FORMAT A TRACK										
Command	W	0	MFM	0	0	1	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	N								Bytes/Sector
	W	SC								Sectors/Track
	W	GPL								Gap 3
	W	D								Filter Byte
Execution										FDC formats an entire cylinder
Result	R	ST 0								Status information after Command execution
	R	ST 1								
	R	ST 2								
	R	C								In this case, the ID information has no meaning
	R	H								
	R	R								
	R	N								

Figure 3-10. Format A Track Command

If the Fault signal is received from the drive during a write operation, the FDC sets the EC flag of Status Register 0 and terminates the command, after setting bits 6 and 7 of Status Register 0 to 1 and 0 respectively. The loss of the Ready signal at the beginning or during the execution phase causes command termination after bits 6 and 7 of Status Register are both set to 1. The relationship between N, SC, and GPL for various sector sizes is shown in table 3-7.

3-15. SCANS

There are three Scan Commands, shown in figures 3-11 through 3-13, each requiring nine bytes to load the command and the command data. After execution, the host processor reads seven bytes to complete the result phase of the command. The three Scan Commands are Scan Equal, Scan Low or Equal, and Scan High or Equal. The basic function of these commands is to compare a sector of data from the diskette to a specific stream of data from the host on a byte-by-byte basis.

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	REMARKS
SCAN EQUAL										
Command	W	MT	MFM	SK	1	0	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	C								Sector ID information prior to Command execution
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	STP								
Execution										Data compared between the FDD and main-system
Result	R	ST 0								Status information after Command execution
	R	ST 1								
	R	ST 2								
	R	C								Sector ID information after Command execution
	R	H								
	R	R								
	R	N								

Figure 3-11. Scan Equal Command

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	REMARKS
SCAN HIGH OR EQUAL										
Command	W	MT	MFM	SK	1	1	1	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	C								Sector ID information prior to Command execution
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	STP								
Execution										Data compared between the FDD and main-system
Result	R	ST 0								Status information after Command execution
	R	ST 1								
	R	ST 2								
	R	C								Sector ID information after Command execution
	R	H								
	R	R								
	R	N								

Figure 3-12. Scan High or Equal Command

Table 3-7. Sector Size Relationships

FORMAT	SECTOR SIZE	N	SC	GPL[1]	GPL[2]	REMARKS
FM	128 bytes/Sector	00	$1A_{(16)}$	$07_{(16)}$	$1B_{(16)}$	IBM Diskette 1
	256	01	$0F_{(16)}$	$0F_{(16)}$	$2A_{(16)}$	IBM Diskette 2
	512	02	08	$1B_{(16)}$	$3A_{(16)}$	
FM Mode	1024 bytes/Sector	03	04	—	—	
	2048	04	02	—	—	
	4096	05	01	—	—	
MFM Mode	256	01	$1A_{(16)}$	$0E_{(16)}$	$36_{(16)}$	IBM Diskette 2D
	512	02	$0F_{(16)}$	$1B_{(16)}$	$54_{(16)}$	
	1024	03	08	$35_{(16)}$	$74_{(16)}$	IBM Diskette 2D
	2048	04	04	—	—	
	4096	05	02	—	—	
	8192	06	01	—	—	

Note: 1. Suggested values of GPL in Read or Write Commands to avoid splice point between data field and ID field of contiguous sections.

2. Suggested values of GPL in format command.

3. The user is responsible for determining proper gap size and number of sectors for specific 5¼" diskette drives.

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	REMARKS
SCAN LOW OR EQUAL										
Command	W	MT	MFM	SK	1	1	0	0	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	C								Sector ID information prior to Command execution
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	STP								
Execution										Data compared between the main system and FDD
Result	R	ST 0								Status information after Command execution
	R	ST 1								
	R	ST 2								
	R	C								Sector ID information after Command execution
	R	H								
	R	R								
	R	N								

Figure 3-13. Scan Low or Equal Command

To satisfy the Scan Equal command, the value of the data read from the diskette must equal the value of the data sent over to be compared.

To satisfy the Scan Low or Equal command, the value of the data read from the diskette must be equal to or less than the value of the data sent over to be compared.

To satisfy the Scan High or Equal command, the value of the data read from the diskette must be equal to or greater than the value of the data sent over to be compared.

After an entire sector of data is compared, and the desired condition is not met, the sector number is incremented (R+STP→R), and the scan operation continues (STP defined in Table 3-4). The scan operation continues until the requirements to satisfy the specific scan command are met, the last sector on the track is read (EOT), or terminal count occurs.

When the scan equal conditions are met, the FDC sets the Scan Equal Hit (SH) flag of Status Register 2, and terminates the command. If the conditions are not met between the starting sector (as specified by R) and the last sector on the cylinder (EOT), the FDC sets the Scan Not Satisfied (SN) flag of Status Register 2, and terminates the command. The receipt of the Terminal Count (TC) signal from the processor or DMA controller during the execution phase causes the FDC to complete the comparison of the particular byte in process, and then terminate the command. Table 3-8 shows the status of the SH and SN bits under various conditions of the Scan command.

Table 3-8. Scan Status Codes

COMMAND	STATUS REGISTER 2		COMMENTS
	BIT 2 = SN	BIT 3 = SH	
Scan Equal	0	1	$D_{FDD} = D_{Processor}$
	1	0	$D_{FDD} \neq D_{Processor}$
Scan Low or Equal	0	1	$D_{FDD} = D_{Processor}$
	0	0	$D_{FDD} < D_{Processor}$
	1	0	$D_{FDD} \not< D_{Processor}$
Scan High or Equal	0	1	$D_{FDD} = D_{Processor}$
	0	0	$D_{FDD} > D_{Processor}$
	1	0	$D_{FDD} \not> D_{Processor}$

If, during a scan operation, the FDC encounters a deleted data address mark on any sector (and SK=0), it regards that sector as the last sector of the cylinder, sets the Control Mark (CM) flag in Status Register 2, and terminates the command. If SK=1 when this condition occurs, the FDC skips the sector with the deleted address mark, and reads the next sector. In the latter case however, the FDC still sets the CM flag in the Status Register 2 in order to show that a deleted sector has been encountered.

During execution of the Scan Command, data is supplied by either the processor or DMA Controller for comparison against data read from the diskette. To prevent the OR flag from being set in Status Register 1, the data must be available every 27 μs (FM Mode) or 13 μs (MFM Mode). If overrun occurs, the FDC terminates the command.

3-16. SPECIFY

The Specify Command, shown in figure 3-14, requires three bytes to load the command and command data. Since this command just loads information into the 8272 FDC chip for future commands, there is no execution or result phase. The Specify Command sets the initial values for each of the three internal timers. The Head Unload Time (HUT) defines the time from the end of the execution phase of a Read/Write Command to the head unload state. This timer is programmable from 16 to 240 ms in 16 ms increments as shown in table 3-9. The Step Rate Time (SRT) defines the time interval between adjacent step pulses sent to the FDD from the 8272. This timer is programmable from 1 to 16 ms in 1 ms increments, as shown in table 3-10. The SRT must be set to 1 ms greater than the minimum desired step interval time. The Head Load Time (HLT) defines the time between activation of the Head Load signal and initiation of a Read or Write operation. This timer is programmable from 2 to 254 ms in increments of 2 ms, as shown in table 3-11.

The time intervals mentioned are a direct function of the clock frequency. Times indicated are for an 8 MHz clock. If the clock is reduced to 4 MHz (Mini-floppy application), then all time intervals are increased by a factor of 2.

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_{b1}	D_0	REMARKS
SPECIFY										
Command	W	0	0	0	0	0	0	1	1	Command Codes
	W	SRT							HUT	
	W	HLT							ND	

Figure 3-14. Specify Command

Table 3-9. Head Unload Time Programming Table

D_3	D_2	D_1	D_0	HEX	HUT TIME
0	0	0	1	1	16MS
0	0	1	0	2	32 MS
0	0	1	1	3	48 MS
0	1	0	0	4	64 MS
0	1	0	1	5	80 MS
0	1	1	0	6	96 MS
0	1	1	1	7	112 MS
1	0	0	0	8	128 MS
1	0	0	1	9	144 MS
1	0	1	0	A	160 MS
1	0	1	1	B	176 MS
1	1	0	0	C	192 MS
1	1	0	1	D	208 MS
1	1	1	0	E	224 MS
1	1	1	1	F	240 MS

Table 3-10. Step Rate Time Programming Table

D_7	D_6	D_5	D_4	HEX	*SRT TIME
1	1	1	1	F	1 MS
1	1	1	0	E	2 MS
1	1	0	1	D	3 MS
1	1	0	0	c	4 MS
1	0	1	1	B	5 MS
1	0	1	0	A	6 MS
1	0	0	1	9	7 MS
1	0	0	0	8	8 MS
0	1	1	1	7	9 MS
0	1	1	0	6	10 MS
0	1	0	1	5	11 MS
0	1	0	0	4	12 MS
0	0	1	1	3	13 MS
0	0	1	0	2	14 MS
0	0	0	1	1	15 MS
0	0	0	0	0	16 MS

*The SRT should be set to 1 MS greater than the desired step internal time.

Table 3-11. Head Load Time Programming Table

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	HEX	HLT TIME
0	0	0	0	0	0	1	ND	02	2 MS
0	0	0	0	0	1	0	ND	04	4 MS
0	0	0	0	0	1	1	ND	06	6 MS
0	0	0	0	1	0	0	ND	08	8 MS
1	1	1	0	1	0	0	ND	E8	232 MS
1	1	1	0	1	0	1	ND	EA	234 MS
1	1	1	0	1	1	0	ND	EC	236 MS
1	1	1	0	1	1	1	ND	EE	238 MS
1	1	1	1	0	0	0	ND	F0	240 MS
1	1	1	1	0	0	1	ND	F2	242 MS
1	1	1	1	0	1	0	ND	F4	244 MS
1	1	1	1	0	1	1	ND	F6	246 MS
1	1	1	1	1	0	0	ND	F8	248 MS
1	1	1	1	1	0	1	ND	FA	250 MS
1	1	1	1	1	1	0	ND	FC	252 MS
1	1	1	1	1	1	1	ND	FE	254 MS

The DMA or Non-DMA selection is made by the Non-DMA (ND) bit. When this bit is set, the Non-DMA mode is selected, and when this bit is reset (ND=0) the DMA mode is selected.

3-17. SEEK

The Seek Command, shown in figure 3-15, requires three bytes to load the command and command data. The Read/Write heads in the FDD are moved from cylinder to cylinder under the control of the seek command. The FDC compares the Present Cylinder Number (PCN), which is the current head position, with the New Cylinder Number (NCN) and performs the following operation when there is a difference.

PCN<NCN: Direction Signal is set to a 1 and step pulses are sent to the FDD.

PCN>NCN: Direction Signal is set to a 0 and step pulses are sent to the FDD.

The Step Pulse rate is controlled by the SRT in the Specify Command. After each step pulse is issued, NCN is compared against PCN, and when NCN=PCN, the SE flag is set in Status Register 0, and the command is terminated. At the termination of the Seek command, the interrupt line to the host is activated, and the host performs a Sense Interrupt Status Command.

During the command phase, the FDC is in the FDC Busy state, but during the execution phase it is in the Non-Busy state. When the FDC is not Busy, another seek command may be issued to another drive. In this manner, parallel seek operations may be done on up to four drives at once.

If the FDD Ready signal is not active at the beginning of the command execution phase or goes inactive during the seek operation, the NR flag is set in Status Register 0 and the command is terminated.

3-18. RECALIBRATE

The Recalibrate Command, shown in figure 3-16, requires two bytes to load the command and command data. This command causes the Read/Write heads in the FDD to be positioned over track 0. The FDC sets the contents of the PCN counter to 0, and checks for the activation of the Track 0 signal from the FDD. The Direction signal remains active and stepping pulses continue until the Track 0 signal goes high. When the Track 0 signal goes high, the SE flag in Status Register 0 is set and the command is terminated. If the Track 0 signal does not go high and 77 Step Pulses have been issued, the FDC sets the SE flag and Equipment Check (EC) flags of Status Register 0, and terminates the command. At the termination of the Recalibrate command, the interrupt line to the host is activated, and the host performs a Sense Interrupt Status Command.

The ability to overlap Recalibrate commands to multiple Drives, and the loss of the Ready signal, as described in the Seek Command, applies to this command.

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	REMARKS
SEEK										
Command	W	0	0	0	0	1	1	1	1	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	NCN								
Execution										Head is positioned over proper Cylinder on Diskette

Figure 3-15. Seek Command

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	REMARKS
RECALIBRATE										
Command	W	0	0	0	0	0	1	1	1	Command Codes
	W	0	0	0	0	0	0	DS1	DS0	
Execution										Head retracted to Track 0

Figure 3-16. Recalibrate Command

3-19. SENSE INTERRUPT STATUS

The Sense Interrupt Status Command, shown in figure 3-17, requires one byte to load the command. Since there is no execution phase, the host FDC goes directly to the result phase, and the host reads two bytes words of status information. The host normally issues the Sense Interrupt Status command as a result of receiving an unexpected Interrupt signal from the FDC. The interrupt signal is generated by the FDC for any of the following reasons.

1. When in the Result Phase of:
   a. Read Data Command
   b. Read a Track Command
   c. Read ID Command
   d. Read Deleted Data Command
   e. Write Data Command
   f. Format a Cylinder Command
   g. Write Deleted Data Command
   h. Scan Commands
2. Ready Signal from the drive changes state.
3. Seek End occurs or completion of the Recalibrate command.
4. During the execution phase in the Non-DMA Mode.

Interrupts caused by reasons 1 and 4 occur during normal command operations and are easily recognized by the processor. However, other interrupts are uniquely identified with the aid of the Sense Interrupt Status Command. This command resets the interrupt signal, and by reading bits 5, 6, and 7 of Status Register 0 identifies the cause of the interrupt, as shown in table 3-12.

Neither the Seek or Recalibrate Commands have a Result phase, therefore, it is mandatory to use the Sense Interrupt Status Command after these commands to effectively terminate them, and to provide verification of the head position (PCN).

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	REMARKS
SENSE INTERRUPT STATUS										
Command	W	0	0	0	0	1	0	0	0	Command Codes
Result	R	ST 0								Status information at the end of each seek operation about the FDC
	R	PCN								

Figure 3-17. Sense Interrupt Status Command

3-20. SENSE DRIVE STATUS

The Sense Drive Status Command, shown in figure 3-18, requires two bytes to load the command. After loading the command, the host processor reads one byte to complete the result phase of the command. This command is used by the host to interrogate the FDC regarding the status of the FDDs. Status Register 3 contains the Drive Status information.

Table 3-12. Seek, Interrupt Codes

SEEK END	INTERRUPT CODE		CAUSE
BIT 5	BIT 6	BIT 7	
0	1	1	Ready Line changed state, either polarity
1	0	0	Normal Termination of Seek or Recalibrate Command
1	1	0	Abnormal Termination of Seek or Recalibrate Command

3-21. INVALID COMMANDS

An Invalid Command, shown in figure 3-19, is recognized whenever the command code issued by the host is not one of the 8272 FDC command set. When the condition occurs, the FDC terminates the command. No interrupt is generated by the FDC during this condition. The DIO and RQM bits are set in the Main Status Register, indicating that the FDC is in the Result Phase and the contents of Status Register 0 must be read. When the processor reads Status Register 0, it will find an 80H, indicating an invalid command was received.

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	REMARKS
SENSE DRIVE STATUS										
Command	W	0	0	0	0	0	1	0	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
Result	R	ST 3								Status information about FDD

Figure 3-18. Sense Drive Status Command

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	REMARKS
INVALID										
Command	W	Invalid Codes								Invalid Command Codes (NoOp-FDC goes into Stand-by State)
Result	R	ST 0								ST 0 = 80 (16)

Figure 3-19. Invalid Command

SENSE INTERRUPT DOES NOT INDICATE WHETHER DRIVE IS NOT READY AFTER SEEK.

As previously stated for the Seek and Recalibrate commands, a Sense Interrupt Status Command is issued by the processor at the termination of these commands. If the processor does not follow this protocol, the next command received will be decoded as invalid. An Invalid command can be used as a No-Op command, to place the FDC in a standby or no operation state.

3-22. SOFTWARE

The software required for the 8272 FDC chip consists of Input/Output drivers whose four major functions are to initialize the FDC chip at power on, issue commands to the FDC chip, handle completion interrupts from the FDC chip, and control the special function port. Appendix A provides examples of I/O drivers to allow immediate use of the 8272 FDC chip as written; or as experience and understanding are gained, the I/O drivers may be restructured or completely rewritten to better suit the particular application. The assembly listing is provided in Appendix A to enable the reader to better understand this section, and as a set of sample I/O driver routines.

NOTE

Some iSBC boards with iSBC 218 capabilities may be too slow to use interrupts. On these boards a polling method must be used.

Level 1 subroutines are primitives which:

a. Are user callable.

b. Use all the 8080/8085 registers

c. Return immediately if the 8272 FDC chip is busy

Using these level 1 subroutines, it is possible to write special user-oriented code for disk I/O.

Level 2 subroutines are written to perform disk I/O transfers merely by calling the proper subroutine and they:

a. Are user callable.

b. Pass parameters via an IOPB (Input/Output Parameter Block)

c. Wait if the 8272 FDC chip is busy

3-23. INPUT/OUTPUT PARAMETER BLOCKS

As discussed in paragraph 3-4, most disk operations require multiple-byte transfers to program the FDC chip before the command can be executed. Reads, for example, must specify the drive number, the track and sector numbers at which the read is to begin, and which side (when using double sided diskettes). The I/O driver routines communicate this data to the FDC chip in the requisite order, reading them out of a user-programmed Input/Output Parameter Block (IOPB).

The IOPB must be programmed before the routine is called. The IOPB may be located at any location in the host iSBC board RAM memory convenient to the user program.

Fourteen of the FDC's fifteen commands use the IOPB. Eight of the commands use the full nine bytes, three use two bytes, and the remaining command uses six bytes of the IOPB. Regardless of the command or the number of bytes it employs, the user program must fill the IOPB prior to the call.

Some bytes of the IOPB are dynamic (e.g., track and sector number) and must be written by the calling program into the IOPB just prior to the CALL. Other bytes (e.g., density, bytes per sector, sector per track, step rate) remain fixed during program operation. Fixed parameters can be specified by a DECLARE at compile time, as shown in Table 3-13, which is an example of a declare table for a double-density drive.

3-24. SUBROUTINE ADDRESSES

The following examples are typical for any iSBC host board. The I/O drivers are assembled to reside at addresses 800H through BFFH. The ROM where they reside is located on the iSBC base board. Normally there is a ROM-resident 2Kb monitor occupying the address 000H through 7FFH. After installing the I/O ROM driver, the memory map is as shown in figure 3-20.

The I/O drivers are accessed by calling the desired subroutine address as listed in table 3-14. The details of each calling sequence are given in the PL/M 80 listing, Appendix A. The I/O addresses listed in table 3-14 are the actual location of the drivers.

The FDC DELAY routine is used as a chip delay in reading the 8272 FDC chip main status port. Since the 8272 FDC chip is a computer, it needs time to change its status after receiving/sending a byte via the data port. The delay is different depending on the iSBX 218 board clock rate. Table 3-15 provides the recommended delay count values.

The I/O drivers assume a iSBX base port address of C0H to CFH or F0H to FFH. If the iSBX 218 address is changed on the base board, the I/O drivers must be recompiled with the corresponding port references.

Table 3-13. Sample Declare Table

```
DECLARE BASE LITERALLY '0F0H';
DECLARE BASE$1 LITERALLY '0F1H';
DECLARE IOPB$POINTER ADDRESS;
DECLARE IOPB BASED IOPB$POINTER BYTE;
DECLARE INTER$LOCATION ADDRESS;
DECLARE INT$DATA BASED INTER$LOCATION(20H) BYTE;
DECLARE C BYTE;      /*CYLINDER NUMBER*/
DECLARE H BYTE;      /*HEAD ADDRESS*/
DECLARE R BYTE;      /*SECTOR NUMBER FOR READ OR WRITE*/
DECLARE N BYTE;      /*N STANDS FOR THE NUMBER OF DATA
                       BYTES WRITTEN IN A SECTOR*/
DECLARE EOT BYTE INITIAL(1AH);  /*EOT STANDS FOR THE FINAL SECTOR
                                  NUMBER ON A CYLINDER*/
DECLARE GPL BYTE INITIAL(0EH);  /*GAP LENGTH*/
DECLARE GPL3 BYTE INITIAL(36H); /*GAP LENGTH USE WHEN FORMATING THE DISK */
DECLARE STP BYTE;    /*IF STP=1 THE DATA IN THE CONTIGUOUS
                       SECTOR IS COMPARED BYTE BY BYTE WHITH
                       DATA SENT FROM THE PROCESSOR AND IF
                       STP=2 THEN ALTERNATE SECTORS ARE READ
                       AND COMPARED*/
DECLARE DTL BYTE;    /*DATA LENGTH*/
DECLARE PCN BYTE;    /*PRECENT CYLINDER NUMBER*/

DECLARE ST0 BYTE;    /*STATUS 0*/
DECLARE ST1 BYTE;    /*STATUS 1*/
DECLARE ST2 BYTE;    /*STATUS 2*/
DECLARE ST3 BYTE;    /*STATUS 3*/
DECLARE SC BYTE;     /*SECTOR PER CYLINDER*/
DECLARE D BYTE;      /*DATA PATTERN WHICH IS GOING TO BE
                       WRITTEN INTO A SECTOR*/
DECLARE HD BYTE;     /*HEAD*/
DECLARE HLT BYTE INITIAL(25H);  /*HEAD LOAD TIME*/
DECLARE MF BYTE;     /*FM OR MFM MODE*/
DECLARE MT BYTE INITIAL(0H);    /*MULTI-TRACK*/
DECLARE NCN BYTE;    /*NEW CYLINDER NUMBER*/
DECLARE HUT BYTE INITIAL(03H);  /*HEAD UNLOAD TIME*/
DECLARE SK BYTE INITIAL(0H);    /*STANDS FOR DELETED DATA ADDRESS MARK*/
DECLARE SRT BYTE INITIAL(60H);  /*STEP RATE TIME*/
DECLARE US0 BYTE;    /*UNIT SELECT*/
DECLARE US1 BYTE;    /*UNIT SELECT*/
DECLARE ND BYTE;     /*DMA OR NON DMA*/
```

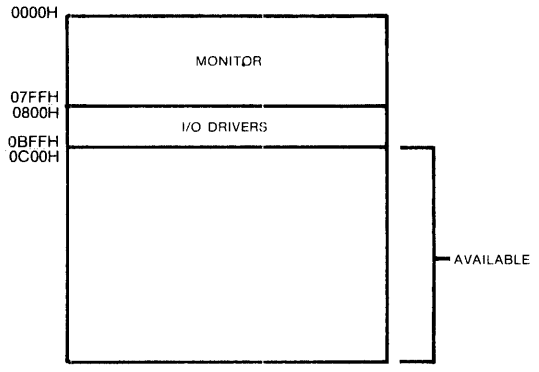


Figure 3-20. Memory Map

Table 3-14. I/O Driver Subroutine Addresses

08E7H	SYM	SENSEDRIVESTATUS
0942H	SYM	SENSEINTSTATUS
095AH	SYM	READCOMMAND
097BH	SYM	RECALCOMMAND
09CDH	SYM	SEEKCOMMAND
0A2EH	SYM	READDELDATA
0A4FH	SYM	READATRACK
0A6CH	SYM	READID
0AAEH	SYM	WRITEDATA
0ACBH	SYM	WRITEDELDATA
0AE8H	SYM	SCANEQUAL
0B0CH	SYM	SCANLOWEQUAL
0B30H	SYM	SCANHIGHEQUAL
0B54H	SYM	FORMATCOMMAND
0BA9H	SYM	STATUSIDINFO
0BE2H	SYM	SPECIFYCOMMAND

Table 3-15. Delay Count

8272 FDC CLOCK RATE	DELAY COUNT
8 MHz	12 μsec
4 MHz	16 μsec

The FDC signals completion of most of its operations by activation of the interrupt request line. The host board jumpers must be connected to the interrupt request logic.

3-25. POWER-ON INITIALIZATION

When power is first applied to the 8272 FDC chip, it is unprogrammed and the board is essentially nonfunctional. The first requisite step is to reset the device. Resetting can be accomplished by hardware (reset switch on system front panel/system reset or programmable port) which causes RESET on the iSBX 218 to activate.

Power-on time is also a good time to "CALL SPECIFY" and set the parameters for the 8272 FDC chip.

3-26. PROGRAMMING THE 8272 FDC CHIP, GENERAL FLOW CHART

Figure 3-21 shows a very generalized program flow chart for the 8272 FDC chip's command phase. The fifteen commands are broken down into SPECIFY, SENSE DRIVE STATUS, SEEK, RECALIBRATE, and ten data transfer commands. The remaining command, SENSE INTERRUPT STATUS, is only issued in response to an interrupt.

Figure 3-22 shows the detailed steps of the command phase. 'Serial' commands (e.g., Read Data) need the exclusive use of the FDC and must wait for the FDC to be idle. A 'Parallel' command (e.g., Seek), may start while another 'Parallel' command is in process but must wait for the FDC to be idle. The two entry points Command FDC Serial and Command FDC Parallel are used in the flow charts in figures 3-21 and 3-22.

The SPECIFY command sets up three internal timers and establishes whether the DMA mode is to be evoked. SPECIFY is typically performed only at power-on initialization. SPECIFY has neither an execution phase or a result phase and therefore generates no interrupt.

The SENSE DRIVE STATUS command can be performed between other commands to obtain the status of any one of the drives. The status of a drive is available immediately; thus Sense Drive Status has no execution phase and generates no interrupt.

The SEEK and RECALIBRATE commands are issued to position the read/write head on a drive in preparation for subsequent data transfer commands. The host processor is not involved during the execution phase of these commands. At the end of the execution phase the 8272 FDC chip generates an interrupt. The 8272 FDC chip can control up to four simultaneous Seek/Recalibrate commands when not involved with a data transfer operation with any of the drives. The Seek/Recalibrate commands and any data transfer commands are mutually exclusive.

The ten data transfer commands are READ DATA, READ DELETED DATA, WRITE DATA, WRITE DELETED DATA, READ A TRACK, READ ID,

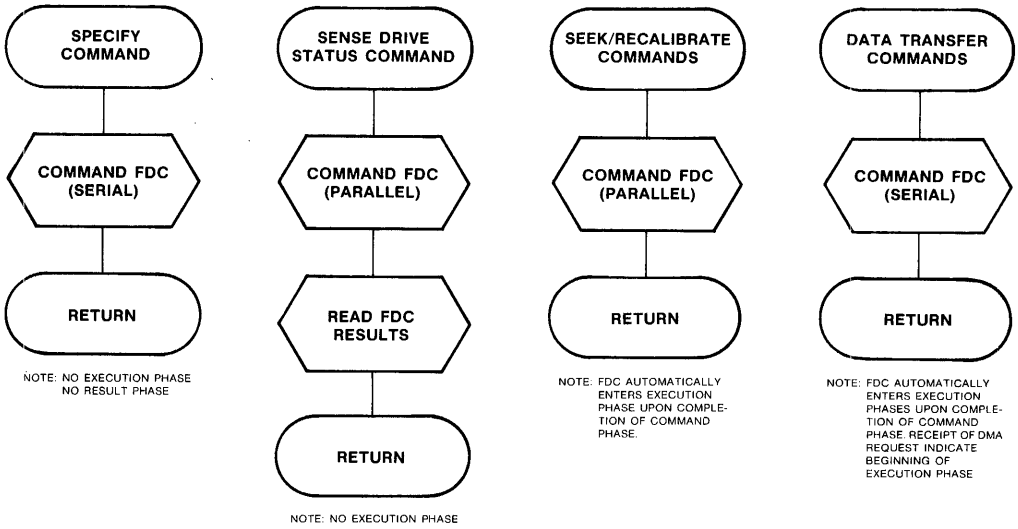


Figure 3-21. Generalized FDC Command Phase Flow Chart

FORMAT A TRACK, SCAN EQUAL, SCAN LOW OR EQUAL, and SCAN HIGH OR EQUAL. All of these commands may be executed in the DMA or Non-DMA mode to transfer the data bytes to or from the diskette.

NOTE

Consult the PL/M 80 listing in Appendix A for the detailed calling sequence of each command.

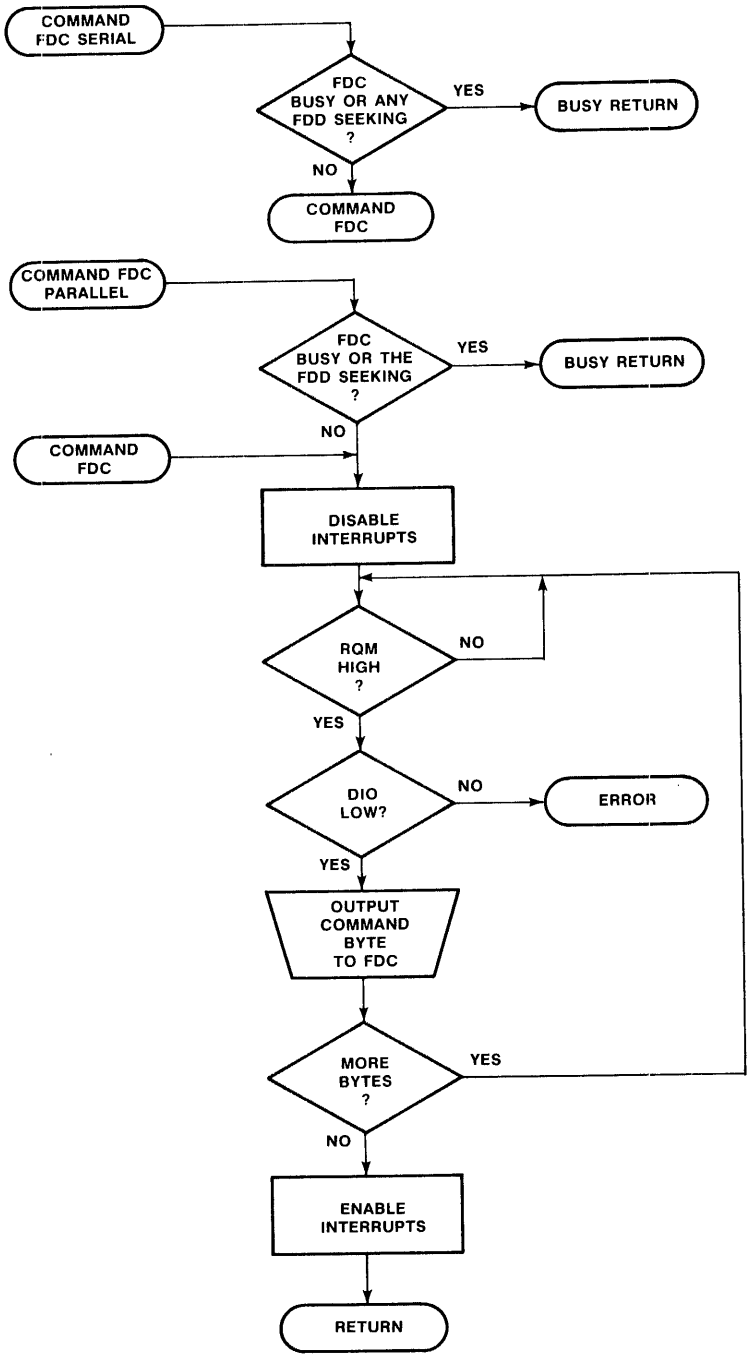


Figure 3-22. Serial/Parallel Command Phase Flow Chart

Figure 3-23 shows how to process interrupts from the iSBX 218 board. When a data transfer command terminates, its result phase has started and the host reads the results from the 8272 FDC chip. The interrupt is automatically reset when the last byte is read in the result phase.

When the completion of a Seek or Recalibrate operation sets an interrupt, a Sense Interrupt Status command must be executed to properly terminate the command. The Seek and Recalibrate commands do not have a result phase, but rather depend on the result phase of the Sense Interrupt Status.

The iSBX 218 board I/O drivers have one entry point for interrupt processing. INT1 is used to interrupt the iSBC host board. The iSBC microcomputer base board supporting the iSBX connector will recognize two interrupt lines. Presumably the user will write his own Interrupt Service Routine (ISR). Typically this ISR will save all registers and then start polling the various possible interrupt sources. The iSBX 218 board may be polled by a CALL INT1. The INT1 subroutine will check the FDC for interrupts. If the iSBX 218 board was the source of the interrupt, then the host Processor will read the 8272 FDC chip result bytes and clear the interrupt.

Figure 3-23 shows the detailed steps necessary in the result phase. As shown in the flow chart, the base board microprocessor does not need to count the bytes read in the result phase. It can tell when the operation is complete by checking the FDC busy bit in the Main Status Register. When this bit goes low, it indicates that all bytes in the result phase have been read and the 8272 FDC chip is ready for the next command.

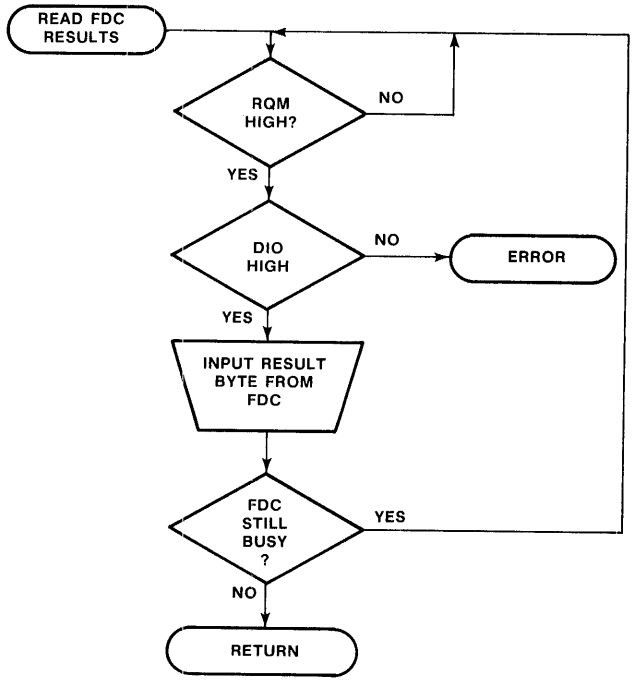


Figure 3-23. Interrupt Process or Result Phase Flow Chart

3-27. SAMPLE CALL TO I/O DRIVERS

The Read Data command is used to illustrate how to use the I/O drivers given in Appendix A. Suppose that it is desired to read Unit 1, Cylinder (track) 1, Sectors 1 on side 1. Unit 1 is assumed to be an 8" double-density IBM System 34 compatible diskette. The IOPB is shown in figure 3-24.

IOPB for Read Data
Track 1, Sector 1, Head 1, Unit 1, MFM Mode

3400H	= 46H	Command
3401H	= 05H	Head and unit number
3402H	= 01H	Cylinder number
3403H	= 01H	Head address
3404H	= 01H	Sector number
3405H	= 01H	Number of bytes
3406H	= 1AH	End of track
3407H	= 0EH	Gap length
3408H	= FFH	Data length

Figure 3-24. Sample IOPB for Read Data

CHAPTER 4
PRINCIPLES OF OPERATION

4-1. INTRODUCTION

This chapter provides a functional description and circuit analysis of the iSBX 218 Flexible Disk Controller Multimodule I/O Board. The functional description includes details on the iSBX 218 board interface to the host iSBC board and to the disk drives. Also, a description of data handling on the iSBX 218 board and the method of transferring this data to a host memory location is included. Figure 4-1 shows a block diagram of the iSBX 218 board.

4-2. INTERFACE SIGNAL DESCRIPTION

The following paragraphs describe the function and use of the signals that communicate data and commands between the host iSBC board and the iSBX 218 board. A description of the signals that communicate data and commands between the iSBX 218 board and the disk drives is also included. To obtain a further description of how these signals are used on their source or destination (host board or disk drive), consult the appropriate manual on that device.

4-3. iSBX BUS INTERFACE

The iSBX bus interface is grouped into six functional classes:

a. Control Lines

b. Address and Chip Select Lines

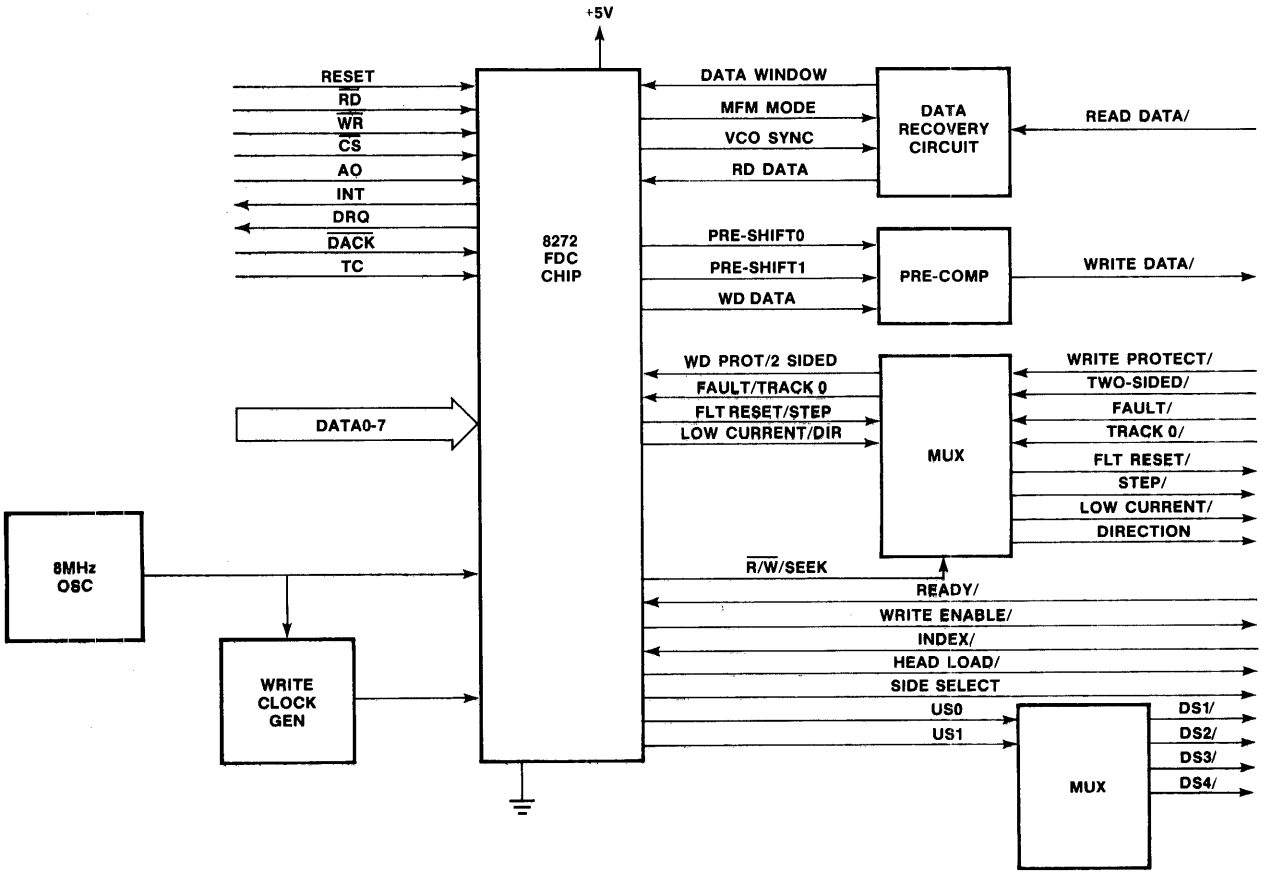


Figure 4-1. Block Diagram of iSBX 218 Board

c. Data Lines
d. Interrupt Lines
e. Option Lines
f. Power Lines

4-4. CONTROL LINES. The Control lines provide the host iSBC board a means to communicate to the iSBX 218 board. This communication link is broken down into four unique functions. These functions, Command Lines, DMA Control, Initialize, and System Control are listed and described in the following text.

4-5. COMMAND LINES. The Command Lines (IORD/, IOWRT/) are active low signals that provide the communications link between the base board and the iSBX 218 board. An active command line, conditioned by Chip Select (MCS0/), indicates to the iSBX 218 board that the address lines are valid and the board should perform the specified operation.

4-6. DMA LINES. The DMA Lines (MDRQT, MDACK/) are the communication link between the DMA controller device on the host board and the iSBX 218 board. MDRQT is an active high output signal from the iSBX 218 board to the DMA device on the host iSBC board, requesting a DMA cycle. MDACK/ is an active low input signal to the iSBX 218 board from the DMA device on the host board, acknowledging that the requested DMA cycle has been granted. One byte of data is transferred between the iSBX 218 board and the host board for each DMA cycle.

An additional line, Terminal Count (TC), sent to the iSBX 218 board from the host board indicates to the 8272 FDC chip that the data transfer is complete, and DMA operations should be terminated. The Terminal Count signal is also used to terminate non-DMA transfers. The Terminal Count signal is sent to the iSBX 218 board as an option line (PIN 30), since the iSBX connector is not configured for this line.

4-7. INITIALIZE LINE. The Initialize Line (RESET) sent to the iSBX 218 board is generated by the base board to put the iSBX 218 board in a known state.

4-8. SYSTEM CONTROL LINE. The System Control Line (MPST/) is an output signal from the iSBX 218 board to the base board. The signal, identified as Multimodule Board Present, is an active low and indicates to the base board I/O decode logic that an iSBX 218 board is installed. The MPST/ signal is physically grounded on the iSBX 218 board.

4-9. ADDRESS AND CHIP SELECT LINES. The iSBX connector provides three address lines (MA0, MA1, MA2). However, the iSBX 218 board uses only MA0. The same is true for the Chip Select Lines, two are provided (MCS0/, MCS1/), but only MCS0/ is used by the iSBX 218 board.

The base board decodes I/O addresses and generates the chip selects for the multimodule boards. The base board decodes all but the lower order three address bits in generating the multimodule board chip selects. Thus, a base board would normally reserve two blocks of 8 I/O ports for each iSBX socket provided.

4-10. ADDRESS LINE. The single address line to the iSBX 218 board, MA0, is used in conjunction with the Chip Select Line to establish the I/O ports being addressed. When data is to be written into or read out of the data port of the 8272 FDC chip, MA0 is set high. When the Main Status Register is being read, MA0 is set low.

4-11. CHIP SELECT LINE. A single chip select line to the iSBX 218 board, MCS0/, is used in conjunction with the address line to establish the I/O port being addressed. This line being active (low) conditions the I/O command signals and thus enables communication with the iSBX 218 board.

4-12. DATA LINES. Eight bidirectional data lines (MD0-MD7) are used to transmit or receive information to or from the iSBX port. A data line is active or set when high. MD0 is the least significant bit.

4-13. INTERRUPT LINES. The Interrupt Lines (MINTR0, MINTR1) are active high output lines used to make interrupt requests to the host iSBC board. The iSBX 218 board requires only one interrupt line (MINTR1).

4-14. OPTION LINES. There are two option lines (OPT0, OPT1) provided as reserve lines, that are connected to wire wrap posts on both the base board and the iSBX board. They are for unique requirements where a special signal not normally provided through the iSBX bus is needed. The iSBX 218 board uses the OPT0 line to provide a path for the Terminal Count (TC) signal, required to terminate data transfer commands.

NOTE

The host iSBC board must provide the Terminal Count signal to Pin 30 of the iSBX connector, via jumper connections. The iSBX 218 board is prewired for this condition. The host iSBC board must activate Terminal Count via a programmed parallel port output or DMA controller Terminal Count output whenever the required number of bytes have been transferred for a particular command.

4-15. POWER LINES. The iSBX connector provides for the base board to send +5 and ±12 Volts; however, the iSBX 218 board requires only +5 Volts. Volts.

4-16. MMIO COMMAND OPERATIONS

The command lines from the base board are driven by tri-state drivers with pull-up resistors or standard TTL totem pole drivers. These lines indicate to the iSBX 218 board what action is being requested.

4-17. I/O READ. The I/O Read command timing is shown in figure 4-2. The base board generates a valid I/O address and chip select for the iSBX 218 board. In the performance of an I/O read operation, the host iSBC board can address either the Data Port or the Main Status Register of the 8272 FDC chip. To address the Data Port, the host iSBC board sets MA0 high; and to address the Main Status Register, MA0 is set low. After the set up timings are met, the host iSBC board activates the IORD/ line. The iSBX 218 board must put valid data on the data bus (MD0-MD7) within 250ns. The host iSBC board reads the data and removes the read command, address, and chip select.

4-18. I/O WRITE. The I/O Write command timing is shown in figure 4-3. The host iSBC board generates a valid I/O address and chip select for the iSBX 218 board. In the performance of an I/O write operation,

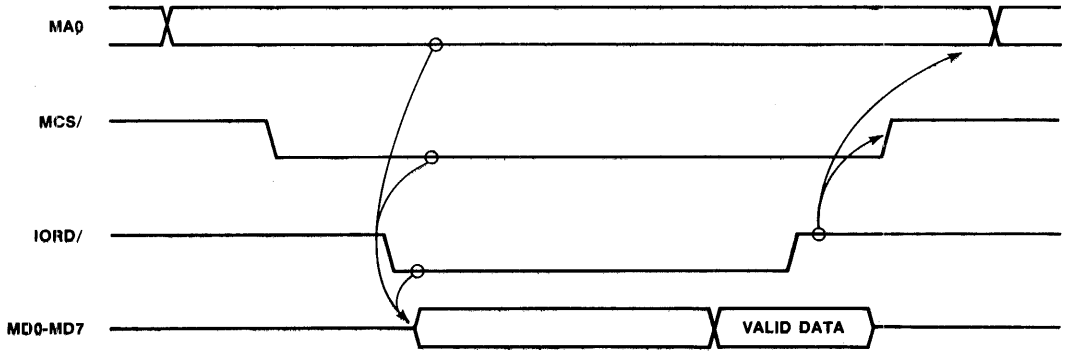


Figure 4-2. I/O Read Timing

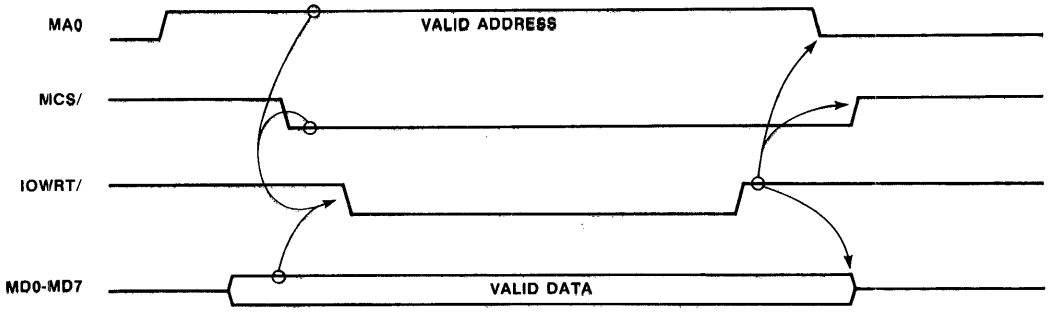


Figure 4-3. I/O Write Timing

the host iSBC board can only address the Data Port of the 8272 FDC chip. To address the Data Port, the base board sets MA0 high. After the set up timings are met, the base board activates the IOWRT/ line. The IOWRT/ line remains active (low) for 300ns and the data is valid for 250ns before IOWRT/ is removed. The host iSBC board then removes the data address and chip select signals.

4-19. DIRECT MEMORY ACCESS (DMA). The iSBX 218 board can be operated in DMA or non-DMA mode. When the base board is equipped with a DMA controller the iSBX bus will support DMA operation, permitting the host board processor to perform other tasks while data is being transferred. The following timing example shows the interface lines in their operational sequence. Because of the similarity between a DMA read and DMA write, only the DMA Read is illustrated, figure 4-4. A DMA cycle is initiated when the iSBX 218 board activates MDREQ to the DMA controller on the base board. Once the DMA controller gains control of the iSBX bus, it acknowledges back to the iSBX 218 board with MDACK/. The DMA controller then activates an I/O Read and the iSBX 218 board puts valid data on the data bus (MD0-MD7) within 250 nsec from the leading edge of IORD/. The DMA controller then activates MEM WRITE/ to load the Read data into the host iSBC board memory. The MDACK/ signal acts as a chip select and address to the Multimodule board (the MCS and MA0-MA1 signals are undetermined as they are driven by the memory address). The iSBX 218 board removes the MDRQT during the cycle to stop the DMA cycle. Once the read operation is complete the DMA controller deactivates the read command providing a data hold time. If the DMA request signal was removed, the DMA controller will release the iSBX bus back to the host processor and remove MDACK/. If the request is not removed, the DMA controller will procede to another DMA cycle. The DMA cycle must be repeated no longer than every 27μs (FM mode) or 13μs (MFM mode) to prevent setting the OR flag. The operation is halted by the activation of Terminal Count.

4-20. DISK DRIVE INTERFACE

The disk drive interface provides the communications link between the iSBX 218 board and the disk drives (up to four). The lines that make up this interface can be divided into three groups: Control, Data, and Power. Since floppy disk drives from different manufacturers can interface to the iSBX 218 board, only a typical disk drive is discussed. All interface connections to the disk drives are made through the cable connected to J1 of the iSBX 218 board. See Chapter 2 for cable configuration. For a more detailed description of these interface lines, or different line nomenclature, consult the manufacturer's OEM manual for the specific disk drive. The disk drive to controller interface cable connections on the iSBX 218 board are listed in table 2-3.

4-21. POWER LINES. All power required by the disk drives is supplied by the user through a separate cable and is not routed through the iSBX 218 board.

4-22. CONTROL/DRIVE STATUS LINES. The following lines either send control signals to the drive, or receive back to the iSBX 218 board, indications of the drive status or condition.

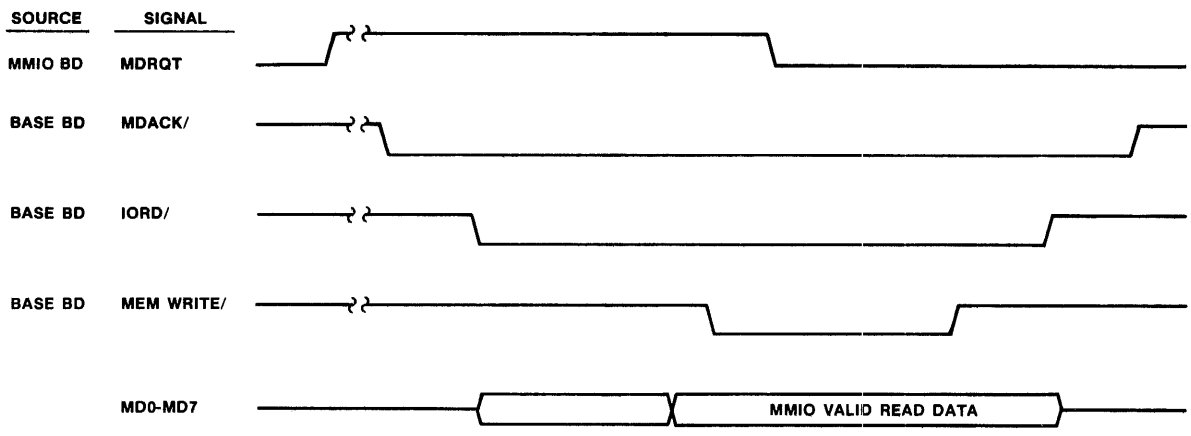


Figure 4-4. DMA Read Timing

4-23. DRIVE SELECT LINES (Drive 1 Sel/ - Drive 4 Sel/). There are four drive select lines going to each of the disk drives. Each drive must be configured to recognize a unique drive select line (see drive manufacturer's OEM manual). This drive select line, when active (low), enables the line receivers and drivers in the disk drive, thus opening the lines of communication between the drive and the iSBX 218 board.

4-24. SIDE SELECT/. This interface line from the iSBX 218 to the disk drive defines which side of a two-sided diskette will be used for reading or writing. When this line is high, the R/W head on side 0 surface of the diskette is selected. Conversely, when this line is low, side 1 is selected. When switching from one head to the other, a 100μs delay is required before any read or write operation can be initiated.

4-25. DIRECTION. The DIRECTION/ control line defines the direction of motion the R/W heads will take when the Step line is pulsed. When this line is high, the drive actuator will cause the heads to move out (away from the diskette center) coincident with the Step pulses. When this line is low, the heads will move toward the diskette center. The outermost track on the diskette is track 00.

4-26. STEP/. The STEP/ control line causes the R/W heads to move with the direction of motion defined by the Direction signal. The access motion is initiated with each transition of the Step signal going from a low to a high, or at the trailing edge of the STEP/ pulse. Any change in the Direction signal must be made at least 1μs before the next step pulse trailing edge.

4-27. HEAD LOAD/. This control signal is useful in disk to disk copy operations. It allows the user to keep the heads loaded on all drives thereby eliminating the 45 to 50ms head load time. When the drive is selected the stepper is energized and a delay (HLT) must be introduced before a Read or Write operation can be performed. This is to allow the R/W heads to settle after the stepper motor is energized. To select a drive requires both Drive Select and Head Load active.

4-28. WRITE ENABLE/. The active state of this control signal (low) enables the drive's write circuitry, permitting the write data to be written on the diskette. The inactive (high) state of this line activates the read circuitry and stepper logic. The R/W heads are therefore prevented from being positioned with WRITE ENABLE/ active.

4-29. LOW CURRENT/. This control interface signal, when active (low), causes a reduced amount of R/W head current to flow when writing on the diskette between tracks 43 and 76. When writing on tracks 0 through 42, this line should be high, thus selecting the higher write current.

4-30. TWO-SIDED/. The TWO-SIDED/ signal sent to the controller, when active (low) indicates a double sided diskette is installed on the selected drive. When this signal is high a single sided diskette is installed in the drive.

4-31. FAULT/. This interface line from the drive, when active (low), indicates to the controller that the drive's fault logic is set. Activation of this signal to the controller indicates that further positioning, read, or write operations would be unsafe and are inhibited on the drive.

4-32. FAULT RESET/. The FAULT RESET/ signal is sent to the drive (active low) from the controller to reset the fault logic in the drive.

4-33. INDEX/. This interface signal is provided by the drive once every revolution (166.67ms for an 8"disk) of the diskette indicating the beginning of the track. This signal is a negative pulse of about 1.8ms. Timing based on this pulse is synchronized on the trailing edge (negative to positive transition).

4-34. READY/. The READY/ signal is an active low signal that indicates to the controller the drive has sensed two index holes after a diskette is properly inserted and the drive door closed. It can also indicate the same index detection following application of +5V power to the drive. Three index holes must be detected for two-sided diskettes.

If a single-sided diskette is installed, READY/ will be active if Side 0 is selected, but READY/ will not be active if Side 1 is selected. Conversely, if a two-sided diskette is installed, READY/ will be active when either side is selected.

4-35. TRACK 00/. The active state of this signal (low), indicates that the drive's R/W heads are positioned over track zero and the accessing circuitry is driving current through phase one of the stepper motor. This signal is high when the R/W heads are not positioned over track 00.

4-36. WRITE PROTECT/. This active low signal provided to the controller indicates that a diskette

that is Write Protected has been installed on the drive. This signal notifies the controller that no write operations should be performed on this diskette. Write protected diskettes have a special notch which the drive detects thus causing the generation of the WRITE PROTECT/ signal.

4-37. DATA LINES. The serial data lines provided by the interface are write data and read data.

4-38. WRITE DATA/. The WRITE DATA/ interface line provides the path from the controller to the drive for the data to be written on the diskette. Each transition from a high to a low level will cause the current through the R/W head to reverse, thereby writing a data or clock bit. This line is enabled by WRITE ENABLE/ being active.

4-39. READ DATA/. The READ DATA/ interface line provides the path from the drive to the controller for the 'raw' data read from the diskette. This data received at the controller contains both the data and clock bits. The controller does the necessary data separation and waveshaping before the data is transferred to the host board in byte form.

4-40. BOARD CIRCUIT DESCRIPTION

The iSBX 218 Flexible Disk Controller board interfaces between the host iSBC board and the flexible disk drives. In order to properly interface with the disk drives without loading down the iSBX 218 board and to provide the necessary isolation, line drivers and receivers are employed on the iSBX 218 board. These circuits are contained in U2, U3, and U4 on the iSBX 218 board (refer to sheet 1 of figure 5-2). The different interface line definitions are discussed in previous paragraphs.

The following paragraphs describe the circuit functions resident on the iSBX 218 board:

4-41. 8272 FDC CHIP

The internal block diagram of the 8272 FDC chip is provided in figure 4-5. This figure is a reference to enable the reader to better understand the following text on circuits that operate with the 8272 FDC chip. The internal operation of the chip is very complicated and a detailed description of how it functionally operates is provided in the "8272 SINGLE/DOUBLE DENSITY FLOPPY DISK CONTROLLER" Data Sheet.

8272 INTERNAL BLOCK DIAGRAM

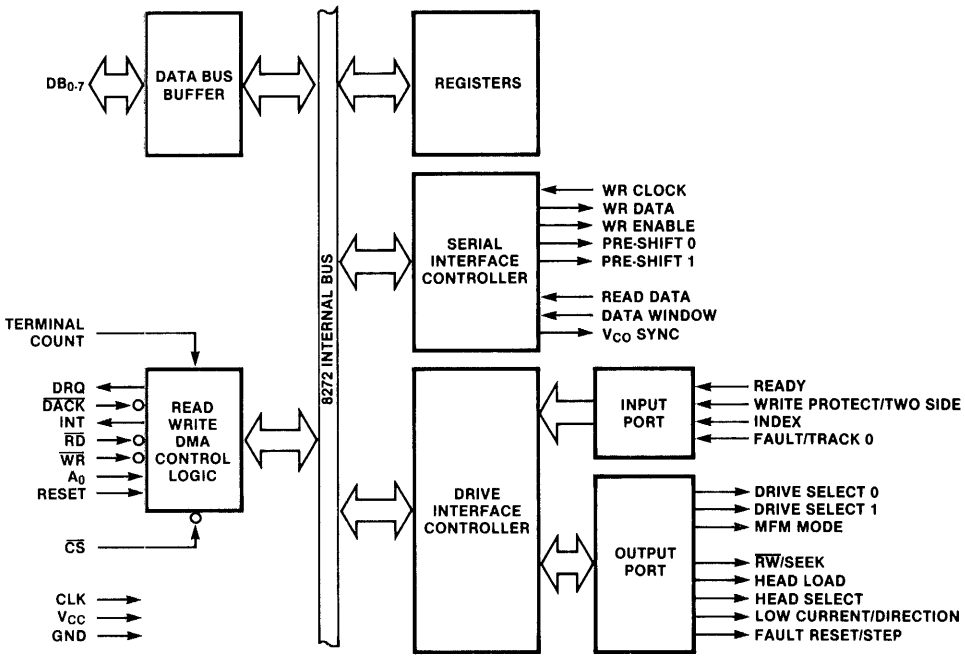


Figure 4-5. 8272 Internal Block Diagram

4-42. CLOCK. The clock circuit is shown in sheet 2 of figure 5-2. This circuit consists of Y1 (8MHz crystal) and U15. Jumper W2 is provided for test purposes only and should not be removed. Jumper W4 provides the user the option of selecting the proper frequency for an 8" or 5¼" disk drive. When the jumper W4 is installed between A-C, for 8" diskettes, the clock frequency out of U11 is 8MHz. Connecting this jumper in the manner described causes U15-2 to remain low and U15-5 stays high enabling U11-2, thus gating the 8MHz clock out to the iSBX 218 board. U11-12 is conditioned in the same manner causing the PLO 8 MHz to be gated through U11-11 to the counter U10.

When jumper W4 is connected between A-B, for 5¼" diskettes, U15-2,12 are both conditioned high, causing the frequency at U15-5 to be divided by two (4MHz). Also, the PLO frequency at U15-9 will be divided by two (4MHz) and sent to U11-12.

The frequency counter U10 scales down the frequency inputs at U10-1 and U10-13 to provide the proper drive signals for the Mode selector chip U14. The Mode Selector chip U14 selects either FM frequency or MFM frequency signals depending on the state of the 8272 MFM output signal which drives the U14-1 select input.The output of the Mode Selector chip provide a driving signal for the Write Clock F/F U19 and also necessary signals to control the sampling time and rates for the PLO. The PLO requires an error voltage to maintain the proper frequency. The error signal generation is discussed in the section on the PLO.

4-43. WRITE CLOCK. The write clock generation is accomplished by the setting and clearing of the write clock F/F U19. The output of the U14-4 sets U19 and is at a rate of either 1MHz or .5MHz for MFM Mode (depending on the configuration of W4, see paragraph 4-42) and either .5MHz or .25MHz for FM mode. U19 will be cleared by the pulse coming from U18-4. The resultant output from the write clock F/F is a 250ns positive pulse occuring at a rate of 1MHz, .5MHz, or .25MHz.

The write clock is used on the iSBX 218 board to control data handling inside the 8272 FDC chip and as an "on-frequency" input sample for the PLO when the controller is not reading diskette data.

4-44. READ DATA. Data read from the diskette is sent to the iSBX 218 board via the interface cable to J1-46 and goes to U11-9. The U11 gate is enabled to pass the read data only when the VCO line is high (from the 8272 FDC chip). The VCO line is high during the execution of a read command and low at all other times. The read data is then gated to U17-5 and U17-11 one shots. The read data signal will trigger U17 (MFM Data one shot). The output of U17-7 is used by the Mode Selector chip when performing a MFM read operation. The pulse width is determined by the position of jumper W5. When W5 is connected between A-B (for 5¼" drive), the pulse width of the signal at U17-7 is approximately 1.25μs. When W5 is connected between A-C (for 8" drive), the pulse width of the signal at U17-7 is approximately 500nsec. This pulse width corresponds to the DATA WINDOW sample time (1μsec for 5¼" drive, 500 nsec for 8" drive).

When the iSBX 218 board is configured to operate in the FM mode. U14 (Mode Selection Chip) uses the signal from U17-9. This signal is generated identical to the MFM Data signal from U17-7, except for pulse width considerations. The pulse width of the signal at U17-9 is determined by the position of jumper W6. When W6 is connected between A-B (for 5¼" disk), the pulse width of the signal at U17-9 is approximately 3μs. When W6 is connected between A-C (for 8" disk), the pulse width of the signal at U17-9 is approximately 1.25μs.

These signals, MFM data and FM data, are the raw data bits (U14-13, U14-14) received from the diskette, and pulse-width standardized (by the one-shot), before being selected by the mode selector chip (U14-12). The trailing edge (low to high transition of the standardized data signal (U14-12) is used to generate the read data (RD) signal for the 8272 and to generate sample (error) inputs for the PLO.

4-45. DATA WINDOW. The read data window created at U7-8 is a result of the FM or MFM frequency generated by counter U10-13, which is clocked by the PLO Clock. This signal is required by the 8272 FDC chip to allow it to identify and route read data pulses properly. If the read data window is incorrect, the data received at the 8272 FDC chip will be meaningless.

When the iSBX 218 board is performing a MFM read operation on an 8" diskette the following conditions are true. A 1 MHz signal at U6-1 causes the F/F to toggle such that U7-4 and U7-1 will each be enabled for 1μs. This signal generates the data window (U7-8). The logic consisting of Flip-Flops U19-5, U16-5, and U16-9 and gate U18-13 perform the dual function of generating a fixed-width 125ns read data (RD) pulse to the 8272 chip and of extending the data window state (0 or 1) for up to 250ns if an RD pulse active when the data window flip-flop toggles (U6-1). This window extender provides higher bit-shift margins in reading a drive's data.

4-46. PLO AND PHASE COMPARATOR

The PLO circuit, U12 is a Voltage Controlled Oscillator (VCO), that operates at 8MHz. The response time of the frequency control circuit is established by the configuration of W7. To fine control the oscillator and reflect any speed variations on the diskette, a phase detection/error voltage circuit is supplied. The phase comparator consists of U8 and the error voltage integration circuit is Q1, Q2, and the associated circuitry around C13. The appropriate read data signal is applied to U8-3 and a reference signal is applied to U8-11. The pulse width and frequency of these signals depend on the recording mode and the disk size being used. To determine what frequencies and pulse widths are used, determine jumper configurations and read paragraphs 4-42 through 4-45.

The voltage at C13(U12-2) controls the VCO frequency output (U12-2). An increase in voltage results in an increase in frequency. Depending on how long each side of U8 stays on, determines via Q1 and Q2 the resulting voltage charge on C13. Normally the charge on C13 is approximately +3 Volts. One transistor (Q2) provides a charge path and the other (Q1) a discharge path. If both are on an equal time, the resultant change to the charge on C13 would be zero. When the times are not equal a change in the nominal voltage across C13 causes the frequency of the PLO to shift to match the data rate being read. As the PLO frequency shifts, the phase comparator decreases the length of corrected pulses until the PLO frequency matches the data rate (no correction).

CHAPTER 5
SERVICE INFORMATION

5-1. INTRODUCTION

This chapter provides a list of replaceable parts, service diagrams, and service and repair assistance instructions for the iSBX 218 Flexible Disk Controller Multimodule I/O board.

5-2. REPLACEABLE PARTS

Table 5-1 provides a list of replaceable parts for the iSBX 218 board. Table 5-2 identifies and locates the manufacturers specified in the MFR CODE column of table 5-1. Parts that are available on the open market are listed in the MFR CODE column as "COML". Every effort should be made to procure these parts from a local (commercial) distributor.

5-3. SERVICE DIAGRAMS

The parts location diagram and schematic diagram are provided in figures 5-1 and 5-2 respectively. On the schematic diagram, a signal mnemonic that ends with a slash (e.g., DACK/) is active low. Conversely, a signal mnemonic without a slash is active high.

5-4. SERVICE AND REPAIR ASSISTANCE

United States customers can obtain service and repair assistance by contacting the Intel Product Service Hotline in Phoenix, Arizona. Customers outside the United States should contact their sales source (Intel Sales Office or Authorized Distributor) for service information and repair assistance.

Before calling the Product Service Hotline, you should have the following information available:

a. Date you received the product.

b. Complete part number of the product (including dash number). On boards, this number is usually silk-screened onto the board. On other MCSD products, it is usually stamped on a label.

c. Serial number of product. On boards, this number is usually stamped on the board. On other MCSD products, the serial number is usually stamped on a label.

d. Shipping and billing addresses.

e. If your Intel product warranty has expired, you must provide a purchase order number for billing purposes.

f. If you have an extended warranty agreement, be sure to advise the Hotline personnel of this agreement.

Use the following numbers for contacting the Intel Product Service Hotline:

Telephone

All U.S. locations,
except Alaska, Arizona, & Hawaii:

(800) 528-0595

All other locations: (602) 869-4600

TWX Number

910 - 951 - 1330

Always contact the Product Service Hotline before returning a product to Intel for repair. You will be given a repair authorization number, shipping instructions, and other important information which will help Intel provide you with fast, efficient service. If you are returning the product because of damage sustained during shipment or if the product is out of warranty, a purchase order is required before Intel can initiate the repair.

In preparing the product for shipment to the Repair Center, use the original factory packing material, if possible. If this material is not available, wrap the product in a cushioning material such as Air Cap TH-240, manufactured by the Sealed Air Corporation, Hawthorne, N.J. Then enclose in a heavy duty corrugated shipping carton, and label "FRAGILE" to ensure careful handling. Ship only to the address specified by Product Service Hotline personnel.

Table 5-1. Replaceable Parts

Reference Designation	Description	Mfr. Part No.	Mfr. Code	Qty.
C1, C2	Capacitor, 330pf, Mica, ±5%, 500V	OBD	COML	2
C3	Capacitor, 50pf, Mica, ±5%, 500V	OBD	COML	1
C4-C14	Capacitor, O.1μf, Cer, ±10%, 50V	OBD	COMl	11
J1	Connector, Multimodule	103109-001	INTEL	1
L1	Inductor, 100μh	OBD	COML	1
P1	Connector, Header, R/A EJ, 50Pin		3M	1
Q1	Transistor, NPN, 2N2222A	OBD	COML	1
Q2	Transistor, PNP, 2N2907A	OBD	COML	1
R1	Resistor, Carb, 270Ω, ¼W, ±5%	OBD	COML	1
R2, R5	Resistor, Carb, 10KΩ, ¼W, ±1%	OBD	COML	2
R3	Resistor, Carb, 4.32KΩ, ¼W, ±1%	OBD	COML	1
R4	Resistor, Carb, 20.5KΩ, ¼W, ±1%	OBD	COML	1
R6	Resistor, Carb, 1.5KΩ, ¼W, ±5%	OBD	COML	1
R7	Resistor, Carb, 1KΩ, ¼W, ±5%	OBD	COML	1
R8	Resistor, Carb, 1.3KΩ, ¼W, ±5%	OBD	COML	2
R10	Resistor, Carb, 2.2KΩ, ±5%	OBD	COML	1
R11	Resistor, Carb, 750Ω, ±5%	OBD	COML	1
R12	Resistor, Carb, 620Ω, ±5%	OBD	COML	1
R13	Resistor, Carb 330Ω, ¼W ±5%	OBD	COML	2
R14	Resistor, Pot, Cer, 20turn 0.5W 1KΩ	64XR1K	BECK	1
RP1	Resistor Pack, 1KΩ, 10 Pin	OBD	COML	1
RP2	Resistor Pack, 150Ω, 8 Pin	OBD	COML	1
U1	IC, 74LS139	SN74LS139	TI	1
U2	IC, 7407	SN7407	TI	1
U3, U4	IC, 74S240	SN74S240	TI	2
U5	IC, 74LS175	SN74LS175	TI	1
U6, U15, U16	IC, 74S112	SN74S112	TI	3
U7	IC, 74LS00	SN74LS00	TI	1
U8, 19	IC, 74S74	SN74S74	TI	1
U9	IC, 74153	SN74153	TI	1
U10	IC, 74LS393	SN74LS393	TI	1
U11	IC, 74132	SN74132	TI	1
U12	IC, 74S124	SN74S124	TI	1
U13	IC, 8272	8272	INTEL	1
U14	IC, 74157	SN74157	TI	1
U17	IC, 9602	MC8602P	MOT	1
U18	IC, 74LS02	SN74LS02	TI	1
XU13	Socket, Dip, 40 Pin	OBD	COML	1
Y1	Crystal, Osc, 8.00MHz	K1115A-8.000MHz	MOT	1
n/a	Term, Post, Sq.,.025 X .443	OBD	COML	23
n/a	Plug, Shorting, .1 CNTR	OBD	COML	8

Table 5-2. Manufacturer Codes

Mfr. Code	Manufacturer	Location
INTEL	Intel Corporation	Santa Clara, CA
TI	Texas Instruments	Dallas, TX
AMP	AMP Incorporated	Harrisburg, PA
3M	3M Corporation	Minneapolis, MN
MOT	Motorola	Phoenix, AZ
COML	Any Commercial Source; Order by Description (OBD)	
BECK	Beckman Corporation	

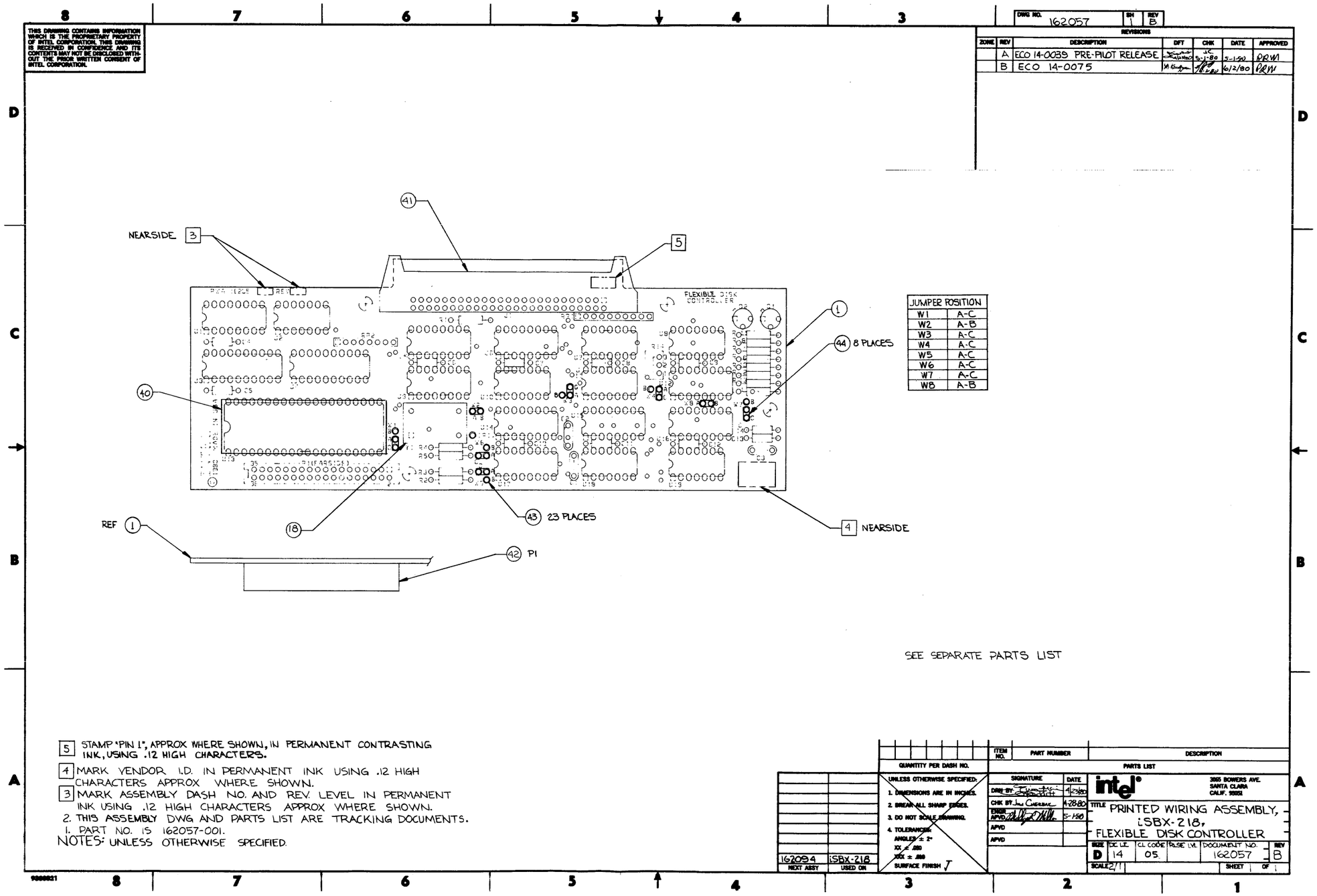


Figure 5-1. Parts Location Diagram

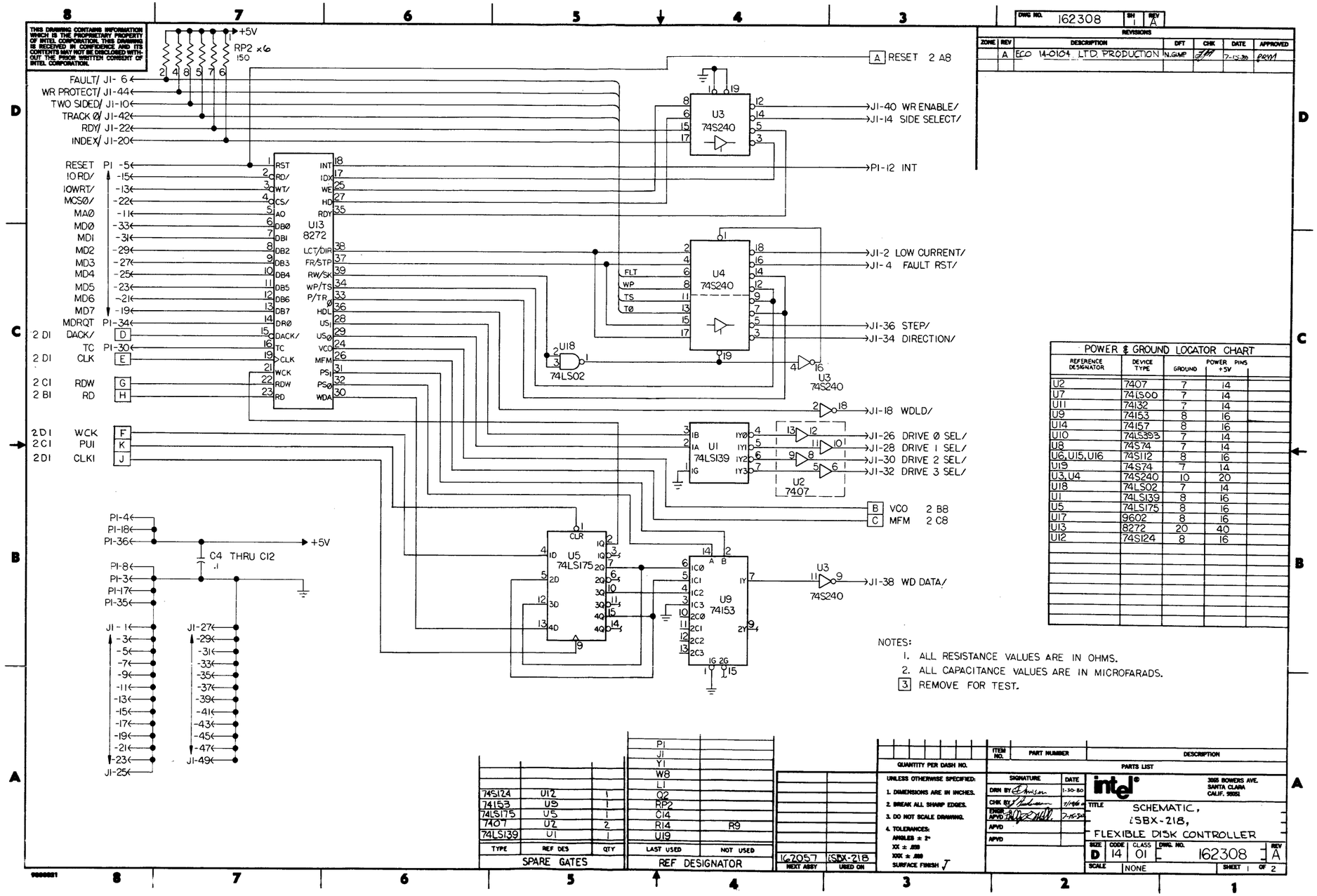


Figure 5-2. Schematic Diagram (Sheet 1 of 2)

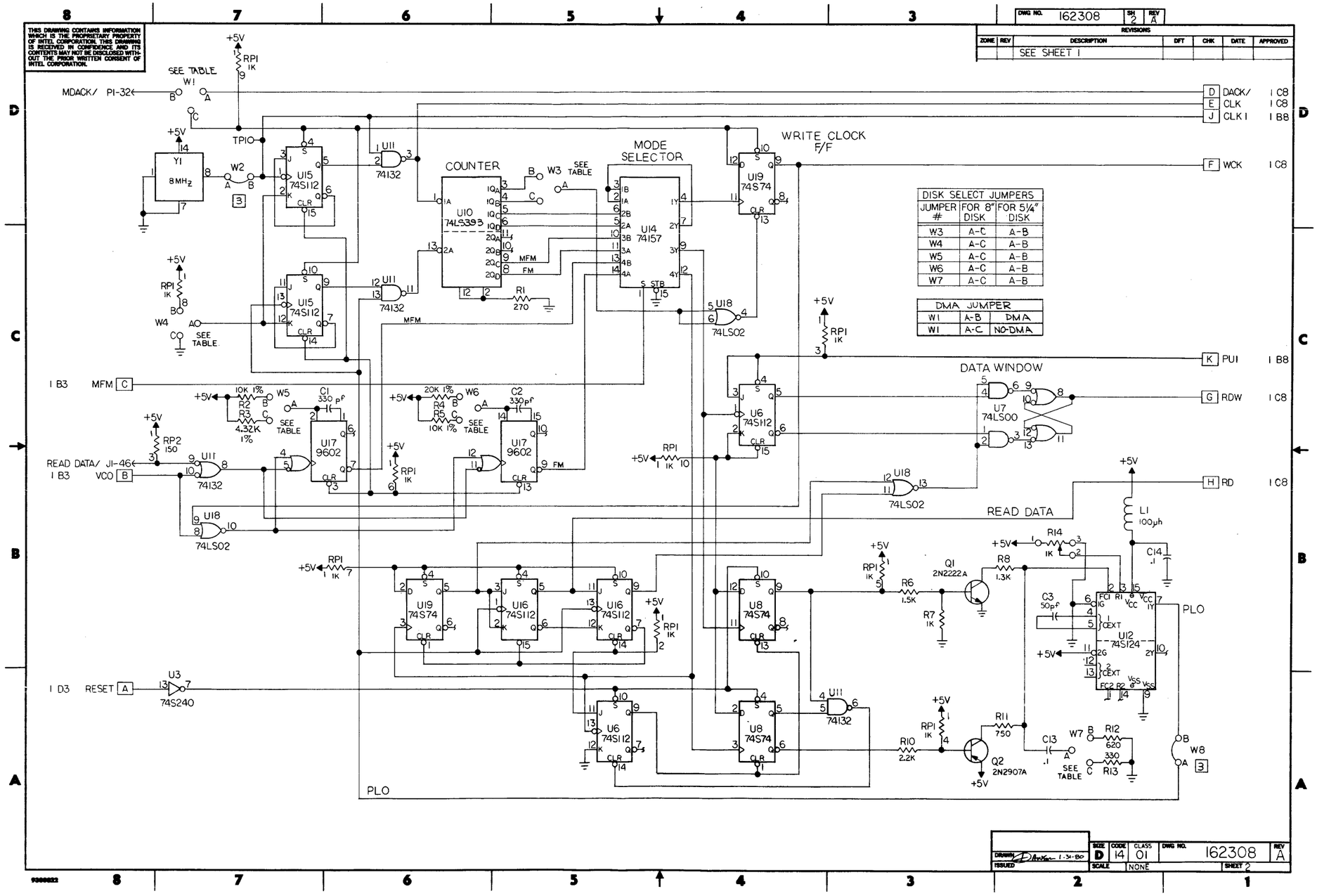


Figure 5-2. Schematic Diagram (Sheet 2 of 2)

APPENDIX A
PLM/80 SAMPLE PROGRAM

```
1          DOC:
           /*
           THE FOLLOWING PL/M-80 LISTING IS A SAMPLE PROGRAM CONTAINTING THE IOPB AND
           CALL ROUTINES TO IMPLIMENT THE iSBX-218 BOARD I/O DRIVER.
           */
           DO;
2    1         DECLARE BASE LITERALLY '0F0H';
3    1         DECLARE BASE$1 LITERALLY '0F1H';
4    1         DECLARE IOPB$POINTER ADDRESS;
5    1         DECLARE IOPB BASED IOPB$POINTER BYTE;
6    1         DECLARE INTER$LOCATION ADDRESS;
7    1         DECLARE INT$DATA BASED INTER$LOCATION(20H) BYTE;
8    1         DECLARE C BYTE;      /*CYLINDER NUMBER*/
9    1         DECLARE H BYTE;      /*HEAD ADDRESS*/
10   1         DECLARE R BYTE;      /*SECTOR NUMBER FOR READ OR WRITE*/
11   1         DECLARE N BYTE;      /*N STANDS FOR THE NUMBER OF DATA
                                      BYTES WRITTEN IN A SECTOR*/
12   1         DECLARE EOT BYTE INITIAL(1AH);  /*EOT STANDS FOR THE FINAL SECTOR
                                                 NUMBER ON A CYLINDER*/
13   1         DECLARE GPL BYTE INITIAL(0EH);  /*GAP LENGTH*/
14   1         DECLARE GPL3 BYTE INITIAL(36H); /*GAP LENGTH USE WHEN FORMATING THE DISK */
15   1         DECLARE STP BYTE;    /*IF STP=1 THE DATA IN THE CONTIGUOUS
                                      SECTOR IS COMPARED BYTE BY BYTE WITH
                                      DATA SENT FROM THE PROCESSOR AND IF
                                      STP=2 THEN ALTERNATE SECTORS ARE READ
                                      AND COMPARED*/
16   1         DECLARE DTL BYTE;    /*DATA LENGTH*/
17   1         DECLARE PCN BYTE;    /*PRESENT CYLINDER NUMBER*/

18   1         DECLARE ST0 BYTE;    /*STATUS 0*/
19   1         DECLARE ST1 BYTE;    /*STATUS 1*/
20   1         DECLARE ST2 BYTE;    /*STATUS 2*/
21   1         DECLARE ST3 BYTE;    /*STATUS 3*/
22   1         DECLARE SC BYTE;     /*SECTOR PER CYLINDER*/
23   1         DECLARE D BYTE;      /*DATA PATTERN WHICH IS GOING TO BE
                                      WRITTEN INTO A SECTOR*/
24   1         DECLARE HD BYTE;     /*HEAD*/
25   1         DECLARE HLT BYTE INITIAL(25H);  /*HEAD LOAD TIME*/
26   1         DECLARE MF BYTE;     /*FM OR MFM MODE*/
27   1         DECLARE MT BYTE INITIAL(00H);   /*MULTI-TRACK*/
28   1         DECLARE NCN BYTE;    /*NEW CYLINDER NUMBER*/
29   1         DECLARE HUT BYTE INITIAL(03H);  /*HEAD UNLOAD TIME*/
30   1         DECLARE SK BYTE INITIAL(00H);   /*STANDS FOR DELETED DATA ADDRESS MARK*/
31   1         DECLARE SRT BYTE INITIAL(60H);  /*STEP RATE TIME*/
32   1         DECLARE US0 BYTE;    /*UNIT SELECT*/
33   1         DECLARE US1 BYTE;    /*UNIT SELECT*/
34   1         DECLARE ND BYTE;     /*DMA OR NON DMA*/
```

```
                $EJECT

                /*****************************************************************************
                    EXTERNALS ROUTINES FOR FORMAT,READ,WRITE,SCAN ARE ASSEMBLY LANGUAGE
                    ROUTINES. DUE TO THE TIME CRITICAL BYTE-BY-BYTE DATA HANDLING THESE
                    TYPE OF ROUTINES MUST BE USED

                */

 35   1         FOREXT:
                    PROCEDURE EXTERNAL;
 36   2         END FOREXT;

 37   1         RAEXT:
                    PROCEDURE EXTERNAL;
 38   2         END RAEXT;

                /*
                IS$DRIVE$BUSY; IS A ROUTINE THAT POLLES THE MAIN STATUS REG OF THE 8272
                               TO SEE IF THE DEVICE IS READY.
                */

 39   1         IS$DRIVE$BUSY:
                    PROCEDURE PUBLIC;
 40   2             DECLARE OK$TO$GO BYTE;
 41   2             OK$TO$GO = 0H;
 42   2             DO WHILE OK$TO$GO <>80H;
 43   3                 OK$TO$GO = INPUT(BASE) AND 0F0H;
 44   3             END;
 45   2         END IS$DRIVE$BUSY;

                /*
                OK$TO$OUTPUT$IOPB; IS A ROUTINE THAT POLLES THE MAIN STATUS REG OF THE 8272
                                   TO SEE IF THE DEVICE IS READY TO WRITE A BYTE OF IOPB TO
                                   THE DATA REGISTER.
                */

 46   1         OK$TO$OUTPUT$IOPB:
                    PROCEDURE PUBLIC;
 47   2             DECLARE OK$TO$WRITE BYTE;
 48   2             OK$TO$WRITE = 0H;
 49   2             DO WHILE OK$TO$WRITE <>90H;
 50   3                 OK$TO$WRITE = INPUT(BASE) AND 0F0H;
 51   3             END;
 52   2         END OK$TO$OUTPUT$IOPB;
```

```
                $EJECT

                /*
                OK$TO$READ$STATUS; IS A ROUTINE THAT POLLES THE MAIN STATUS REG OF THE 8272 TO
                                   SEE IF THE DEVICE IS READY TO BE READ.
                */

53   1          OK$TO$READ$STATUS:
                    PROCEDURE PUBLIC;
54   2              DECLARE OK$TO$READ BYTE;
55   2              OK$TO$READ = 0;
56   2              DO WHILE OK$TO$READ <> 0D0H;
57   3                  OK$TO$READ = INPUT(BASE) AND 0F0H;
58   3              END;
59   2          END OK$TO$READ$STATUS;

                /*
                IOPB$OUT;  THIS ROUTINE IS USED TO OUTPUT THE IOPB TO THE 8272 EXCEPT FOR
                           COMMAND WRITE DATA, WRITE DELETED DATA, AND THE SCAN COMMANDS.
                */

60   1          IOPB$OUT:
                    PROCEDURE;
61   2              DO IOPB$POINTER = 3400H TO 3408H;
62   3                  OUTPUT(BASE$1) = IOPB;
63   3                  IF IOPB$POINTER < 3408H THEN
64   3                      CALL OK$TO$OUTPUT$IOPB;
65   3              END;
66   2          END;
```

```
$EJECT

/*
IOPB$SET; THIS PROCEDURE SETS THE VALUES OF IOPB BUFFER BEFORE THE COMMAND
          BYTES ARE SENT TO THE 8272.
          MOST COMMANDS HAVE 9 BYTES, FOLLOWING IS AN EXAMPLE

                ---------------------
                !  COMMAND BYTE     !
                ---------------------
                ! HEAD & UNIT SELECT !
                ---------------------
                ! CYLINDER NUMBER   !
                ---------------------
                ! HEAD ADDRESS      !
                ---------------------
                ! SECTOR NUMBER     !
                ---------------------
                ! NUMBER OF BYTES   !
                ---------------------
                ! END OF TRACK      !
                ---------------------
                ! GAP LENGTH        !
                ---------------------
                ! DATA LENGTH       !
                ---------------------
*/

67   1      IOPB$SET:
                PROCEDURE PUBLIC;
68   2          C = PCN;
69   2          IOPB$POINTER = IOPB$POINTER + 1;
70   2          IOPB = HD OR US1 OR US0;
71   2          IOPB$POINTER = IOPB$POINTER + 1;
72   2          IOPB = C;
73   2          IOPB$POINTER = IOPB$POINTER + 1;
74   2          IOPB=H;
75   2          IOPB$POINTER = IOPB$POINTER + 1;
76   2          IOPB=R;
77   2          IOPB$POINTER = IOPB$POINTER + 1;
78   2          IOPB=N;
79   2          IOPB$POINTER = IOPB$POINTER + 1;
80   2          IOPB=EOT;
81   2          IOPB$POINTER = IOPB$POINTER + 1;
82   2          IOPB=GPL;
83   2          IOPB$POINTER = IOPB$POINTER + 1;
84   2          IOPB=DTL;
85   2      END IOPB$SET;
```

```
$EJECT
/*
THIS IS A LIST OF ABBREVATIONS USED IN THIS PROGRAM
C = CYLINDER NUMBER
D = DATA
DTL = DATA LENGTH
EOT = END OF TRACK
GPL = GAP LENGTH
H = HEAD ADDRESS
HLT = HEAD LOAD TIME
HUT = HEAD UNLOAD TIME
MF  = FM IF 0,MFM IF 1
MT = MULTI-TRACK
N = NUMBER OF DATA BYTES WRITTEN IN A SECTOR
NCN = NEW CYLINDER NUMBER
ND = NON DMA MODE
PCN = PRESENT CYLINDER NUMBER
R = SECTOR NUMBER
SC = NUMBER OF SECTOR PER CYLINDER
SK = SKIP DELETED DATA ADDRESS MARK
SRT = STEP RATE TIME
ST0 = STATUS 0
ST1 = STATUS 1
ST2 = STATUS 2
ST3 = STATUS 3
US0 & US1 = UNIT SELECT



SENSE$DRIVE$STATUS; THIS PROCEDURE SETS THE IOPB WITH THE SENSE DRIVE STATUS
                    COMMAND. AN OUTPUTS IT TO THE iSBX-218. THE RESULT IS
                    STORED IN THE ST3 BYTE.
*/


86  1   SENSE$DRIVE$STATUS:
        PROCEDURE PUBLIC;
87  2          IOPB$POINTER = 3400H;
88  2          IOPB = 04H;      /*COMMAND FOR SENSE DRIVE STATUS*/
89  2          IOPB$POINTER = IOPB$POINTER + 1;
90  2          IOPB = HD OR US1 OR US0;/*WHAT DRIVE AND WHAT HEAD*/
91  2          CALL IS$DRIVE$BUSY;
92  2          DO IOPB$POINTER= 3400H TO 3401H;
93  3              OUTPUT(BASE$1) = IOPB;
94  3              IF IOPB$POINTER < 3401H THEN
95  3                  CALL OK$TO$OUTPUT$IOPB;
96  3          END;
97  2          ST3 = INPUT(BASE$1);
98  2  END SENSE$DRIVE$STATUS;
```

```
                $EJECT

                /*
                SENSE$INT$STATUS; THIS PROCEDURE OUTPUTS THE COMMAND FOR SENSE INTERRUPT
                                  STATUS, READS BACK THE RESULTS STO AND PRESENT CYCLINDER
                                  NUMBER.
                */
 99    1        SENSE$INT$STATUS:
                    PROCEDURE PUBLIC;
100    2                CALL IS$DRIVE$BUSY;
101    2                OUTPUT(BASE$1)=08H;
102    2                CALL OK$TO$READ$STATUS;
103    2                STO = INPUT(BASE$1);
104    2                CALL OK$TO$READ$STATUS;
105    2                PCN = INPUT(BASE$1);
106    2        END SENSE$INT$STATUS;

                /*
                READ$COMMAND; THIS PROCEDURE LOADS THE IOPB WITH THE NINE COMMAND
                              BYTES ,CALLS 'IOPB$OUT' TO OUTPUT THE IOPB TO THE iSBX-218
                              AND CALLS THE ROUTINE TO READ THE BYTE FROM THE iSBX-218
                              TO THE READ BUFFER.
                */

107    1        READ$COMMAND:
                    PROCEDURE PUBLIC;
108    2            IOPB$POINTER = 3400H;
109    2            IOPB = MT OR MF OR SK OR 06H;    /*COMMAND FOR READ DATA*/
110    2            CALL IOPB$SET;
111    2            CALL IS$DRIVE$BUSY;
112    2            CALL RAEXT;
113    2        END READ$COMMAND;

                /*
                RECAL$COMMAND;  LOADS THE IOPB WITH THE TWO BYTES NEEDED TO DO THE
                                RECALIBRATION ON THE DRIVE AND OUTPUTS THE IOPB TO
                                8272. NO RESULT PHASE REQUIRED. AFTER THE INTERRUPT
                                A SENSE INTERRUPT STATUS SHOULD BE SENT.
                */

114    1        RECAL$COMMAND:
                    PROCEDURE PUBLIC;
115    2            IOPB$POINTER = 3400H;
116    2            IOPB = 07H;          /*COMMAND TO RECALIBRATE*/
117    2            IOPB$POINTER = IOPB$POINTER + 1;
118    2            IOPB= 00H OR US1 OR US0;     /*WHAT DIVE TO RECAL.*/
119    2            CALL IS$DRIVE$BUSY;
120    2            DO IOPB$POINTER = 3400H TO 3401H;
121    3                OUTPUT(BASE$1)=IOPB;
122    3                IF IOPB$POINTER < 3401H THEN
123    3                    CALL OK$TO$OUTPUT$IOPB;
124    3            END;
125    2        END RECAL$COMMAND;
```

```
                $EJECT

                /*
                 SEEK$COMMAND; IS A ROUTINE THAT LOADS THE IOPB WITH THREE COMMAND
                               BYTES. THE THIRD BYTE IS THE NCN(NEW CYLINDER NUMBER)
                               OR THE DESTINATION OF THE HEADS ON THE DRIVE. THERE
                               IS NO RESULT PHASE WITH THIS COMMAND. A SENSE INT STATUS
                               SHOULD BE SENT TO THE FDC AFTER THE ENSUING INTERRUPT.
                */
126   1         SEEK$COMMAND:
                    PROCEDURE PUBLIC;
127   2             IOPB$POINTER = 3420H;
128   2             IOPB = 0FH;          /*COMMAND FOR SEEK*/
129   2             IOPB$POINTER = IOPB$POINTER + 1;
130   2             IOPB= HD OR US1 OR US0; /*SET UP WHICH HEAD AND DRIVE*/
131   2             IOPB$POINTER = IOPB$POINTER + 1;
132   2             IOPB= NCN;           /*TRACK TO SEEK TO*/
133   2             CALL IS$DRIVE$BUSY;
134   2             DO IOPB$POINTER = 3420H TO 3422H;
135   3                 OUTPUT(BASE$1)=IOPB;
136   3                 IF IOPB$POINTER < 3422H THEN
137   3                     CALL OK$TO$OUTPUT$IOPB;
138   3             END;
139   2         END SEEK$COMMAND;

                /*
                READ$DEL$DATA; THIS COMMAND IS HANDLED THE SAME AS THE READ COMMAND
                               ONLY THE FIRST BYTE(COMMAND BYTE) IS DIFFERENT.
                */

140   1         READ$DEL$DATA:
                    PROCEDURE PUBLIC;
141   2             IOPB$POINTER = 3400H;
142   2             IOPB = MT OR MF OR SK OR 0CH;    /*COMMAND FOR READ DELETED DATA*/
143   2             CALL IOPB$SET;
144   2             CALL IS$DRIVE$BUSY;
145   2             CALL RAEXT;
146   2         END READ$DEL$DATA;

                /*
                READ$A$TRACK; THIS COMMAND IS HANDLED THE SAME AS THE READ COMMAND.
                              AGAIN THE FIRST BYTE(COMMAND BYTE) IS DIFFERENT. THIS
                              COMMAND READ ALL SECTORS FROM INDEX HOLE TO INDEX HOLE.
                */

147   1         READ$A$TRACK:
                    PROCEDURE PUBLIC;
148   2             IOPB$POINTER = 3400H;
149   2             IOPB = MF OR SK OR 02H;      /*COMMAND FOR READ A TRACK*/
150   2             CALL IOPB$SET;
151   2             CALL IS$DRIVE$BUSY;
152   2             CALL RAEXT;
153   2         END READ$A$TRACK;
```

```
                $EJECT
                /*
                READ$ID$COMMAND; IS A TWO BYTE COMMAND. THE IOPB IS LOADED WITH
                                 THE COMMAND BYTES AND IS THEN OUTPUTED TO THE FDC.
                                 THE FDC THEN LOOKS FOR THE FIRST VALID SECTOR ID.
                                 THE RESULTS OF THE SECTOR ID READ ARE IN THE
                                 RESULT BYTES READ IN THE RESULT PHASE.
                */

154   1         READ$ID:
                    PROCEDURE PUBLIC;
155   2             IOPB$POINTER = 3400H;
156   2             IOPB = MF OR 0AH;        /*COMMAND FOR READ ID DATA*/
157   2             IOPB$POINTER = IOPB$POINTER + 1;
158   2             IOPB = HD OR US1 OR US0;     /*WHAT HEAD AND WHAT DRIVE*/
159   2             IOPB$POINTER = 3400H;
160   2             CALL IS$DRIVE$BUSY;
161   2             OUTPUT(BASE$1) = IOPB;
162   2             IOPB$POINTER = IOPB$POINTER + 1;
163   2             CALL OK$TO$OUTPUT$IOPB;
164   2             OUTPUT(BASE$1) = IOPB;
165   2         END READ$ID;

                /*
                WRITE$DATA; THIS COMMAND LOADS IOPB WITH THE NINE BYTES OF THE COMMAND.
                            AND THEN CALLS A 8080/8085 ASSEMBLY ROUTINE TO OUTPUT THE IOPB
                            AND WRITE THE WRITE BUFFER TO THE FDC. AN ASSEMBLY LANGUAGE
                            ROUTINE IS USED BECAUSE OF THE VERY SHORT TIME FROM THE LAST
                            BYTE OF THE IOPB TO THE FIRST BYTE NEEDED TO BE SENT FROM THE
                            WRITE BUFFER.
                */

166   1         WRITE$DATA:
                    PROCEDURE PUBLIC;
167   2             IOPB$POINTER = 3400H;
168   2             IOPB = MT OR MF OR 05H;      /*COMMAND FOR WRITE DATA*/
169   2             CALL IOPB$SET;
170   2             CALL IS$DRIVE$BUSY;
171   2             CALL RAEXT;
172   2         END WRITE$DATA;

                /*
                THE WRITE$DEL$DATA COMMAND IS THE SAME AS THE WRITE DATA COMMAND EXCEPT
                FOR THE FIRST BYTE IN THE IOPB.
                */

173   1         WRITE$DEL$DATA:
                    PROCEDURE PUBLIC;
174   2             IOPB$POINTER = 3400H;
175   2             IOPB = MT OR MF OR 09H;      /*COMMAND FOR WRITE DELETED DATA*/
176   2             CALL IOPB$SET;
177   2             CALL IS$DRIVE$BUSY;
178   2             CALL RAEXT;
179   2         END WRITE$DEL$DATA;
```

```
                $EJECT
                /*
                SCAN$EQUAL; SCAN$LOW$EQUAL; AND SCAN$HIGH$EQUAL; THESE COMMANDS ALL
                            HAVE NINE BYTES IN THE IOPB. THE FIRST BYTE(COMMAND BYTE)
                            IS DIFFERENT FOR EACH OF THE COMMANDS. AN ASSEMBLY LANGUAGE
                            ROUTINE IS USED TO OUTPUT THE BYTES FROM THE BUFFER TO
                            THE FDC FOR THE COMPARE WITH THE BYTES COMMING OFF THE DISK.
                            THE RESULT PHASE OCCURS ON A COMPARE HIGH, LOW, OR EQUAL OR AT
                            THE INDEX HOLE WHICH EVER OCCURS FIRST.
                */

180   1         SCAN$EQUAL:
                    PROCEDURE PUBLIC;
181   2             IOPB$POINTER = 3400H;
182   2             IOPB = MT OR MF OR SK OR 11H;       /*COMMAND FOR SCAN EQUAL*/
183   2             CALL IOPB$SET;
184   2             CALL IS$DRIVE$BUSY;
185   2             CALL IOPB$OUT;
186   2             CALL RAEXT;
187   2         END SCAN$EQUAL;

188   1         SCAN$LOW$EQUAL:
                    PROCEDURE PUBLIC;
189   2             IOPB$POINTER = 3400H;
190   2             IOPB = MT OR MF OR SK OR 19H;       /*COMMAND FOR SCAN LOW OR EQUAL*/
191   2             CALL IOPB$SET;
192   2             CALL IS$DRIVE$BUSY;
193   2             CALL IOPB$OUT;
194   2             CALL RAEXT;
195   2         END SCAN$LOW$EQUAL;

196   1         SCAN$HIGH$EQUAL:
                    PROCEDURE PUBLIC;
197   2             IOPB$POINTER = 3400H;
198   2             IOPB=MT OR MF OR SK OR 1DH;     /*COMMAND FOR SCAN HIGH OR EQUAL*/
199   2             CALL IOPB$SET;
200   2             CALL IS$DRIVE$BUSY;
201   2             CALL IOPB$OUT;
202   2             CALL RAEXT;
203   2         END SCAN$HIGH$EQUAL;
```

```
                $EJECT

                /*
                FORMAT$COMMAND; IN THIS ROUTINE THE IOPB IS LOADED WITH THE INFORMATION
                                TO FORMAT A TRACK. AN ASSEMBLY LANGUAGE ROUTINE IS USED
                                TO OUTPUT THE IOPB TO FDC AS WELL AS THE 104 BYTES NEEDED
                                TO FORMAT A TRACK WITH SECTOR INFORMATION(ASSUMES A 26
                                SECTOR TRACK).
                */

204   1         FORMAT$COMMAND:
                    PROCEDURE PUBLIC;
205   2             IOPB$POINTER = 3400H;
206   2             IOPB = MF OR 0DH;    /*COMMAND FOR FORMAT TRACK*/
207   2             IOPB$POINTER = IOPB$POINTER + 1;
208   2             IOPB = HD OR US1 OR US0;
209   2             IOPB$POINTER = IOPB$POINTER + 1;
210   2             IOPB = N;
211   2             IOPB$POINTER = IOPB$POINTER + 1;
212   2             IOPB = SC;
213   2             IOPB$POINTER = IOPB$POINTER + 1;
214   2             IOPB = GPL3;
215   2             IOPB$POINTER = IOPB$POINTER + 1;
216   2             IOPB = D;
217   2             CALL IS$DRIVE$BUSY;
218   2             CALL FOREXT;
219   2         END FORMAT$COMMAND;

                /*
                STATUS$ID$INFO; THIS PROCEDURE READS THE STATUS ID INFO FROM THE FDC
                                AFTER THE EXECUTION OF A READ DATA, READ DELETED DATA,
                                WRITE DATA, WRITE DELETED DATA, READ ID, READ A TRACK,
                                THE SCAN COMMANDS, OR THE FORMAT COMMAND. ON THE FORMAT
                                COMMAND THE LAST FOUR BYTES OF THE STATUS ID ARE INVALID.
                */

220   1         STATUS$ID$INFO:
                    PROCEDURE;
221   2             CALL OK$TO$READ$STATUS;
222   2             ST0 = INPUT(BASE$1);
223   2             CALL OK$TO$READ$STATUS;
224   2             ST1 = INPUT(BASE$1);
225   2             CALL OK$TO$READ$STATUS;
226   2             ST2 = INPUT(BASE$1);
227   2             CALL OK$TO$READ$STATUS;
228   2             C = INPUT(BASE$1);
229   2             CALL OK$TO$READ$STATUS;
230   2             H = INPUT(BASE$1);
231   2             CALL OK$TO$READ$STATUS;
232   2             R = INPUT(BASE$1);
233   2             CALL OK$TO$READ$STATUS;
234   2             N = INPUT(BASE$1);
235   2         END STATUS$ID$INFO;
```

```
                $EJECT

                /*
                SPECIFY$COMMAND; THIS IS A ROUTINE THAT LOADS THE IOPB WITH THREE
                                 BYTES.THE FIRST BYTE IS THE COMMAND FOR SPECIFY.
                                 THE SECOND BYTE SPECIFIES THE HEAD STEP RATE TIME
                                 (SRT) AND THE HEAD UNLOAD TIME(HUT) FOR THE DRIVE.
                                 THE THRID BYTE SPECIFIES THE HEAD LOAD TIME(HLT) AND
                                 SELECTION OF DMA OR NON-DMA MODE(ND BIT).
                */

 236   1        SPECIFY$COMMAND:
                    PROCEDURE PUBLIC;
 237   2              IOPB$POINTER = 3400H;
 238   2              IOPB=03H;
 239   2              IOPB$POINTER = IOPB$POINTER + 1;
 240   2              IOPB = SHL(SRT,4) OR HUT;
                /*    D7    D6    D5    D4    D3    D2    D1    D0
                    ---------------------------------------------------
                    ! SRT ! SRT ! SRT ! SRT ! HUT ! HUT ! HUT ! HUT !
                    ---------------------------------------------------
                    COMMAND BYTE 2 FOR SPECIFY COMMAND
                */

 241   2              IOPB$POINTER = IOPB$POINTER + 1;
 242   2              IOPB = SHL(HLT,1) OR ND;
                /*     D7    D6    D5    D4    D3    D2    D1    D0
                     ---------------------------------------------------
                     ! HLT ! HLT ! HLT ! HLT ! HLT ! HLT ! HLT ! ND !
                     ---------------------------------------------------
                     COMMAND BYTE 3 FOR SPECIFY COMMAND
                */
 243   2              CALL IS$DRIVE$BUSY;
 244   2              DO IOPB$POINTER = 3400H TO 3402H;
 245   3                  OUTPUT(BASE$1) = IOx@
 246   3                  IF IOPB$POINTER < 3402H THEN
 247   3                      CALL OK$TO$OUTPUT$IOPB;
 248   3              END;
 249   2       END SPECIFY$COMMAND;

 250   1       END;


MODULE INFORMATION:

     CODE AREA SIZE     = 044BH   1099D
     VARIABLE AREA SIZE = 0022H     34D
     MAXIMUM STACK SIZE = 0004H      4D
     521 LINES READ
     0 PROGRAM ERROR(S)

END OF PL/M-80 COMPILATION
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0

 LOC  OBJ          LINE        SOURCE STATEMENT

                     1 START:
 1300                2 ORG     1300H
                     3 PUBLIC  RAEXT,CHECK
                     4 ;
                     5 ;THIS PROGRAM OUTPUTS THE IOPB TO THE iSBX-218 FOR THE READ,
                     6 ;WRITE, AND SCAN COMMANDS.
                     7 ;
                     8 ;THIS PROGRAM HAS BEEN CHECKED OUT ON THE iSBC-80/10B AND THE iSBC-80/24
                     9 ;
                    10 RAEXT:
                    11 ;
                    12 ;SAVE ALL REGISTER
                    13 ;
 1300 C5            14          PUSH    B
 1301 D5            15          PUSH    D
 1302 E5            16          PUSH    H
 1303 F5            17          PUSH    PSW
 1304 C31013        18          JMP             IOPB
 1310               19 ORG     1310H
                    20 ;
                    21 ;1310H, IN THIS LOCATION A JUMP TO 1370H IS PLACED. 10H IS ONE OF THE
                    22 ;POSSIBLE STATUS BYTES THAT CAN BE RETURN FROM THE MAIN STATUS REGISTER
                    23 ;REGISTERS USED ARE H & L, AND B. THE D & E REGISTERS ARE SET
                    24 ;TO THE FIRST LOCATION OF THE WRITE OR READ BUFFER.
                    25 ;
                    26 IOPB:
 1310 110030        27          LXI     D,3000H
 1313 21003E        28          LXI     H,3E00H             ;THE LOCATION OF THE IOPB IN USER RAM
 1316 0609          29          MVI     B,09H               ;THE B REGISTER IS LOADED WITH THE IOPB
                    30                                              ;COUNT
 1318 7E            31          MOV     A,M                 ;MOVE FIRST BYTE OF IOPB INTO ACC
 1319 D3F1          32          OUT     0F1H                ;OUTPUT TO THE FDC ON THE iSBX-218
 131B 2C            33          INR     L
 131C 05            34          DCR     B
                    35 REST:
 131D CD0314        36          CALL CHECK                  ;CHECK IF READY FOR NEXT BYTE OF IOPB
 1320 7E            37          MOV     A,M                 ;GET NEXT BYTE OF THE IOPB INTO THE ACC
 1321 D3F1          38          OUT     0F1H                ;OUTPUT TO THE FDC ON THE iSBX-218
 1323 2C            39          INR     L
 1324 05            40          DCR     B
 1325 C21D13        41          JNZ     REST                ;DONE OUTPUTING THE IOPB?
                    42 ;
                    43 ;SET THE H REG TO 13H THIS IS NEEDED WHEN DOING THE PCHL COMMAND
                    44 ;AT THE BRANCH ON STATUS
                    45 ;
 1328 2613          46          MVI     H,13H
 132A C37013        47          JMP     1370H               ;JUMP TO THE STATUS BRANCH
                    48 ;
                    49 ;1330H, AT THIS LOCATION A CHECK ON STATUS IS DONE. WHEN DOING A
                    50 ;WRITE OR A SCAN COMMAND, THE MAIN STATUS, (WHILE WAITING FOR A DATA BYTE)
                    51 ;RETURNS A 30H. THIS IS MOVED FROM THE ACCUMULATOR TO THE L REGISTER.
                    52 ;THEN THE PCHL INSTRUCTION IS EXE.
                    53 ;
 1330               54 ORG     1330H
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0

 LOC  OBJ          LINE        SOURCE STATEMENT

 1330 DBF0           55          IN      0F0H
 1332 6F             56          MOV     L,A
 1333 E9             57          PCHL
                     58 ;
                     59 ;1350H, AT THIS LOCATION AN ERROR OR TWO INDEX MARKS HAVE BEEN SEEN BY
                     60 ;THE 8272, AND TERMINATION OF THE COMMAND IS NEEDED.
                     61 ;
 1350                62 ORG      1350H
 1350 C3F713         63          JMP     FINSH
                     64 ;
                     65 ;1370H, AT THIS LOCATION A CHECK ON STATUS IS PREFORMED. THIS IS VERY
                     66 ;SIMILIAR TO THE ONE AT LOCATION 1330H.
                     67 ;
 1370                68 ORG 1370H
 1370 DBF0           69          IN      0F0H
 1372 6F             70          MOV     L,A
 1373 E9             71          PCHL
                     72 ;
                     73 ;13B0H, AT THIS LOCATION THE 8272 WANTS A BYTE OF DATA TO SEND TO THE
                     74 ;DISK DRIVE OR TO COMPARE WITH A BYTE FROM THE DISK DRIVE.
                     75 ;
 13B0                76 ORG      13B0H
 13B0 1A             77          LDAX    D
 13B1 D3F1           78          OUT     0F1H
 13B3 1C             79          INR     E
 13B4 C23013         80          JNZ     1330H
 13B7 C3F713         81          JMP     FINSH
                     82 ;
                     83 ;13D0H, AT THIS LOCATION THE 8272 HAS TERMINATED THE COMMAND OR THE
                     84 ;COMMAND WAS TERMINATED BY THE USER USING THE TC PULSE.
                     85 ;
 13D0                86 ORG      13D0H
 13D0 C3F713         87          JMP     FINSH
                     88 ;
                     89 ;13F0H, AT THIS LOCATION THE 8272 WANTS THE HOST iSBC BOARD TO READ A
                     90 ;BYTE FROM THE DATA PORT, AND WRITE IT IN MEMORY.
                     91 ;
 13F0                92 ORG      13F0H
 13F0 DBF1           93          IN      0F1H
 13F2 12             94          STAX    D
 13F3 1C             95          INR     E
 13F4 C27013         96          JNZ     1370H
                     97 ;
                     98 ;FINSH, THIS ROUTINE TERMINATES THE COMMAND BY OUTPUTING A TC PULSE
                     99 ;TO THE 8272, RESTORES THE REGISTER, AND RETURNS TO THE MAIN PROGRAM.
                    100 ;PORT E6 IS ONE PORT ON THE 80/10-B AND THE 80/24. THIS PORT IS USED
                    101 ;FOR MAKING THE TC PULSE. ONLY ONE BIT OF THE PORT NEEDS TO BE USED.
                    102 ;
                    103 FINSH:
 13F7 3EFF          104          MVI     A,0FFH
 13F9 D3E6          105          OUT     0E6H
 13FB AF            106          XRA     A
 13FC D3E6          107          OUT     0E6H
 13FE F1            108          POP     PSW
 13FF E1            109          POP     H
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0


  LOC  OBJ          LINE        SOURCE STATEMENT

  1400 D1            110          POP     D
  1401 C1            111          POP     B
  1402 C9            112          RET
                     113 ;
                     114 ;CHECK IS A ROUTINE THAT IS USED TO CHECK WHETHER OR NOT THE
                     115 ;8272 IS READY TO THE NEXT COMMAND BYTE.
                     116 ;
                     117 CHECK:
  1403 DBF0          118          IN      0F0H
  1405 FE90          119          CPI     90H
  1407 C20314        120          JNZ     CHECK
  140A C9            121          RET
                     122 END;


PUBLIC SYMBOLS
CHECK  A 1403    RAEXT  A 1300

EXTERNAL SYMBOLS


USER SYMBOLS
CHECK  A 1403    FINSH  A 13F7    IOPB   A 1310    RAEXT  A 1300    REST   A 131D    START  A 0000

ASSEMBLY COMPLETE,   NO ERRORS
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0

 LOC  OBJ          LINE        SOURCE STATEMENT

                     1 START:
 1420                2 ORG     1420H
                     3 PUBLIC  FOREXT
                     4 EXTRN   CHECK
                     5 ;
                     6 ;       THIS PROGRAM IS USED TO FORMAT A TRACK ON A 8" OR 5 1/4" DRIVE
                     7 ;       THE TRACK WILL HAVE 26 SECTORS WITH 128 OR 256 BYTES PER SECTOR
                     8 ;       DEPENDING ON WHETHER IT IS FM OR MFM FORMAT OR NOT.
                     9 ;
                    10 ;       THE A REG IS DESTROYED
                    11 ;       AND THE FLAGS ARE DESTROYED
                    12 ;
                    13 ;       CHECK IS AN EXTERNAL ROUTINE LOCATED IN REWR PROGRAM
                    14 ;
                    15 FOREXT:
                    16 ;
                    17 ;SAVE THE REGISTERS
                    18 ;
 1420 C5            19         PUSH    B
 1421 D5            20         PUSH    D
 1422 E5            21         PUSH    H
                    22 ;
                    23 ;LOAD THE LOCATION OF THE IOPB INTO THE H & L REGISTER PAIR.
                    24 ;
 1423 21003E        25         LXI     H,3E00H                 ;IOPB LOCATION
                    26 ;
                    27 ;THE NUMBER FOR THE FORMAT COMMAND IS FIVE.
                    28 ;5 IS LOADED INTO THE B REGISTER AND IS USED AS A
                    29 ;COUNTER
                    30 ;
 1426 0605          31         MVI     B,5H
                    32 ;
                    33 ;0B0H IS THE STATUS BYTE THAT IS LOOKED FOR WHEN WRITING
                    34 ;BYTES OF DATA TO THE 8272. 0B0H IS STORED IN THE E REGISTER
                    35 ;
 1428 1EB0          36         MVI     E,0B0H
                    37 ;
                    38 ;GO AND GO1 ROUTINES ARE USED TO OUTPUT THE IOPB TO THE
                    39 ;8272. THE BYTE TO BE OUTPUTED IS MOVED INTO THE ACC.
                    40 ;AND THEN OUTPUTED TO THE iSBX-218. NEXT THE L REGISTER IS
                    41 ;INCREMENTED BY ONE AND THE B REGISTER IS DECREMENTED
                    42 ;BY ONE.
                    43 ;
                    44 GO:
 142A 7E            45         MOV     A,M
 142B D3F1          46         OUT     0F1H                    ;OUTPUT FIRST COMMAND BYTE TO 8272
 142D 2C            47         INR     L
                    48 GO1:
 142E CD0000    E   49         CALL    CHECK                   ;CHECK TO SEE IF 8272 IS READY FOR NEXT COMMAND BYTE
 1431 7E            50         MOV     A,M                             ;MOVE NEXT BYTE INTO THE ACC.
 1432 D3F1          51         OUT      0F1H
 1434 2C            52         INR     L                               ;GET NEXT BYTE OF IOPB
 1435 05            53         DCR     B                               ;DECREMENT COUNTER FOR NUMBER OF COMMAND BYTES
 1436 C22E14        54         JNZ     GO1                             ;IF THE LAST BYTE CONTINUE/IF NOT WAIT FOR
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0

 LOC  OBJ          LINE        SOURCE STATEMENT

                     55                                              ;COMMAND PORT TO CHANGE STATUS
                     56 ;
                     57 ;WAIT TO WAIT4 ROUTINES ARE USED TO OUTPUT THE SECTOR INFORMATION TO
                     58 ;THE 8272. THE C, H, R, N VALUES ARE LOCATED AT RAM LOCATION 3D1C TO
                     59 ;3D1F IN USER RAM IN THIS EXAMPLE.
                     60 ;
 1439 061A           61          MVI     B,1AH                   ;LOAD COUNTER WITH THE NUMBER OF SECTOR ON A TRACK
 143B 211C3D         62          LXI     H,3D1CH
                     63 WAIT1:
 143E DBF0           64          IN      0F0H                            ;CHECK TO SEE IF READY TO SEND SECTOR INFO.
 1440 BB             65          CMP     E
 1441 C23E14         66          JNZ     WAIT1
 1444 7E             67          MOV     A,M
 1445 D3F1           68          OUT 0F1H                                ;OUTPUT INFO FOR SECTOR
 1447 2C             69          INR     L
                     70 WAIT2:
 1448 DBF0           71          IN 0F0H                 ;CHECK TO SEE IF READY TO SEND SECTOR INFO.
 144A BB             72          CMP E
 144B C24814         73          JNZ WAIT2
 144E 7E             74          MOV     A,M
 144F D3F1           75          OUT 0F1H                                ;OUTPUT INFO FOR SECTOR
 1451 2C             76          INR     L
                     77 WAIT3:
 1452 DBF0           78          IN  0F0H                ;CHECK TO SEE IF READY TO SEND SECTOR INFO.
 1454 BB             79          CMP     E
 1455 C25214         80          JNZ     WAIT3
 1458 7E             81          MOV     A,M
 1459 D3F1           82          OUT 0F1H
 145B 2C             83          INR     L
                     84 WAIT4:
 145C DBF0           85          IN      0F0H                    ;CHECK TO SEE IF READY TO SEND SECTOR INFO.
 145E BB             86          CMP     E
 145F C25C14         87          JNZ WAIT4
 1462 7E             88          MOV     A,M
 1463 D3F1           89          OUT     0F1H                    ;NUMBER BYTES PER SECTOR
                     90 ;
                     91 ;THIS NEXT PART GET THE H & L REGISTER PAIR BACK TO 3E1C AND ALSO INCREMENTS THE R VALUE
                     92 ;BY ONE.
                     93 ;
 1465 2D             94          DCR     L
 1466 34             95          INR     M
 1467 2D             96          DCR     L
 1468 2D             97          DCR     L
 1469 05             98          DCR B
 146A C23E14         99          JNZ WAIT1
                    100 ;
                    101 ;RESTORE THE REGISTERS AND RETURN THE THE MAIN PROGRAM
                    102 ;
 146D E1            103          POP H
 146E D1            104          POP D
 146F C1            105          POP      B
 1470 C9            106          RET
                    107 END
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0

PUBLIC SYMBOLS
FOREXT A 1420

EXTERNAL SYMBOLS
CHECK  E 0000

USER SYMBOLS
CHECK  E 0000    FOREXT A 1420    GO     A 142A    GO1    A 142E    START  A 0000    WAIT1  A 143E    WAIT2  A 1448
WAIT3  A 1452    WAIT4  A 145C

ASSEMBLY COMPLETE,   NO ERRORS
```