iSBC™ 208 FLEXIBLE DISK DRIVE CONTROLLER HARDWARE REFERENCE MANUAL

Order Number: 143078-001

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

A475/1081/ 6K DD

REV.	REVISION HISTORY	DATE
-001	Original issue.	10/81

PREFACE

This manual is the hardware reference for the iSBC 208 Flexible Disk Controller. The manual is divided into five chapters that describe general information, preparation for use, programming information, principles of operation, and service information. Three appendices, describing sample I/O drivers, the iSBX Multimodule interface, and drive interfaces are also included. Supplemental information can be found in the following Intel publications:

- *Intel Multibus Specification*, order number 9800683
- *iSBX Bus Specification*, order number 142686
- *iSBC Applications Manual*, order number 142687
- Intel *Component Data Catalog*
- *MCS-80/85 Family User's Manual*, order number 121506
- *The 8086 Family User's Manual*, order number 9800722
- *8080/8085 Assembly Language Programming Manual*, order number 9800940
- *MCS-86 Macro Assembly Language Reference Manual* order number 9800640

CONTENTS

CONTENTS (Cont'd.)

TABLES

ILLUSTRATIONS

CHAPTER 1
GENERAL INFORMATION

1-1. INTRODUCTION

The iSBC 208 Flexible Disk Controller is one product within a complete line of Intel iSBC single board computer expansion modules. The iSBC 208 controller is designed to interface up to four single- or double-sided, standard 8-inch floppy disk drives or four single- or double-sided 5¼-inch mini-floppy drives. The controller permits both single- and double-sided drives of the same size to be interfaced, and both single-density (FM) and double-density (MFM) recording formats to be used concurrently. The controller supports a soft-sector format with sector sizes ranging from 128 bytes to 4096 bytes in the IBM 3740-compatible single-density format and ranging from 256 bytes to 8192 bytes in the IBM system 34-compatible double-density format.

The iSBC 208 controller is designed expressly for Intel Multibus interface compatibility and can be inserted directly into a standard iSBC 604/614 cardcage as found in the iSBC System 80 series mainframe or into any of the Intel microcomputer development systems. All circuitry is contained on a single printed circuit board and operates from a single +5 volt source. A majority of the controller's logic is LSI (large scale integration) and includes both an Intel 8237 DMA Controller (DMAC) and an Intel 8272 Floppy Disk Controller (FDC). Additionally, data separation logic is included on the board to eliminate the necessity of this logic within the drive or off-board. The controller interfaces directly with any multibus-compatible single board computer. This computer, referred to in the remainder of this manual as the "host processor," provides all information required to perform a disk operation. Once all of the information is received, further host processor involvement is unnecessary, and the controller takes control of the bus for the duration of the data transfer. When the transfer is complete, the controller interrupts the host processor. When interrupted, the host processor examines the controller's status register to determine the outcome of the operation.

In addition to programmable sector sizes and recording density, the head load time, head unload time and track-to-track access time (step rate) operating characteristics also can be program specified. Additionally, a number of jumper-selectable options are provided to support various drive features and drive interface pin assignments. As shown in figure 1-1, the controller has two drive-interface connectors, a 50-pin connector for interfacing standard 8-inch drives and a 34-pin connector for interfacing 5¼-inch mini drives. A 36-pin connector is incorporated on the controller board for the installation of either a single- or double-wide iSBX Multimodule board. The controller extends Multibus capability to the Multimodule board and also provides up to two

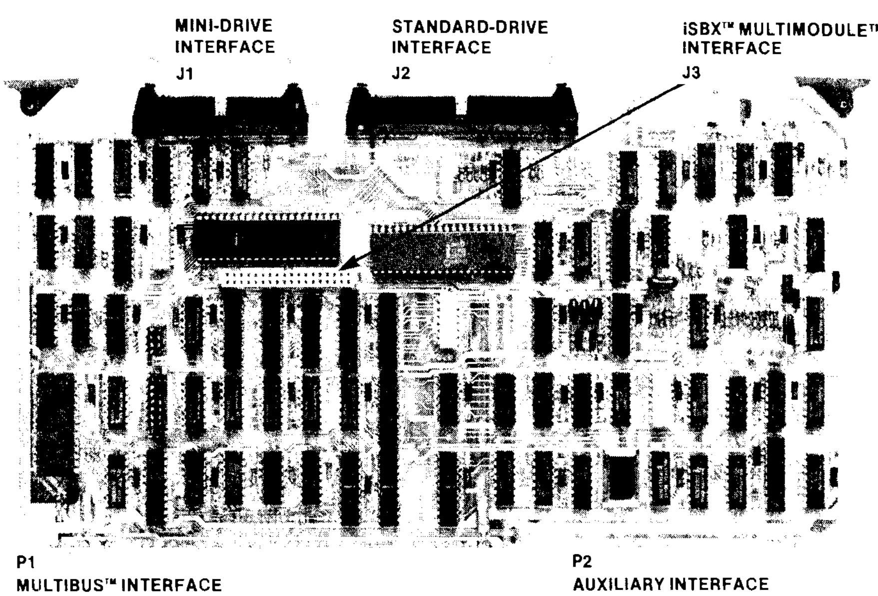


Figure 1-1. iSBC™ 208 Flexible Disk Drive Controller

143078-1

DMA channels for use by the iSBX board. The P2 auxiliary edge connector includes four address lines that extend the controller's memory addressing capability to 16 megabytes (24-bit address bus).

1-2. SPECIFICATIONS

Table 1-1 lists the physical and performance characteristics of the iSBC 208 controller.

Table 1-1. Specifications

Compatibility	
Host Processor	Any Intel mainframe, microcomputer development system or Multibus-compatible CPU. The controller supports either 16-, 20- or 24-bit addresses and an 8-bit data bus width.
Diskette Drive	Single- or double-sided, standard 8-inch or 5¼-inch mini drives. Up to four drives of one size can be interfaced; single- and double-density, and single- and double-sided drives can be mixed.
Drive Interface	Compatible with Shugart SA850 (standard 8-inch) and Shugart SA450 (5¼-inch mini) or any other drive with a similar interface.
Typical Drive Characteristics	
Transfer Rate Standard 8-inch Drive 250 kilobits per second, single density (FM) 500 kilobits per second, double density (MFM) 5¼-inch Mini Drive 125 kilobits per second, single density (FM) 250 kilobits per second, double density (MFM)	
Disk Speed 360 rpm (standard 8-inch) 300 rpm (5¼-inch mini)	
Track-to Track Access Time (Step Rate) Programmable from 1 to 16 ms in 1 ms steps (standard) or from 2 to 32 ms in 2 ms steps (mini).	
Head Load Time Programmable from 2 to 254 ms in 2 ms increments (standard) or from 4 to 508 ms in 4 ms increments (mini).	
Head Unload Time Programmable from 16 to 240 ms in 16 ms increments (standard) or from 32 to 480 ms in 32 ms increments (mini); jumper selectable for 1 second.	
Physical	
Dimensions Length: 30.48 cm (12.0 inches) Width: 17.15 cm (6.75 inches) Height: 1.27 cm (0.5 inches)	
Shipping Weight 0.82 kg (1.8 pounds)	
Power Requirements 5.0 volts (±5%), 3 amperes (maximum)	
Environmental Temperature: 0°C to +55°C, operating (+32°F to +131°F) −55°C to +85°C, non-operating (−67°F to +185°F) Humidity: Up to 90% relative humidity without condensation.	

Table 1-1. Specifications (Cont'd.)

Data Organization and Capacity (Standard 8-inch drives*)		
Single Density	**IBM Format**	**Non-IBM Format**
Bytes per Sector	128 256 512	1024 2048 4096
Sectors per Track	26 15 8	4 2 1
Tracks per Side	77 77 77	256 Addressable
Bytes per Side	256,256 (128-byte sector) 295,680 (256-byte sector) 315,392 (512-byte sector)	315,392 (77 tracks)
Double Density	**IBM Format**	**Non-IBM Format**
Bytes per Sector	256 512 1024	2048 4096 8192
Sectors per Track	26 15 8	4 2 1
Tracks per Side	77 77 77	256 Addressable
Bytes per Side	512,512 (256-byte sector) 591,360 (512-byte sector) 630,784 (1024-byte sector)	630,784 (77 tracks)

*Consult manufacturer's data for mini-floppy drive organization and capacity.

2-1. INTRODUCTION

This chapter presents information on the preparation and installation of the iSBC 208 Controller. Included within this chapter are instructions describing the unpacking and inspection, installation, board configuration, host processor bus interface and drive cabling for the controller.

2-2. UNPACKING AND INSPECTION

On receipt of the controller from the carrier, immediately inspect the shipping carton for evidence of mishandling in transit. If the shipping carton is damaged or waterstained, 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 if the contents of the carton are damaged, keep the carton and packing materials intact for the agent's inspection.

For repairs or replacement of an Intel product damaged in shipment, contact the Intel Technical Support Center (see Chapter 5) to obtain a Return Authorization Number and further instructions. A copy of the purchase order should be submitted to the carrier with the claim.

Carefully unpack the shipping carton and verify that the following items are included. Compare the packaging slip with your purchase order to verify that the order is complete. The carton and packing materials should be saved in case it becomes necessary to reship the controller at a later date.

Item 1: iSBC 208 Interface Printed Circuit Assembly.

Item 2: Schematic Diagram.

2-3. INSTALLATION CONSIDERATIONS

The controller is designed expressly for installation into the Intel iSBC 604/614 modular backplane and card cage as found in the Series 80 single board computer mainframes. The controller can also be installed into any odd-numbered slot in an Intellec Model 800 or in any slot in an Intellec microcomputer development system. The controller additionally can be installed into a user's Multibus-compatible backplane assembly that meets the controller's mating connector dimensional requirements.

2-4. POWER REQUIREMENTS

The controller operates from a single +5 volt (±5%) source and requires a maximum of 3.0 amperes. When installing the interface in an iSBC 80 Series, microcomputer development, or custom system, ensure that the system's power supply can meet the additional current requirements of the controller.

2-5. COOLING REQUIREMENTS

The iSBC 80 Series and Intellec microcomputer development systems use forced-air cooling that generally is adequate to maintain an internal operating temperature below 55°C. When installing the controller in a high-temperature environment or in any other system enclosure, ensure that the internal operating temperature is not permitted to exceed the 55°C maximum.

2-6. BUS INTERFACE

The controller communicates with the host processor (and memory) via the Multibus interface. Tables 2-1 and 2-2 define the Multibus interface pin assignments and corresponding signal definitions. The controller connects to the Multibus interface through connector P1, an 86-pin, double-sided printed circuit edge connector with 3.96mm (0.156 inch) contact centers.

2-7. MULTIBUS INTERFACE AC CHARACTERISTICS

Figures 2-1 and 2-2 show the Multibus interface ac timing characteristics when the controller is operating as a "bus master" (Bus Acquisition and Memory Transfer Timing) and as a "bus slave" (I/O Transfer Timing).

2.8 MULTIBUS INTERFACE DC CHARACTERISTICS

The controller's dc signal characteristics for the Multibus interface are given in table 2-3.

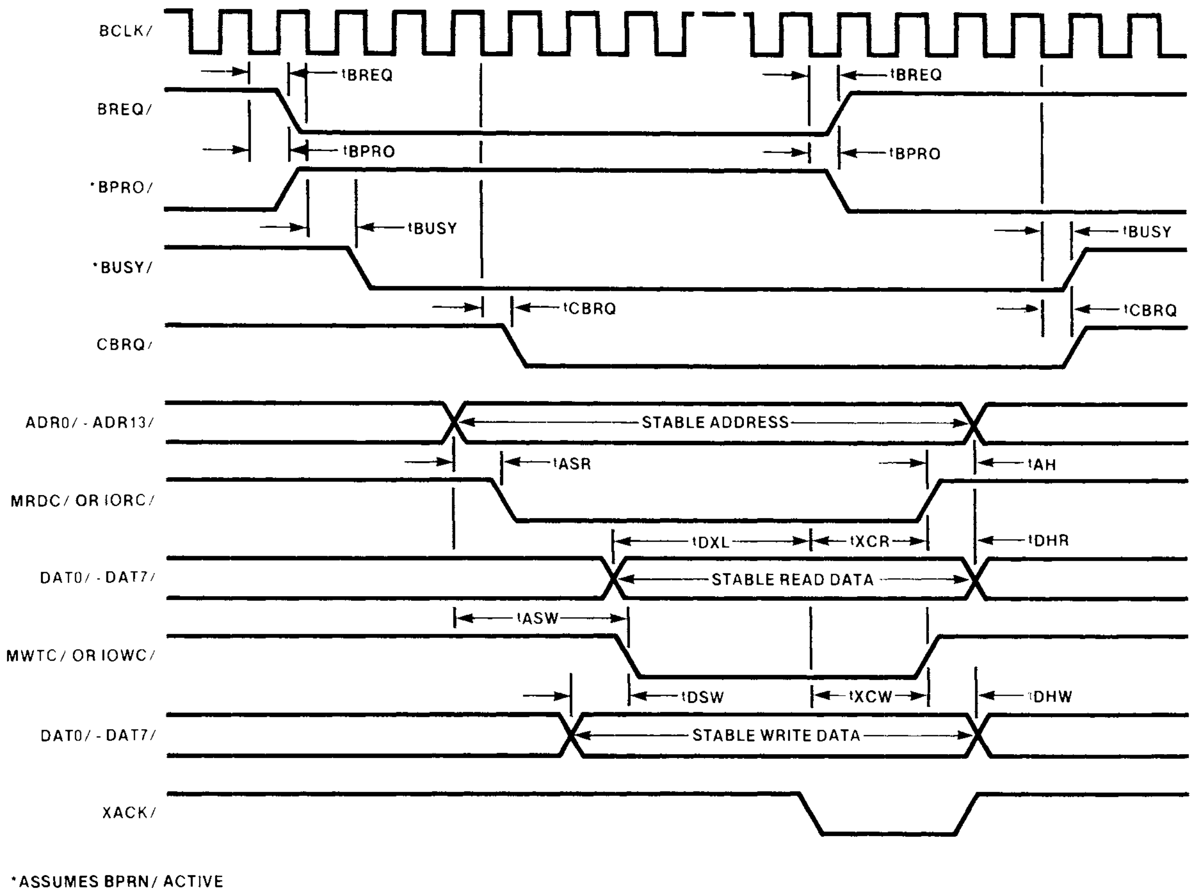
Table 2-1. Multibus Interface Pin Assignments

Pin*	Signal	Function	Pin*	Signal	Function
1	GND	Ground	44	ADRF/	Address Bus
2	GND	Ground	45	ADRC/	Address Bus
3	+5VDC	Power Inputs	46	ADRD/	Address Bus
4	+5VDC	Power Inputs	47	ADRA/	Address Bus
5	+5VDC	Power Inputs	48	ADRB	Address Bus
6	+5VDC	Power Inputs	49	ADR8/	Address Bus
7	+12VDC	Power Inputs	50	ADR9/	Address Bus
8	+12VDC	Power Inputs	51	ADR6/	Address Bus
9			52	ADR7/	Address Bus
10			53	ADR4/	Address Bus
11	GND	Ground	54	ADR5/	Address Bus
12	GND	Ground	55	ADR2/	Address Bus
13	BCLK/	Bus Clock	56	ADR3/	Address Bus
14	INIT/	Initialization	57	ADR0/	Address Bus
15	BPRN/	Bus Priority In	58	ADR1/	Address Bus
16	BPRO/	Bus Priority Out	59		
17	BUSY/	Bus Busy	60		
18	BREQ/	Bus Request	61		
19	MRDC/	Memory Read Command	62		
20	MWTC/	Memory Write Command	63		
21	IORC/	I/O Read Command	64		
22	IOWC/	I/O Write Command	65		
23	XACK/	Transfer Acknowledge	66		
24			67	DAT6/	Data Bus
25			68	DAT7/	Data Bus
26			69	DAT4/	Data Bus
27			70	DAT5/	Data Bus
28	ADR10/	Address Bus	71	DAT2/	Data Bus
29	CBRQ/	Common Bus Request	72	DAT3/	Data Bus
30	ADR11/	Address Bus	73	DAT0/	Data Bus
31	CCLK/	Constant Clock	74	DAT1/	Data Bus
32	ADR12/	Address Bus	75	GND	Ground
33			76	GND	Ground
34	ADR13/	Address Bus	77		
35	INT6/	Interrupt Request 6	78		
36	INT7/	Interrupt Request 7	79	-12VDC	Power Inputs
37	INT4/	Interrupt Request 4	80	-12VDC	Power Inputs
38	INT5/	Interrupt Request 5	81	+5VDC	Power Inputs
39	INT2/	Interrupt Request 2	82	+5VDC	Power Inputs
40	INT3/	Interrupt Request 3	83	+5VDC	Power Inputs
41	INT0/	Interrupt Request 0	84	+5VDC	Power Inputs
42	INT1/	Interrupt Request 1	85	GND	Ground
43	ADRE/	Address Bus	86	GND	Ground

*Unassigned Pins are reserved.

Table 2-2. Multibus Interface Signal Definitions

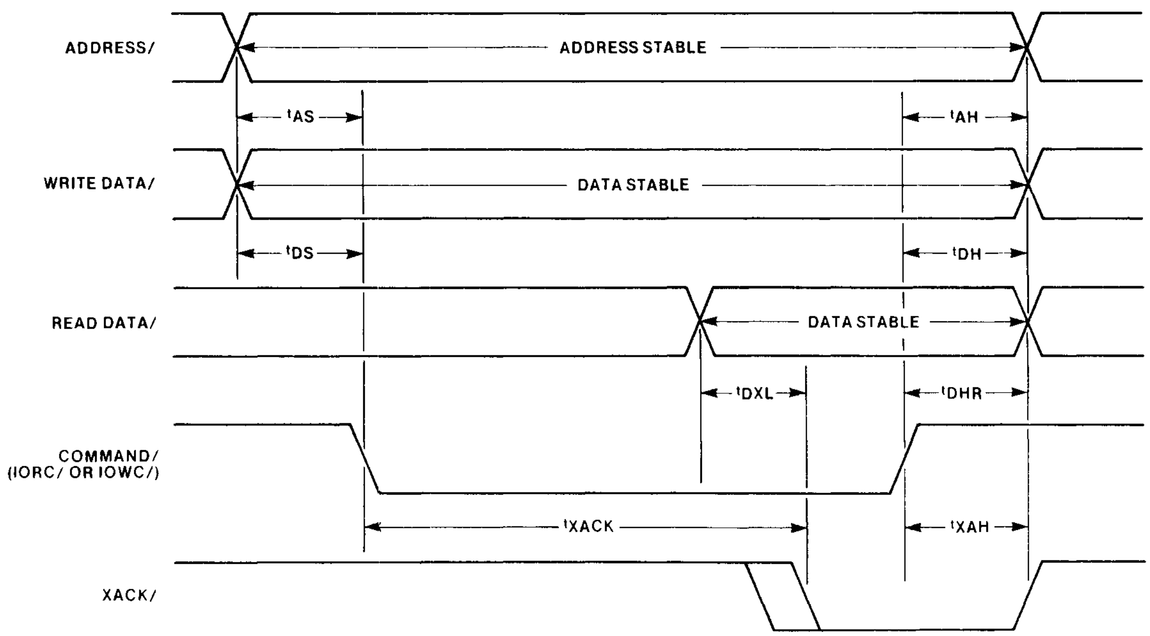
Signal	Function
ADR0/-ADRF/	Address. These 16 bidirectional lines specify the address of the memory location or I/O port to be accessed. ADRF/ is the most significant bit.
ADR10/-ADR13/	Extended Address. These four output lines extend the controller's memory addressing to 1 megabyte. ADR13/ is the most significant bit.
BCLK/	Bus Clock. This input signal is used to synchronize the controller's bus control logic.
BPRN/	Bus Priority In. This input signal level indicates that no higher-priority master board has requested control of the bus.
BPRO/	Bus Priority Out. This output signal level is used with serial priority resolution schemes and indicates to the next lower-priority master board that either the controller or another higher-priority master board has requested control of the bus.
BREQ/	Bus Request. This output signal is used with parallel priority resolution schemes and indicates that the controller is requesting control of the bus.
BUSY/	Bus Busy. This bidirectional signal indicates that either the controller or another master board is currently in control of the bus and consequently prevents any other master board from gaining access to the bus.
CBRQ/	Common Bus Request. This output signal indicates that the controller requires access to the bus while the bus is in the use by another bus master.
CCLK/	Constant Clock. A clock signal routed through the controller to the iSBX multimodule.
DAT0/-DAT7/	Data. These eight bidirectional lines transfer data either to or from the memory location or I/O port addressed. DAT7/ is the most significant bit.
INIT/	Initialization. This input signal generally originates from a power-up reset circuit or a contact closure to ground (i.e., a front panel reset switch) and resets all devices on the bus to an initialized state.
INT0/-INT7/	Interrupt. A set of eight, multi-level interrupt request lines for use with parallel interrupt resolution logic. The selected (jumper determined) output interrupt signal is used to indicate a controller-initiated interrupt request.
IORC/	I/O Read Command. This input signal instructs the controller to place the data associated with the addressed input port onto the data lines.
IOWC/	I/O Write Command. This input signal instructs the controller to accept the data associated with the addressed output port that is present on the data lines.
MRDC/	Memory Read Command. This output signal indicates that the address of a memory location is on the address lines and that the contents of that location are to be placed on the data lines for acceptance by the controller.
MWTC/	Memory Write Command. This output signal indicates that the address of a memory location is on the address lines and that the data presented by the controller on the data lines is to be written into that location.
XACK/	Transfer Acknowledge. This signal originates from the controller during I/O port transfers and indicates that the controller has accepted or is presenting the associated data on the data lines. During memory transfers, this signal originates from the random access memory board and indicates that the data on the data lines either has been written into the addressed memory location or that the data is present and is to be accepted by the controller.



Parameter	Minimum	Maximum	Description
t_{CBRQ}	67 ns		BCLK/ to CBRQ/ Delay
t_{BCY}	100 ns		Bus Clock Period
t_{BW}	35 ns		Bus Clock Pulse Width
t_{BREQ}		35 ns	BCLK/ to BREQ/ Delay
t_{BPRNS}	22 ns		BPRN/ to BCLK/ Setup Time
t_{BPRO}		40 ns	BCLK/ to BPRO/ Delay
t_{BPRNO}		30 ns	BPRN/ to BPRO/ Delay
t_{BUSY}		55 ns	BCLK/ to BUSY/ Low Delay
t_{ASR}	286 ns		Address Setup Time (Read)
t_{AH}	147 ns		Address Hold Time
t_{DXL}	−250 ns		Data Setup to Acknowledge Time (Read)
t_{XCR}	567 ns	1327 ns	Acknowledge to Command High (Read)
t_{DHR}	−80 ns		Data Hold Time (Read)
t_{ASW}	786 ns		Address Setup Time (Write)
t_{DS}	80 ns		Data Setup to Command Time (Write)
t_{XCW}	567 ns	1257 ns	Acknowledge to Command High (Write)
t_{DHW}	65 ns		Data Hold Time (Write)
t_{INIT}	4 ns		Reset Pulse Width

Figure 2-1. Bus Acquisition and Memory Transfer Timing

121746-2



Parameter	Minimum	Maximum	Description
t_{AS}	−384 ns		Address Setup Time
t_{AH}	50 ns		Address Hold Time
t_{XACK}	906 ns	1100 ns	Command to Acknowledge
*t_{XACK}	4400 ns	5100 ns	
t_{DXL}	2 ns		Read Data Setup Time
t_{DHR}	20 ns		Read Data Hold Time
t_{XAH}	61 ns		Acknowledge Hold Time
t_{DS}	−189 ns		Write Data Setup Time
t_{DH}	35 ns		Write Data Hold Time

*Software RESET command only.

Figure 2-2. I/O Transfer Timing

143078-2

Table 2-3. iSBC 208 Board DC Characteristics

Signals	Symbol	Parameter Description	Test Conditions	Min	Max	Units
XACK/	V_{OL}	Output Low Voltage	I_{OL}=32 mA		.04	V
	V_{OH}	Output High Voltage	I_{OH}=−5.2 mA	2.0		V
	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN}=0.4V		−1.2	mA
	I_{IH}	Input Current at High V	V_{IN}=2.4V		40	μA
	*C_L	Capacitive Load			15	pF
ADR0/-ADRF/	V_{OL}	Output Low Voltage	I_{OL}=32 mA		0.45	V
	V_{OH}	Output High Voltage	I_{OH}=−5 mA	2.4		V
	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN}=−0.45 V		−0.2	mA
	I_{IH}	Input Current at High V	V_{IN}=5.25V		50	μA
	*C_L	Capacitive Load			18	pF
BCLK/, BPRN/	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN}=0.45V		−0.5	mA
	I_{IH}	Input Current at High V	V_{In}=5.25V		100	μA
	*C_L	Capacitive Load			15	pF
ADR10/-ADR13/	V_{OL}	Output Low Voltage	I_{OL}=24 mA		0.4	V
	V_{OH}	Output High Voltage	I_{OH}=−15 mA	2.4		V
	V_{LL}	Output Leakage Low			20	μA
	I_{LH}	Output Leakage High			20	μA
ADR14/-ADR17/ (on P2)	*C_L	Capacitive Load			15	pF
BPRO/	V_{OL}	Output Low Voltage	I_{OL}=3.2 mA		0.45	V
	V_{OH}	Output High Voltage	I_{OH}=−0.4 mA	2.4		V
	*C_L	Capacitive Load			15	pF
BREQ/	V_{OL}	Output Low Voltage	I_{OL}=20 mA		0.45	V
	V_{OH}	Output High Voltage	I_{OH}=−0.4 mA	2.4		V
	*C_L	Capacitive Load			10	pF
BUSY/ (Open Collector)	V_{OL}	Output Low Voltage	I_{OL}=20 mA		0.45	V
	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	V_{IL}	Input Current at Low V	V_{IN}=0.45V		−0.5	mA
	I_{IH}	Input Current at High V	V_{IN}=5.25V		100	μA
	*C_L	Capacitive Load			20	pF
DAT0/-DAT7/	V_{OL}	Output Low Voltage	I_{OL}=32 mA		0.45	V
	V_{OH}	Output High Voltage	I_{OH}=−5 mA	2.4		V
	V_{IL}	Input Low Voltage			0.80	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN}=0.45V		−0.20	mA
	I_{LH}	Output Leakage High	V_O=5.25V		100	μA
	*C_L	Capacitive Load			18	pF
CBRQ/ (Open Collector)	V_{OL}	Output Low Voltage	I_{OL}=60 mA		0.8	V
	*C_L	Capacitive Load			15	pF
IORC/, IOWC/, INIT/, CCLK/	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN}=0.4V		−1.2	mA
	I_{IH}	Input Current at High V	V_{IN}=2.4V		40	μA
	*C_L	Capacitive Load			18	pF

Table 2-3. iSBC™ 208 Board DC Characteristics (Cont'd.)

Signals	Symbol	Parameter Description	Test Conditions	Min	Max	Units
INT0/-INT7/	V_{OL}	Output Load Voltage	I_{OL}=60 mA		0.45	V
	V_{OH}	Output High Voltage	Open Collector			
	I_{LH}	Output Leakage High	V_O=5.25V		250	μA
	I_{LL}	Output Leakage Low	V_O=0.45V		−500	μA
	*C_L	Capacitive Load			15	pF
MRDC/, MWTC/	V_{OL}	Output Low Voltage	I_{OL}=32 mA		0.45	V
	V_{OH}	Output High Voltage	I_{OH}=−2 mA	2.4		V
	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_O	Output Leakage Current	V_{IN}=0.45V		−100	μA
			V_{IN}=5.25		100	μA
	*C_L	Capacitive Load			25	pF

*Capacitive load values are approximations.

2-9. AUXILIARY CONNECTOR

The auxiliary connector (P2) provides the four 1-megabyte paging bits to effectively allow the controller to address up to 16 megabytes. The bits are set in the controller's auxiliary port and are routed to the P2 connector as noted in table 2-4.

2-10. BOARD LOCATION CONSIDERATIONS

Since the controller functions as a bus master during DMA transfers, when installing the controller in a serial priority environment (e.g., within any of the Intel Series 80 mainframes), the controller should occupy the highest priority slot (top physical slot) in the 604/614 backplane and card cage assembly, with any other bus masters and the host processor board located below. The backplane provides bus priority in and out signal continuity among adjacent bus masters. The BPRN/ (Bus Priority In) input to the top slot (J2) of either the single (604) or expansion (614) backplane must be connected to logic ground. Both backplanes provide the BPRN/ input on a wire-wrap terminal post. As shown in the following illustration (figure 2-3), a wire-wrap jumper must be installed from terminal post B (BPRN/) to logic ground at terminal post N (604) or terminal post L (614).

CAUTION

Always remove system power prior to installing or removing a board in the backplane. Failure to observe this precaution can result in circuit damage.

Note that if a bus slave (e.g., a memory board) is installed between two bus masters (or if a vacant slot exists between two bus masters), the serial priority input-output chain must be physically jumpered on the backplane to maintain signal continuity. Figure 2-3 shows the installation of a jumper between terminal posts C and E that would provide the required BPRO/-BPRN/ continuity around a "slave" installed in the second slot (J3).

When installing the controller in a parallel priority resolution environment, the controller should be given the highest bus priority. In an Intellec Model

Table 2-4. P2 Bus Connector

Pin	Signal	Function	DC Characteristics (each signal)			
			Current Drive		Current Load	
			Low (I_{OL})	High (I_{OH})	Low (I_{IL})	High (I_{IH})
56	ADR17/	High-Order Page Address Bit	24mA	−15mA	0	0
55	ADR16/					
58	ADR15/					
57	ADR14/	Low-Order Page Address Bit				

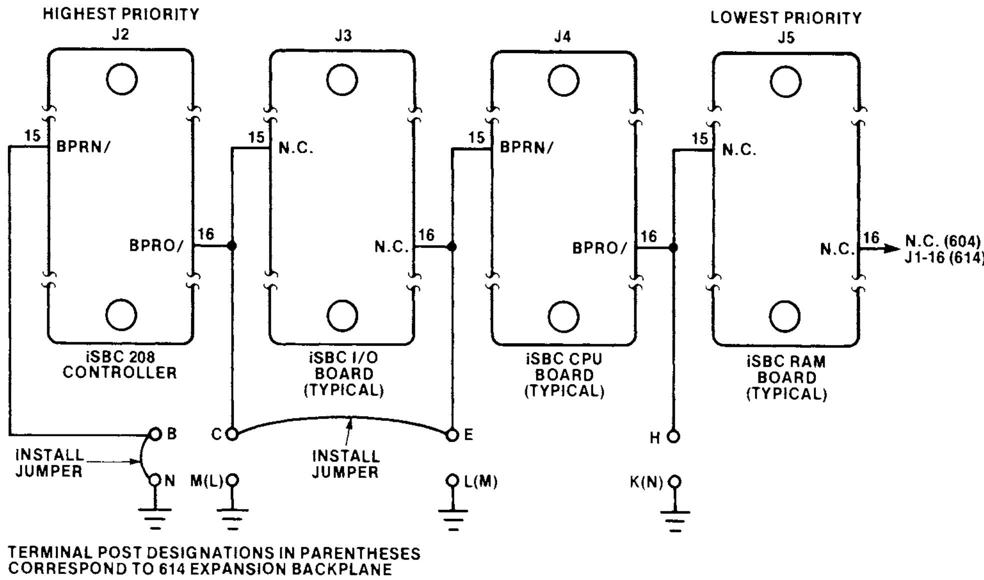


Figure 2-3. Serial Priority Resolution 143078-3

800 development system, the controller must be installed in an odd-numbered (bus master) slot and ideally should be installed in slot 17 (highest bus priority). In an Intellec Series II or Series III development system, the controller should be installed in the bottom (highest bus priority) slot.

2-11. CONTROLLER BOARD CONFIGURATION

The controller board includes alterable jumpers that are used to configure the controller to its intended system environment. The jumpers can be divided into three major groups: host processor configuration, drive configuration, and auxiliary port configuration. The locations of the jumpers are shown in figure 5-1. Note that the controller jumpers associated with the iSBX Multimodule interface are described in Appendix B.

2-12. HOST PROCESSOR CONFIGURATION

The jumpers associated with the host processor interface are used to specify the I/O address bit length, the I/O base address of the controller, parallel or serial bus priority resolution, and Multibus interface interrupt level selection.

The I/O address bit length (8 or 16 bits) is determined by the jumper link at E41-E45-E49. When shipped from the factory, a push-on shorting plug is installed between E45 and E49 to select 8-bit I/O address decoding. To implement 16-bit I/O address decoding, remove the shorting plug connecting E45 and E49 and install the plug between E41 and E45.

The controller's I/O base address is specified by a set of jumpers that provides either a high ("1") or low ("0") input to the I/O address decode comparators.

Depending on the I/O address bit length selected (8 or 16 bits), either three (8-bit addressing) or all eleven (16-bit addressing) jumpers must be configured. When shipped from the factory, all of the I/O base address shorting plugs are in the "0" position (corresponding to a 16-bit I/O base address of 0000H). To relocate the I/O base address, reposition the shorting plugs according to table 2-5. As an example, to select an I/O base address of F800H, address bits F, E, D, C, and B would be jumpered to the "1" position, and the remaining address bits would be jumpered to the "0" position. 8-bit addressing allows base addresses from 00H to E00H, while 16- bit addressing gives addresses from 0000H to FFE0H.

Table 2-5. I/O Base Address Selection

Address Bit	Shorting - Plug Position	
	"1"	"0"
5	E42-E46	E46-E50
6	E43-E47	E47-E51
7	E44-E48	E48-E52
8*	E53-E61	E61-E69
9*	E54-E62	E62-E70
A*	E55-E63	E63-E71
B*	E56-E64	E64-E72
C*	E57-E65	E65-E73
D*	E58-E66	E66-E74
E*	E59-E67	E67-E75
F*	E60-E68	E68-E76

*Only required for 16-bit I/O addressing.

Parallel/serial bus priority resolution is determined by jumper E77-E78. The controller is configured at the factor for serial bus priority resolution (jumper installed between E77 and E78) as found in the Intel System 80 mainframes. To select parallel bus priority resolution (e.g., when installing the controller in an Intellec microcomputer development system), remove the jumper between E77 and E78.

The controller's Multibus interface interrupt level is selected by installing a jumper from E79 to one of the eight Multibus interface lines on E82 through E89. The following list defines the interrupt/jumper correspondence.

Jumpers	Interrupt Level
E79-E89	INT0/
E79-E88	INT1/
E79-E87	INT2/
E79-E86	INT3/
E79-E85	INT4/
E79-E84	INT5/
E79-E83	INT6/
E79-E82	INT7/

Note that an interrupt level jumper is not installed at the factory and that the interrupt level selected must *not* have been previously assigned to another bus master.

2-13. DRIVE CONFIGURATION

The jumpers associated with drive configuration are used to define both the controller pin assignments on the drive interface connectors and the type of drive being interfaced (mini or standard) as well as to support optional features within the drive. Table 2-6 defines the usual functions of the drive configuration jumper links; any unused jumper associated with the interface connectors can be used to implement other functions within the drive or to reassign pin assignments for radial signals when interfacing multiple drives.

Table 2-6. Drive Configuration Jumper Links

Function	Jumper Posts	Factory Configuration	Description
FAULT RESET/	E27,E28	Removed	When this jumper link is installed, the controller provides a FAULT RESET/ output on J2-50 during read/write operations. This output is used to reset optional fault detection circuitry within a drive.
LOW CURRENT/	E25,E26	Installed	With this jumper link installed, the controller provides a LOW CURRENT/ output on J2-2 during read/write operations whenever the track address is 43 or greater (to reduce write current on the inner tracks). If the drive interfaced does not support low write current compensation, remove the jumper link between E25 and E26.

Table 2-6. Drive Configuration Jumper Links (Cont'd.)

Function	Jumper Posts	Factory Configuration	Description
READY/	E17,E18,E19	E18-E19	A jumper link is installed between E18 and E19 (factory configuration) when the drive interfaced provides a READY/ signal to the controller on J2-22 or J1-6. When a drive does not provide a READY/ signal (most mini-sized drives do not provide this signal), remove the jumper link between E18 and E19 and install a jumper link between E17 and E19.
TWO SIDED/	E21,E22	Installed	With this jumper installed, the TWO SIDED/ status signal from a drive is available to the controller on J2-10 or J1-34. When all of the drives interfaced are single-sided, this jumper link *can* be omitted.
FAULT/	E23,E24	Removed	When this jumper link is installed, the optional FAULT/ status signal from a drive is available to the controller on J2-48.
Mini/Standard	E4,E5	Removed	This jumper link identifies the type of drive (mini or standard) interfaced to the controller. With the jumper link removed, the controller is configured for standard 8-inch drives. When interfacing mini-sized drives, install the jumper link between E4 and E5.
HEAD LOAD/	E29 thru E40	E31-E32, E38-E39	In the factory configuration (jumper links E31-E32 and E38-E39 installed), a common HEAD LOAD/ signal is output (on J2-18) to all drives interfaced. The head load and head unload time intervals associated with the HEAD LOAD/ signal are user programmable. Individual (radial) HEAD LOAD/ signals for each drive can be made available at the J2 connector by removing the jumper link between E38 and E39 and installing the following jumper links: E37 to E38 E39 to E40 E29 to E30 E35 to E36 E33 to E34 In this configuration, the programmed head load interval remains unchanged, but the programmed head unload interval is increased by 1 second (fixed) to decrease wear on the head load mechanism during heavy usage. The HEAD LOAD/ jumper link matrix also allows a common HEAD LOAD/ signal (on J2-18) with the additional 1 second head unload delay. This configuration is implemented by installing the following jumper links: E37 to E38 E39 to E40 E34 to E36 E36 to E30 E30 to E32 E32 to E31

Table 2-6. Drive Configuration Jumper Links (Cont'd.)

Function	Jumper Posts	Factory Configuration	Description
Mini Drive Select	E20	Removed	When shipped from the factory, the controller does not provide a DRIVE SELECT 3/ signal on mini-drive interface connector J1. To interface four mini drives, the DRIVE SELECT 3/ signal on jumper post E20 must be connected to one of the jumper posts corresponding to an unused pin on the J1 connector. Depending on the functions supported by the mini drive, the following jumper posts may be available: E18 (READY/ input from drive on J1-6) E21 (TWO SIDED/ input from drive on J1-34) Also, any unassigned J1 connector pin in the auxiliary port matrix can be used (see Section 2-14).

2-14. AUXILIARY PORT CONFIGURATION

The auxiliary port jumper links form a matrix that includes four jumper posts on the low-order four bits of the controller's auxiliary I/O port and three jumper posts on specific pins of drive interface connectors J2 and J1. By interconnecting auxiliary port and connector pin jumper posts, special drive functions and signals can be defined through the auxiliary port. The primary function of the port is to provide MOTOR ON/ signals to mini-sized drives. Table 2-7 defines the jumper posts in the auxiliary port matrix.

Table 2-7. Auxiliary Port Jumper Matrix

Jumper Post	Auxiliary Port Assignment	Jumper Post	Interface Connector Pin Assignment
E11	Bit 0	E10	J1-2, J2-8
E9	Bit 1	E8	J1-4, J2-12
E7	Bit 2	E6	J1-16*, J2-16
E2	Bit 3		

*J1-16 is defined as the MOTOR ON/ signal pin on the Shugart drive interface.

2-15. DRIVE INTERFACING

The iSBC 208 controller can interface up to four single- or double-sided, standard 8-inch or 5¼-inch mini-sized drives. The controller includes two drive interface connectors, a 50-pin connector (J2) for interfacing standard-sized drives and a 34-pin connector (J1) for interfacing mini-sized drives. Figure 2-4 depicts a typical four-drive system.

2-16. CONTROLLER INTERFACE SIGNALS

The individual pin assignments for the J2 and J1 drive interface connectors are given in tables 2-8 and 2-9, respectively. Table 2-10 describes the individual signal functions.

2-17. DRIVE INTERFACE AC CHARACTERISTICS

The drive interface ac timing characteristics are shown in the following timing diagrams (figures 2-5 through 2-7); the individual timing values are given in table 2-11.

2-18. DRIVE INTERFACE DC CHARACTERISTICS

The drive interface dc signal characteristics are given in table 2-12. Note that all controller output signals are open collector and that all input signals are terminated on the controller with 220/330 ohm resistor networks.

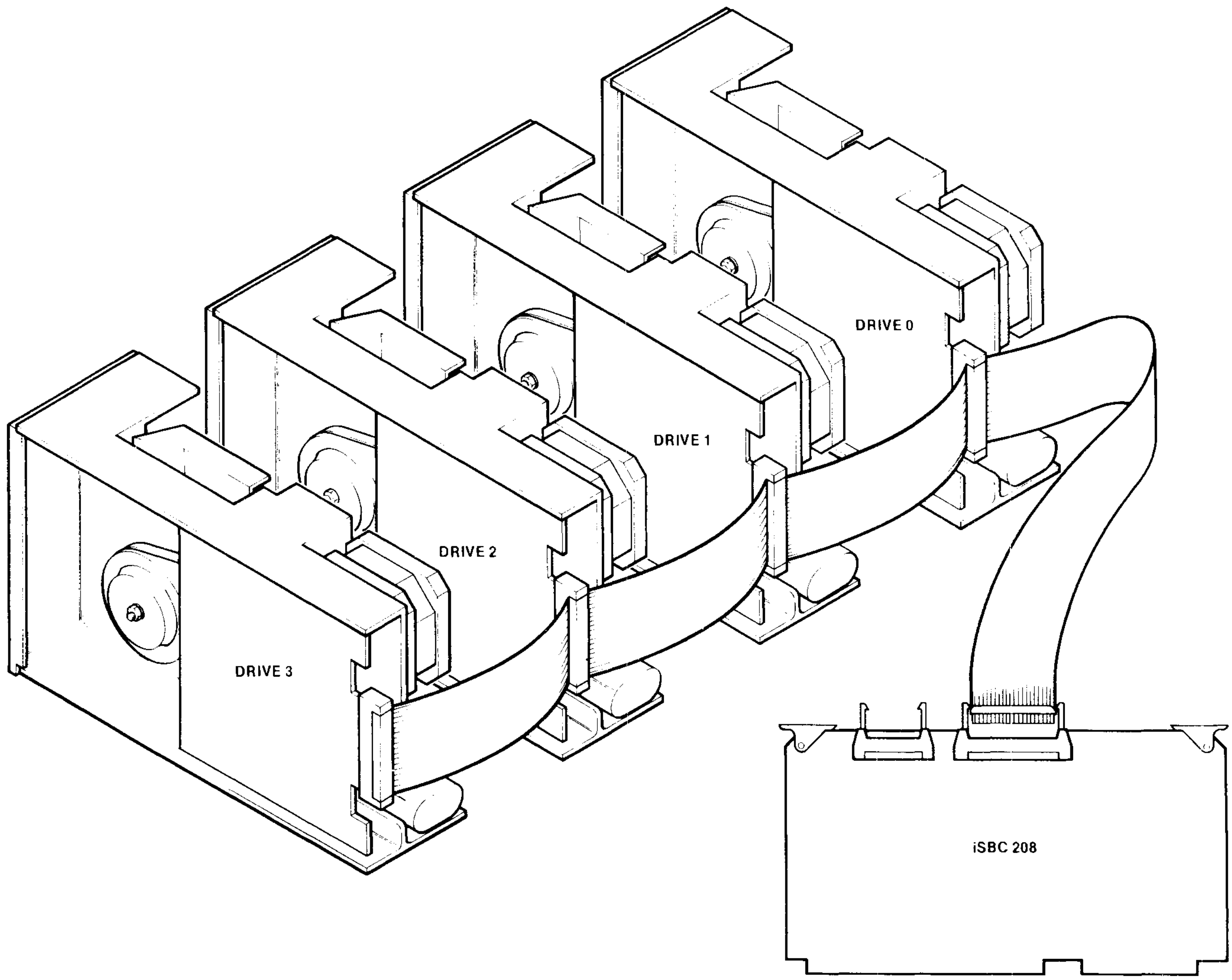


Figure 2-4. Typical Four-Drive System (Standard-Sized Drives)

143078-4

Table 2-8. J2 Interface Connector Pin Assignments

Pin Assignment	Signal	Pin Assignment	Signal
2	LOW CURRENT/	28	DRIVE SELECT 1/
4	HEAD LOAD 2/	30	DRIVE SELECT 2/
6	HEAD LOAD 3/	32	DRIVE SELECT 3/
8	Spare	34	DIRECTION/
10	TWO SIDED/	36	STEP/
12	Spare	38	WRITE DATA/
14	SIDE SELECT/	40	WRITE GATE/
16	Spare	42	TRACK 0/
18	HEAD LOAD 0/	44	WRITE PROTECT/
20	INDEX/	46	READ DATA/
22	READY/	48	FAULT/
24	HEAD LOAD 1/	50	FAULT RESET/
26	DRIVE SELECT 0/		

Note that all odd-numbered pins are connected to logic ground.

Table 2-9. J1 Interface Connector Pin Assignments

Pin Assignment	Signal	Pin Assignment	Signal
2	Spare	20	STEP/
4	Spare	22	WRITE DATA/
6	READY/	24	WRITE GATE/
8	INDEX/	26	TRACK 0/
10	DRIVE SELECT 0/	28	WRITE PROTECT/
12	DRIVE SELECT 1/	30	READ DATA/
14	DRIVE SELECT 2/	32	SIDE SELECT/
16	Spare	34	TWO SIDED/
18	DIRECTION/		

Note that all odd-numbered pins are connected to logic ground.

Table 2-10. Interface Connector Signal Functions

Signal	Function
LOW CURRENT/	A low-state active output signal used to select low write current compensation circuitry available in some drives. This signal is enabled during read/write operations and is active (low) when the track address is 43 or greater. Note that a factory-installed jumper link is used to route this signal to pin 2 of connector J2.
HEAD LOAD 2/	An optional (jumper selectable) low-state active output signal used to load the read/write head in drive 2. When the head is initially loaded, the controller provides a programmed delay (head load time) prior to initiating any read/write operation. Following a read/write operation, the controller delays inactivating the HEAD LOAD 2/ signal until the programmed head unload time and the one-second fixed delay intervals time out. Note that a jumper link must be installed to route the HEAD LOAD 2/ signal to the J2 interface connector.
HEAD LOAD 3/	An optional low-state active output signal that is functionally identical to HEAD LOAD 2/ except routed to drive 3.
TWO SIDED/	A low-state active status input signal that indicates the installation of a double-sided diskette within the drive. Note that a factory-installed jumper link is used to route this signal into the controller from drive interface connectors J2 and J1, and that this signal is only examined during the Sense Drive Status command.
SIDE SELECT/	An output control signal that selects one side of a double-sided drive. When SIDE SELECT is low, read/write operations are performed on side 1 of the drive.
HEAD LOAD 0/	A low-state active output signal used to load the read/write head in drive 0. When configured at the factory, this signal is the only HEAD LOAD/ signal available on interface connector J2 (common HEAD LOAD/ signal for all drives interfaced), and the additional one-second head unload delay is not used.
INDEX/	A low-state active input pulse that is coincident with the detection of the index hole in the diskette (indicates the logical beginning of a track).
READY/	A low-state active input signal indicating that the drive is ready to perform an operation. The qualifications for READY/ are drive dependent and usually include diskette in place, door closed and diskette rpm at specified speed. The controller uses a common READY/ input and requires that the drives interfaced provide a gated READY/ output when individually selected.
HEAD LOAD 1/	An optional low-state active output signal that is functionally identical to HEAD LOAD 2/ except routed to drive 1.
DRIVE SELECT 0/ DRIVE SELECT 1/ DRIVE SELECT 2/ DRIVE SELECT 3/	Individual low-state active output signals for selecting the individual drives interfaced. Note that a DRIVE SELECT 3/ signal is not included on the J1 interface connector and that when interfacing four mini drives, this signal must be connected to one of the jumper posts associated with an unassigned pin of connector J1.
DIRECTION/	An output control signal that specifies the direction in which the drive's read/write head is stepped. This signal is only enabled during seek operations and when at a logic low level, causes the head to be stepped toward the spindle (step in).
STEP/	A low-state active output pulse that causes the drive to move (step) the read/write head one track position. The direction that the head is stepped is determined by the state of the DIRECTION/ output signal. Like the DIRECTION/ signal, STEP/ is only enabled during seek operations.
WRITE DATA/	The serial data/clock composite write signal to the drive. The high-to-low-going transition of this signal indicates a bit to be written on the diskette.
WRITE GATE/	A low-state active control signal that is used to enable the drive's write electronics (allowing data to be written on the diskette). When this signal is in its inactive state, the write electronics are disabled, and the drive reads data from the diskette.
TRACK 0/	A low-state active input status signal that indicates the drive's read/write head currently is positioned over track 0. Note that this signal is only examined during a seek or recalibrate operation.
WRITE PROTECT/	A low-state active input status signal that indicates the installation of a write-protected diskette in the drive. Note that this signal is only examined during a write or format operation.
READ DATA/	The composite (unseparated) data and clock input signal generated by the drive during a diskette read operation. A high-to-low-going transition indicates a clock or data "one" bit.
FAULT/	An optional low-state active input signal that indicates a write fault condition within the drive. This signal is only examined during read/write operations and requires the installation of a jumper link to route the signal into the controller from the J2 interface connector.
FAULT RESET/	A low-state active output control signal that is used to reset fault detection logic optional in some drives. This signal is automatically generated at the beginning of every read/write operation and requires the installation of a jumper link to route the signal to the J2 interface connector.

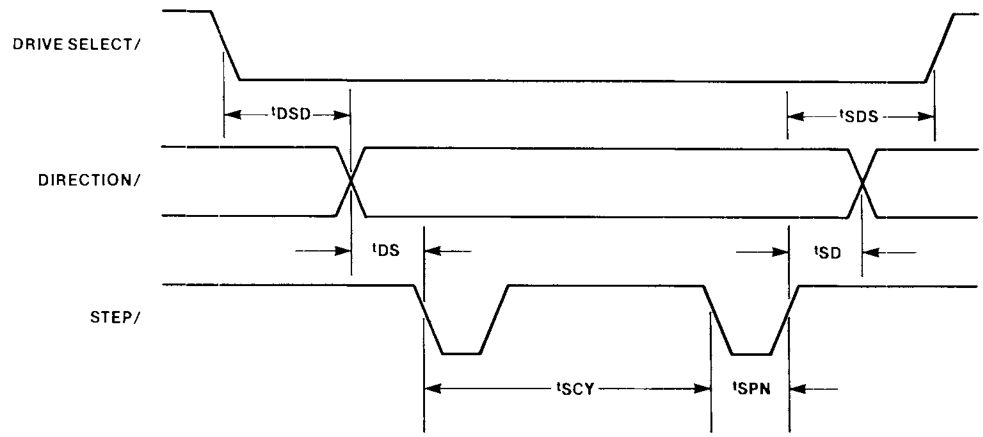


Figure 2-5. Seek Timing 143078-5

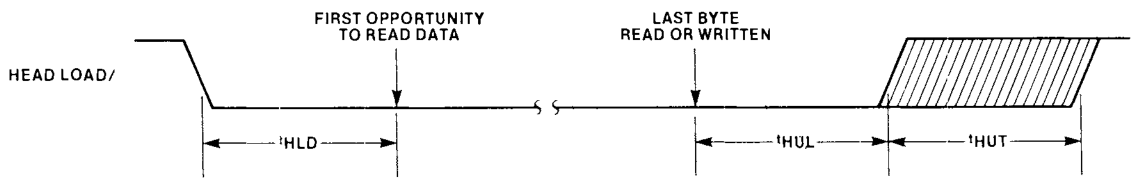


Figure 2-6. Head Load Timing 143078-6

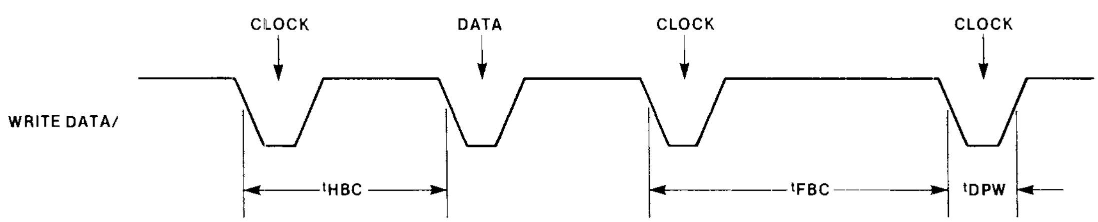


Figure 2-7. Write Data Timing 143078-7

Table 2-11. Drive Interface AC Timing Characteristics

Symbol	Parameter	Standard 8-inch Drive			5¼-inch Mini Drive			Units
		Minimum	Typical	Maximum	Minimum	Typical	Maximum	
Seek Timing								
t_{DSD}	DRIVE SELECT/ to DIRECTION/ Setup Time	19			38			μs
t_{SDS}	DRIVE SELECT/ Hold Time from STEP/	5			10			μs
t_{DS}	DIRECTION/ to STEP/ Setup Time	1			2			μs
t_{SD}	DIRECTION/ Hold Time from STEP/	24			48			μs
t_{SCY}	STEP/ Cycle Time	1		16	2		32	ms
t_{SPW}	STEP/ Pulse Width	5			10			μs
Head Load Timing								
t_{HLD}	Head Load Time	2		254	4		508	ms
t_{HUL}	Head Unload Time	16		240	32		480	ms
t_{HUT}	Head Unload Time-Out (Optional)		1			1		s
Write Data Timing								
t_{HBC}	Half Bit Cell		1 or 2*			2 or 4*		μs
t_{FBC}	Full Bit Cell		2 or 4*			4 or 8*		μs
t_{DPW}	Data Pulse Width	200	250		200	250		ns

*FM Mode Values.

Table 2-12. Drive Interface DC Characteristics

Signal	Current Drive		Current Load	
	I_{OL}	I_{OH}	I_{IL}	I_{IH}
All Output Signals*	48mA	−250μA	—	—
READY/, INDEX/, READ DATA/	—	—	−0.8mA	40μA
WRITE PROTECT/, TWO SIDED/, FAULT/, TRACK 0/	—	—	−0.2mA	20μA

*Auxiliary port output signals have an additional 10k ohm pullup resistor to V_{CC}.

2-19. DRIVE CABLING

The controller uses two drive interface connectors, a 34-pin connector for interfacing mini-sized drives (J1) and a 50-pin connector for interfacing standard-sized drives (J2). Each interface connector can interface up to four drives using a daisy-chain technique. Since most drives compatible with the controller follow the Shugart flexible disk drive interface requirements, flat ribbon cable and mass-termination type connectors are recommended for cable fabrication. (A number of the individual drive interface signal pin assignments can be altered or defined by jumpers on the controller board.) To fabricate the I/O interface cable, the cable ends are fitted with the appropriate mating connectors, and when interfacing multiple drives, additional drive mating connectors are inserted directly into the cable to form a daisy-chain cable. The recommended maximum cable length between the controller and the (last) drive is 10 feet (3 meters); consult the drive manufacturer's specifications for additional limitations. Figure 2-8 illustrates a typical daisy-chain flat ribbon cable designed to interface two standard-sized drives.

Table 2-13 lists compatible controller mating connectors and cable. Refer to the drive manufacturer's documentation for the required drive mating connectors.

Table 2-13. Mating Connectors

Controller Connector	Mating Connector	Cable
J1	3M 3414-7034 or T&B/Ansley 609-3401M	3M 3365/34 T&B/Ansley 171-34 Spectra-Strip (twisted pair) 455-248-34
J2	3M 3425-7050 or T&B/Ansley 609-5001M	3M 3365/50 T&B/Ansley 171-50 Spectra-Strip (twisted pair) 455-248-50

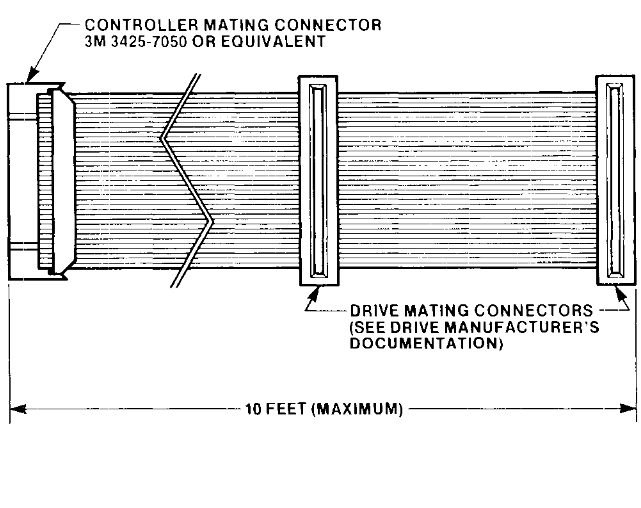


Figure 2-8. Flat-Ribbon I/O Interface Cable

143078-8

2-20. DRIVE MODIFICATIONS

The following subsections define the general drive modifications that may be necessary to ensure proper interface with the controller. Detailed information is included in the drive manufacturer's documentation.

2-21. READY LOGIC

Most standard-sized drives compatible with the controller provide a ready indication to the controller only when the drive is selected. If the drive provides an ungated READY/ output (generally referred to as radial ready), the drive must be modified to condition the drive's READY/ output with DRIVE SELECT/. Most mini-sized drives do not provide a ready indication. Accordingly, when interfacing drives that do not provide a READY/ output, make sure that the controller's READY/ input is permanently enabled with the installation of a jumper link between jumper posts E17 and E19 as described in table 2-6.

2-22. MOTOR-ON CONTROL

The MOTOR ON/ control output for mini-sized drives must be enabled prior to drive selection to allow time for the drive to reach operating speed before an operation is initiated. Accordingly, the MOTOR ON/ input to the drive must not be gated with DRIVE SELECT/. Note that this restriction applies when using either a common MOTOR ON/ signal or a radial MOTOR ON/ signal in multiple-drive configurations.

2-23. RADIAL HEAD LOAD

As an option, the controller can be configured to provide individual (radial) HEAD LOAD/ outputs. When this option is used, the individual HEAD LOAD/ signals must not be gated with their associated DRIVE SELECT/ signal within the drive.

2-24. DRIVE TERMINATION

When two or more drives are interfaced (daisy-chained) to the controller, the termination resistors/networks on the following common drive input signal lines must be removed from all but the last physical drive on the cable:

DIRECTION/
STEP/
WRITE DATA/
WRITE GATE/
HEAD LOAD/ (common only)
LOW CURRENT/ (if used)
FAULT RESET/ (if used)
MOTOR ON/ (if used; common only)

2-25. DRIVE NUMBERING

When interfacing multiple drives, each drive must be assigned a unique drive unit number. Depending on the manufacturer, internal drive unit assignment may be determined by wire jumper, shorting plug, or individual switch contacts. Generally, drives are shipped by their manufacturer configured for single-drive systems (i.e., the drive is assigned unit 0 with drives numbered 0 through 3 or unit 1 with drives numbered 1 through 4).

2-26. MULTIPLE DRIVE PIN ASSIGNMENTS

When interfacing more than one drive, unique pin assignments for the individual DRIVE SELECT/, MOTOR ON/ (when used in radial configuration), and radial HEAD LOAD/ (optional) signal lines associated with each drive must be provided. The actual pin assigned will depend on pin availability based on drive features supported and interface signal requirements. Note that it may be necessary to cut traces within the drive in order to reroute the input signal within the drive. It also may be necessary to cut traces to omit non-critical drive status signals (e.g., TWO-SIDED/ or IN USE) in order to provide additional pin assignments on the interface. Appendix C lists the pin assignments for a number of the standard- and mini-sized drives compatible with the controller.

2-27. STEPPER MOTOR POWER

Many drives compatible with the controller support a power-down feature that allows power to the stepper motor to be enabled only when the drive is selected. Since the controller automatically polls all four possible drives for a change in drive-ready status by cycling through the DRIVE SELECT/ lines, the power down feature cannot be supported directly (i.e., power to the stepper motor must not be dependent on drive selection). Note that in addition to the above restriction, the interval between drive selection and the generation of the first STEP/ pulse is too short to allow the stepper motor to be enabled by the DRIVE SELECT/ lines. When the stepper motor power-down featue is to be used, the host processor must enable and disable the stepper motor through the controller's auxiliary port.

CHAPTER 3
PROGRAMMING INFORMATION

3-1. INTRODUCTION

This chapter describes the I/O port commands that are executed by the host processor to convey information to and from the controller's programmable flexible disk controller (FDC) and DMA controller (DMAC) circuits and the individual FDC commands that control all disk operations and the transfer of data to and from the drive. Additionally, this chapter contains a description of the diskette formats supported and individual flow charts depicting the various diskette operations.

All disk operations are defined and initiated by the host processor through the execution of a series of I/O port commands while the controller is functioning as a bus slave. Once all information required to define the operation has been received, the controller functions as a bus master; the controller accesses and maintains control of the system bus and completes the specified operation without further intervention from the host processor. When the operation is complete, the controller reverts to a bus slave; the host processor must interrogate the controller to determine the outcome of the operation.

To initiate a disk operation, a series of I/O port commands is executed by the host processor. This series of commands defines the FDC operation to be performed, provides all supplemental information (parameters) required to perform the operation, and, if a data transfer to or from the diskette is indicated, defines the direction of the data transfer, the starting memory address of the first data byte to be transferred and the number of bytes to be transferred.

3-2. I/O PORT COMMANDS

Host processor communication with the controller is accomplished through an I/O port address block as defined by the least-significant bits of the I/O address. The location of this block (the I/O base address) in host processor memory must be on a 32-bit boundary (64-bit boundary with iSBX Multimodule board installed) and is defined by the user through a set of jumpers on the controller. These jumpers correspond to the three most-significant bits of an 8-bit I/O address or the eleven most-significant bits of a 16-bit I/O address (8- or 16-bit I/O addressing is user-selectable by an additional jumper on the controller).

The host processor executes an I/O port read or write instruction at one of the locations within the I/O port address block to transfer information either to (I/O write) or from (I/O read) the controller. Table 3-1 defines the controller's I/O port command set. Note that a number of the ports can be both read and written while other ports are either read-only or write-only. Each port command transfers one byte of data; a number of the I/O port commands require two data bytes (i.e., the port command must be issued twice to transfer all data associated with the I/O port command).

3-3. READ/WRITE DMAC ADDRESS REGISTERS

The controller's DMAC circuit has four DMA channels of which three channels are available. Each channel has an identical pair of 16-bit address registers, a "current-address" register, and a "base-address" register (each channel also has an identical pair of 16-bit word-count registers). Channel 0 is used by the controller for all diskette data transfers, Channel 1 is not used, and Channels 2 and 3 are available for use by an iSBX Multimodule board installed on the controller.

The Write DMAC Address Register command is used to simultaneously load a channel's current-address register and base-address register with the memory address of the first byte to be transferred. (The Channel 0 current/base address register must be loaded prior to initiating a diskette read or write operation.) Since each channel's address registers are 16 bits in length (64K address range), two "write address register" commands must be executed in order to load the complete current/base address registers for any channel. The register byte loaded (high- or low-order) is determined by the state of the DMAC's first/last flip-flop. (When the flip-flop is reset, the associated data byte is written into the low-order eight bits of the register; the flip-flop is toggled with each command so that a second address register command accesses the "other" byte.) The current-address register is incremented with each byte transferred; the base-address register maintains its initial value until it is reloaded by a subsequent Write Address Register command (or until the DMAC or controller is reset).

The Read DMAC Address Register command reads the low- or high-order byte of a channel's current-address register (a channel's base-address register

Table 3-1. I/O Port Controller Commands

Port Address	Mode	Command Function
0	Write Read	Load DMAC Channel 0 Base and Current Address Regsiters Read DMAC Channel 0 Current Address Register
1	Write Read	Load DMAC Channel 0 Base and Current Word Count Registers Read DMAC Channel 0 Current Word Count Register
2,3	—	Reserved
4	Write Read	Load DMAC Channel 2 Base and Current Address Registers Read DMAC Channel 2 Current Address Register
5	Write Read	Load DMAC Channel 2 Base and Current Word Count Registers Read DMAC Channel 2 Current Word Count Register
6	Write Read	Load DMAC Channel 3 Base and Current Address Registers Read DMAC Channel 3 Current Address Register
7	Write Read	Load DMAC Channel 3 Base and Current Word Count Registers Read DMAC Channel 3 Current Word Count Register
8	Write Read	Load DMAC Command Register Read DMAC Status Register
9	Write	Load DMAC Request Register
0A	Write	Set/Reset DMAC Mask Register
0B	Write	Load DMAC Mode Register
0C	Write	Clear DMAC First/Last Flip-Flop
0D	Write	DMAC Master Clear
0E	—	Reserved
0F	Write	Load DMAC Mask Register
10	Read	Read FDC Status Register
11	Write Read	Load FDC Data Register Read FDC Data Register
12	Write Read	Load Controller Auxiliary Port Poll Interrupt Status
13	Write	Controller Reset
14	Write	Load Controller Low-Byte Segment Address Register
15	Write	Load Controller High-Byte Segment Address Register
16-1F	—	Not Used
20-2F	—	Reserved for iSBX Multimodule Board (see Appendix B)

cannot be read). The current-address register byte accessed is determined by the state of the DMAC's first/last flip-flop as previously described.

3-4. READ/WRITE DMAC WORD COUNT REGISTERS

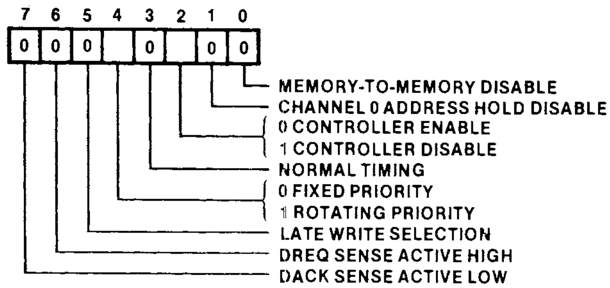
Like the DMAC address registers, each DMA channel also has an identical pair of 16-bit word-count registers, a "current word-count register" and a "base word-count register." The channel 0 word-count registers are used to specify the number of bytes to be transferred during a diskette read or write operation. The channel 1 word-count registers are not used, and the word-count registers for channels 2 and 3 are dedicated to DMA functions associated with a Multimodule board.

The Write DMAC Word Count Register command is used to simultaneously load a channel's current and base word-count registers with the number of bytes to be transferred during a subsequent DMA operation. Since the word-count registers are 16-bits in length, two commands must be executed to load both halves of the registers. As described in section 3-3, the register half loaded (low- or high-order) is determined by the state of the DMAC's first/last flip-flop. The actual count loaded is a binary value that is one less than the number of bytes to be transferred (i.e., the register value 01FFH transfers 512 bytes). During the subsequent DMA transfer, the current word-count register is decremented with each byte transferred; the base word-count register maintains its initially-loaded value until it is reloaded by a subsequent Write Word Count Register command or until either the DMAC or controller is reset. When the word count decrements to zero, the DMA transfer is stopped and the corresponding TC (terminal count) bit in the DMAC status register is set.

The Read DMAC Word Count Register command reads the low- or high-order byte of a channel's current word-count register (a channel's base word-count register cannot be read). The current word-count register byte accessed is determined by the state of the DMAC's first/last flip-flop.

3-5. WRITE DMAC COMMAND REGISTER

The Write DMAC Command Register command loads an 8-bit byte into the DMAC's command register to define the operating characteristics of the DMAC. The functions of the individual bits in the command register are defined in the following diagram. Note that only two bits within the register are applicable to the controller; the remaining bits select functions that are not supported and, accordingly, must always be set to zero.



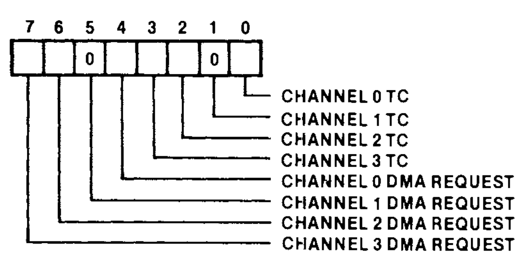
Bit 2: Controller Enable/Disable. This bit, when set to one, prevents all DMA channels from responding to data transfer requests. Normally, this bit is always set to zero to enable the DMAC. When multiple DMA channels are used and a non-essential DMA request from the iSBX Multimodule board could interrupt the programming of channel 0, the DMAC could be disabled while it is being programmed and then enabled by a subsequent Write DMAC Command Register command after it has been programmed.

Bit 4: Fixed/Rotating Priority. This bit, when set to zero, selects fixed priority (channel 0 has the highest priority, channel 3 has the lowest priority) and when set to one, selects rotating priority (each channel is granted highest priority on a rotational scheme).

Note that when programming the command register, an all-zero byte enables the DMAC and gives the disk controller (channel 0) the highest priority. The command register is cleared by a DMAC master clear or controller reset.

3-6. READ DMAC STATUS REGISTER COMMAND

The Read DMAC Status Register command accesses an 8-bit status byte that identifies the DMA channels that have reached terminal count or that have a pending DMA request.



Bits 0 through 3 are set when their corresponding channel has reached terminal count (i.e., when the channel's current word count register decrements to zero). Since DMA channel 1 is not used, bit 1 always is zero. Bits 2 and 3 are associated with an iSBX Multimodule board and indicate a terminal count condition on channels 2 and 3. Note that if external EOP (End of Process) logic is implemented on the iSBX Multimodule board, the generation of an external EOP signal sets the active channel's TC bit irrespective of the current word count.

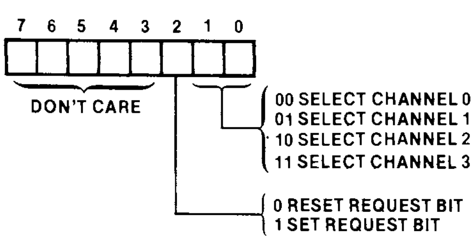
Bits 4 through 7 are set when their corresponding channel requests DMA service (DMAC's DREQ input activated or corresponding bit in the request register set). Again, since DMA channel 1 is not used, bit 5 always is zero, and bits 6 and 7 indicate DMA requests originating from an iSBX Multimodule board.

The TC bits in the status register are cleared whenever the register is read by a DMAC master clear or by a controller reset.

3-7. WRITE DMAC REQUEST REGISTER

The Write DMAC Request Register command is used with DMAC channels 2 and 3 (the iSBX Multimodule board channels) to allow DMA requests to be initiated by the host processor. The command only can be used when the selected channel is operated in the "block transfer mode" (see section 3-9); the controller's DMA channel (channel 0) operates in either the "single transfer mode" or the "demand transfer mode" and does not use the Write DMAC Request Register command.

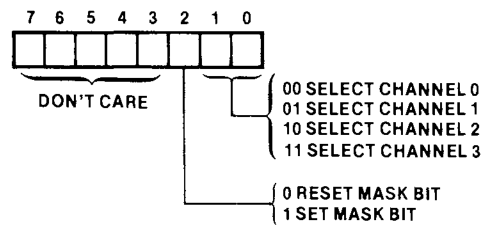
The data byte associated with the Write DMAC Request Register command sets or resets a channel's associated request bit within the DMAC's internal 4-bit request register.



The individual channel request bits are non-maskable and are subject to channel prioritization (fixed or rotating). Each request bit is individually set or reset according to the state of bit 2 and, when once set within the register, is cleared when the corresponding channel reaches terminal count or when an external EOP signal is applied. The entire request register is cleared by a DMAC master clear or controller reset.

3-8. SET/RESET DMAC MASK REGISTER

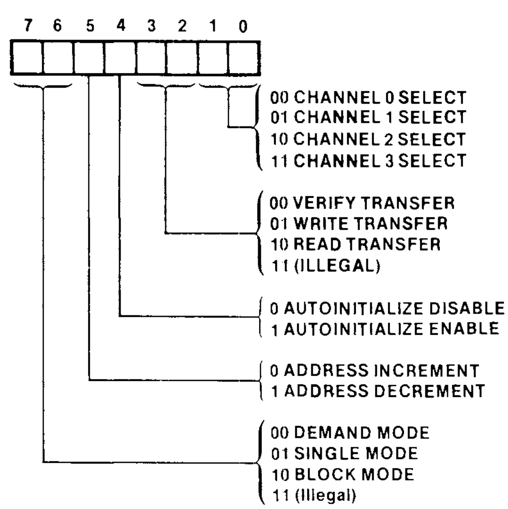
The Set/Reset DMAC Mask Register command is used to reset (or set) individual bits within the DMAC's internal 4-bit mask register. Each DMAC channel has an associated mask bit within the register that, when reset, enables the channel's DREQ (DMA Request) input and, when set, disables (masks) the DREQ input.



Prior to a DREQ-initiated DMA transfer, the channel's mask bit must be reset to enable recognition of the DREQ input. When the transfer is complete (terminal count reached or external EOP applied) and the channel is not programmed to autoinitialize, the channel's mask bit is automatically set (disabling DREQ) and must be reset prior to a subsequent DMA transfer. All four bits of the mask register are set (disabling the DREQ inputs) by a DMAC master clear or controller reset. Additionally, all four bits can be set/reset by a single Write DMAC Mask Register command (see section 3-12).

3-9. WRITE DMAC MODE REGISTER

The Write DMAC Mode Register command is used to define the operating mode characteristics for each DMA channel. Each channel has an internal 6-bit mode register; the high-order six bits of the associated data byte are written into the mode register addressed by the two low-order bits.



Verify Transfer. The verify transfer mode is not used by the controller; this mode may be used by an iSBX Multimodule board.

Write Transfer. The write transfer mode programs the selected DMA channel to transfer data from the I/O device to host memory. This mode must be selected to read data from the diskette (i.e., the data read from the diskette is written into host memory).

Read Transfer. The read transfer mode programs the selected DMA channel to transfer data from host memory to the I/O device. This mode must be selected to write data on the diskette (i.e., the data read from host memory is written onto the diskette).

Autoinitialize. The autoinitialize enable/disable bit is used to control a channel's autoinitialization function. When this bit is set (1), the autoinitialize mode is enabled, and the current word-count and current address register values are automatically restored from the corresponding base registers when the DMA transfer is complete (terminal count or EOP). The mask bit is not set when the autoinitialize mode is enabled, and the channel is prepared to perform a subsequent DMA transfer without reprogramming the DMAC. Note that for most controller applications, the autoinitalize mode is not used.

Address Increment/Decrement. The address increment/decrement bit determines the sequence in which memory addresses are generated. When this bit is reset (0), the memory address in the current address register is incremented with each byte transferred. Conversely, when the address increment/decrement bit is set (1), the memory address in the current address register is decremented with each byte transferred. Note that for most controller applications, the address increment mode is used.

Demand Mode. In the demand transfer mode the channel data transfer is initiated by DREQ. The channel continues to transfer data until DREQ goes inactive or until either a terminal count condition is reached or an external EOP is received. If DREQ is held active throughout the entire transfer, the channel maintains bus access until the transfer is complete. The controller (DMAC channel 0) can use the demand transfer mode; however, since the DREQ input from the FDC goes inactive following each byte transferred, the channel releases the bus after each byte transferred.

Single Transfer Mode. In single transfer mode, the channel performs a sequence of single byte transfers (the channel releases the bus after each byte transferred) until the transfer is complete (terminal count reached or external EOP applied). DREQ must be held active by the I/O device until DACK is received. Unlike the demand mode, if DREQ is held active throughout the entire transfer, the bus is released with each byte transferred. The controller normally uses the single transfer mode for all DMA data transfers.

NOTE

Since both the demand and single transfer modes used by the controller release the bus with each byte transferred, the controller should be given "high bus priority" to prevent interruptions in the DMA transfer.

Block Mode. In block transfer mode the channel data transfer again is initiated by DREQ. The transfers continue, irrespective of the state of DREQ, until terminal count is reached or an external EOP is applied. (DREQ must be held active until DACK is received.) The controller does not support operation in the block transfer mode; this mode may be used by an iSBX Multimodule board.

3-10. CLEAR DMAC FIRST/LAST FLIP-FLOP

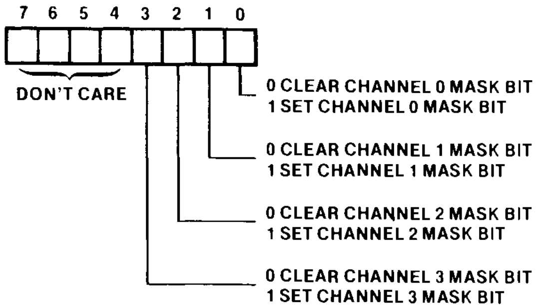
The Clear DMAC First/Last Flip-Flop command initializes the DMAC's internal first/last flip-flop so that the next byte written to or read from the 16-bit address or word-count registers is the low-order byte. The flip-flop is toggled with each register access so that a second register read or write command accesses the high-order byte. The first/last flip-flop also is initialized (to access the low-order register byte) by a DMAC master clear or controller reset. Note that the Clear DMAC First/Last Flip-Flop command does not require a specific bit pattern in the associated command data byte.

3-11. DMAC MASTER CLEAR

The DMAC Master Clear command clears the DMAC's command, status, request, and temporary registers to zero, initializes the first/last flip-flop, and sets the four channel mask bits in the mask register to disable all DMA requests (i.e., the DMAC is placed in an idle state). Note that the DMAC Master Clear command does not require a specific bit pattern in the associated command data byte.

3-12. WRITE DMAC MASK REGISTER

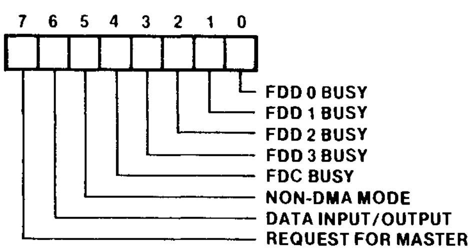
The Write DMAC Mask Register command allows all four bits of the DMAC's mask register to be written with a single command.



Like the Set/Reset DMAC Mask Register command, clearing a channel's mask bit enables recognition of the associated DREQ input, and setting a channel's mask bit disables (masks) the associated DREQ input. Again, a DMAC master clear or controller reset sets all four mask register bits (disabling the DREQ inputs).

3-13. READ FDC STATUS REGISTER

The Read FDC Status Register command accesses the FDC's main status register. The individual status register bits are as follows:



FDD 0 BUSY. This bit, when set (1), indicates that drive 0 is in the process of performing a seek operation.

FDD 1 BUSY. This bit, when set, indicates that drive 1 is in the process of performing a seek operation.

FDD 2 BUSY. This bit, when set, indicates that drive 2 is in the process of performing a seek operation.

FDD 3 BUSY. This bit, when set, indicates that drive 3 is in the process of performing a seek operation.

FDC BUSY. This bit, when set, indicates that the FDC is in the command execution phase (i.e., the FDC is in the process of performing a diskette read or write operation).

NON-DMA MODE. This bit only is applicable in systems that do not support DMA tranfers and is irrelevant to the controller.

DATA INPUT/OUTPUT. The data input/output (DIO) bit indicates the direction of the transfer between the FDC's data register and the host processor. When this bit is set, the direction of the transfer is from the FDC to the host processor, and when this bit is reset, the direction of the transfer is from the host processor to the FDC.

REQUEST FOR MASTER. The request for master (RQM) bit, when set, indicates that the FDC's data register is ready to present a byte to or accept a byte from the host processor.

The host processor can read the main status register at any time and should use the DIO and RQM status bits to perform a "handshaking" function with the FDC when transferring data to or from the FDC's data register. Figure 3-1 shows the status register timing.

Note that like any microprocessor, the FDC requires a finite amount of time to update its RQM status bit between byte transfers to or from the data register. The sample PL/M and assembly language drivers in Appendix A illustrate typical wait subroutines that must be inserted between successive byte transfers to or from the FDC's data register.

3-14. READ/WRITE FDC DATA REGISTER

The Read and Write FDC Data Register commands are used to write command and parameter bytes to the FDC in order to specify the operation to be performed (referred to as the "command phase") and to read status bytes from the FDC following the operation (referred to as the "result phase"). During the command and result phases, the 8-bit data register is actually a series of 8-bit registers in a stack. Each register is accessed in sequence; the number of registers accessed and the individual register contents are defined by the specific disk command (refer to the FDC command descriptions in sections 3-20 through 3-32).

3-15. WRITE CONTROLLER AUXILIARY PORT

The Write Controller Auxiliary Port command is used to set or clear individual bits within the controller's auxiliary port. The four low-order port bits are dedicated to auxiliary drive control functions

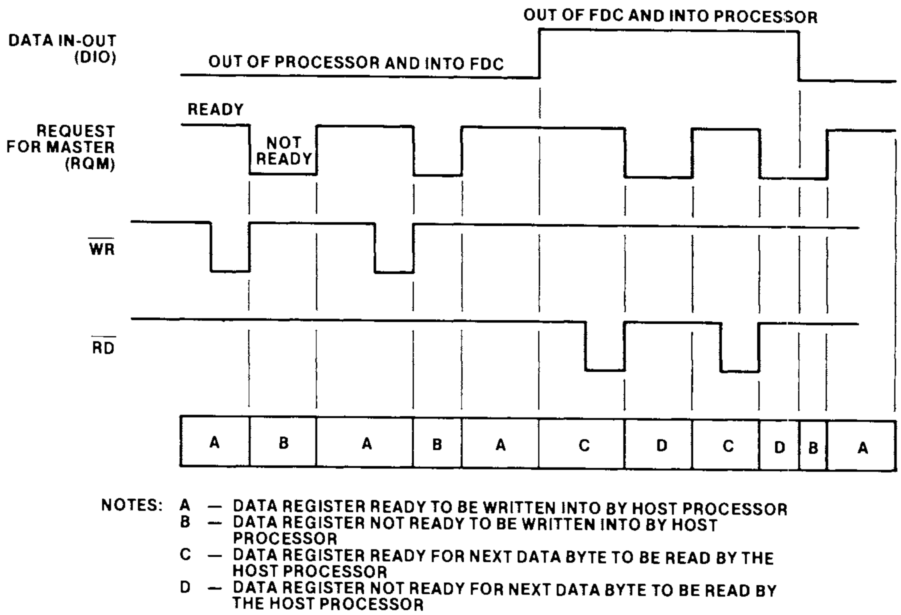
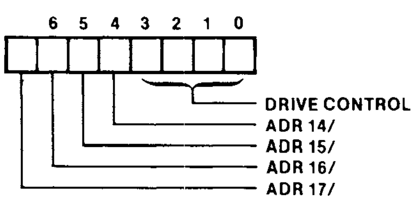


Figure 3-1. Main Status Register Timing 143078-9

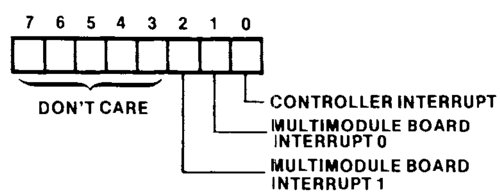
(jumper links are required to connect the desired port bit to an available pin on the drive interface connectors; see section 2-14). The most common application for these bits is the "Motor-On" control function for mini-sized drives.

The four high-order bits of the auxiliary port are the ADR14 through ADR17 address bits that are used to extend the DMA addressing capability of the controller to 16 megabytes (24-bit addressing). These bits are set prior to initiating a diskette read or write operation to define the specific 1-megabyte page of memory to be accessed.



3-16. POLL INTERRUPT STATUS

The Poll Interrupt Status command presents the interrupt status of the controller and the two interrupt status lines dedicated to the iSBX Multimodule board. This command is used by the host processor when interrupts are disabled to poll the controller (and iSBX Multimodule board) in order to determine when an operation has been completed. A bit set in the status byte returned indicates a pending interrupt.



3-17. CONTROLLER RESET

The Controller Reset command is the software reset for the controller. This command clears the controller's auxiliary port and segment address register, provides a reset signal to the iSBX Multimodule board and initializes the bus controller (releases the bus), the DMAC (clears the internal registers and masks the DREQ inputs), and the FDC (places the FDC in an idle state and disables the output control lines to the diskette drive). Following reset, the controller is in an idle state. Note that the Controller Reset command does not require a specific bit pattern in the associated command data byte.

3-18. WRITE CONTROLLER LOW- AND HIGH-BYTE SEGMENT ADDRESS REGISTERS

The Write Controller Low- and High-Byte Address Registers commands are required when the controller uses 20-bit addressing (memory address range from 0 to 0FFFFFH). These commands are issued prior to initiating a diskette read or write operation to specify the 16-bit segment address. The data byte loaded into the low-order half of the register is the A4 through A11 address bits, and the data byte loaded into the high-order half of the register is the A12 through A19 address bits.

7	6	5	4	3	2	1	0
A11	A10	A9	A8	A7	A6	A5	A4

LOW-ORDER SEGMENT ADDRESS

7	6	5	4	3	2	1	0
A19	A18	A17	A16	A15	A14	A13	A12

HIGH-ORDER SEGMENT ADDRESS

During the subsequent DMA transfer, the segment address is combined with the DMAC's current address to form a 20-bit "effective" address. As shown in figure 3-2, the segment address value is offset by four bit positions and added to the current address value.

The segment address register is reset (to zero) by the Reset Controller command.

3-19. DISKETTE ORGANIZATION

The controller is compatible with two physical sizes of diskettes: a single- or double-sided, standard 8-inch diskette that typically consists of 77 tracks and a single- or double-sided 5¼-inch mini diskette that typically consists of 35 tracks. Note that the term "cylinder" is used with double-sided drives to indicate the set of two tracks at a given head position.

The tracks are numbered sequentially (beginning at the outermost track) from 0 to 76 (standard size) or from 0 to 34 (mini size). Each track, in turn, is divided into sections or "sectors." The number of sectors on each track and the number of bytes per sector are program-determined ("soft sectoring") and are established when the track is formatted. The controller is programmed to operate in either the single-density (FM) format or the double-density (MFM) format. Figure 3-3 and table 3-2 describe the track and sector formats for both single- and double-density recording, and table 3-3 defines the recording capacities for both the standard 8-inch and 5¼-inch mini drives.

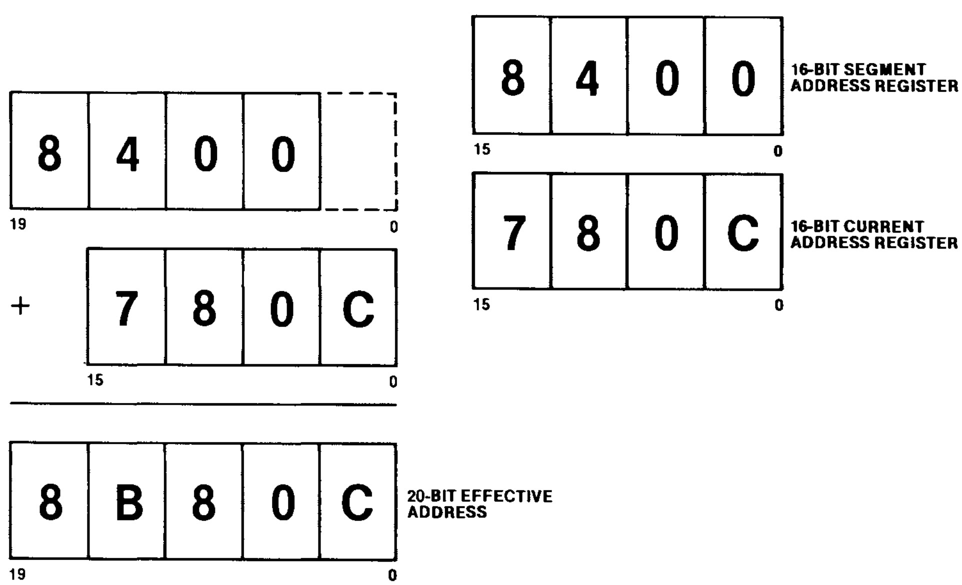


Figure 3-2. 20-Bit Addressing

143078-10

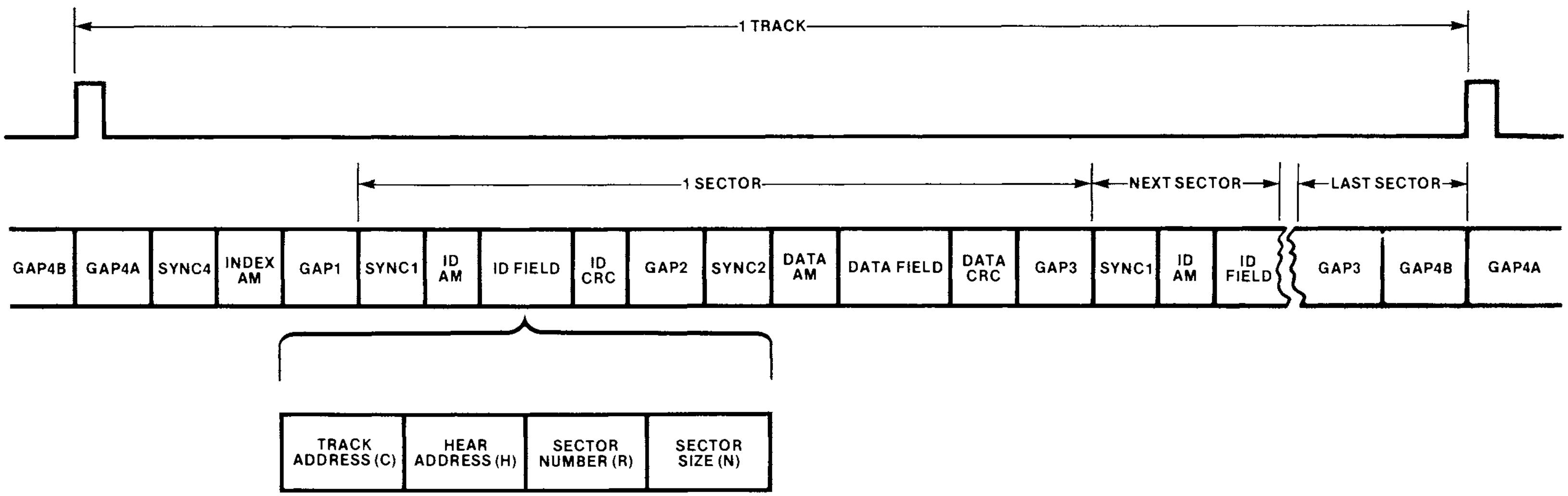


Figure 3-3. Track Format

143078-11

Table 3-2. Track Format

Designation	Description	FM Format		MFM Format	
		Number of Bytes	Pattern (Hexadecimal)	Number of Bytes	Pattern (Hexadecimal)
GAP 4A	Preamble gap; written by the FDC when the track is formatted.	40	FF	80	4E
SYNC 4	A sequence of all-zero bytes used to synchronize the controller's data separation logic prior to reading the index address mark; written by the FDC when the track is formatted.	6	00	12	00
INDEX AM	Index address mark. Unique data pattern that identifies the logical beginning of a track; written by the FDC when the track is formatted.	1	Data=FC Clock=D7	4	3 Bytes C2, 1 Byte FC
GAP 1	Post index gap; written by the FDC when the track is formatted.	26	FF	50	4E
SYNC 1	A sequence of all-zero bytes used to synchronize the controller's data separation logic prior to reading an ID field address mark; written by the FDC when the track is formatted.	6	00	12	00
ID AM	ID field address mark. Unique data pattern that identifies the beginning of a sector ID field; written by the FDC when the track is formatted.	1	Data=FE Clock=C7	4	3 Bytes A1, 1 Byte FE
ID FIELD	Four bytes used to uniquely identify each sector on a diskette by track address, side, sector number and sector size; these bytes are supplied to the FDC by the program (format table) and written in the ID field when the track is formatted.	4		4	
ID CRC	A 16-bit cyclic redundancy check character derived by the FCD from the ID address mark and the four ID field bytes and written immediately following the ID field when the track is formatted.	2		2	
GAP 2	Post ID field gap; written by the FDC when the track is formatted. During sector write operations, the controller switches the drive electronics from read to write during the post ID field gap interval.	11	FF	22	4E

Table 3-2. Track Format (Cont'd.)

Designation	Description	FM Format		MFM Format	
		Number of Bytes	Pattern (Hexadecimal)	Number of Bytes	Pattern (Hexadecimal)
SYNC 2	A sequence of all-zero bytes used to synchronize the controller's data separation logic prior to reading a data field address mark. These sync bytes are rewritten by the FDC during every sector write operation.	6	00	12	00
DATA AM	Data field address mark. Unique data pattern that identifies the beginning of a sector's data field; the data field address mark is written by the FDC each time the sector is written. Note that data pattern F8 (deleted data mark) is used in place of FB to identify a deleted sector.	1	FB	4	3 Bytes A1, 1 Byte FB
DATA FIELD	The sector's data field. The length of the data field (number of bytes) is programmable: 128 (single-density only), 256, 512, 1024, 2048, 4096, and 8192 (double-density only).				
DATA CRC	A 16-bit cyclic redundancy check character derived by the FDC from the data field address mark and the data field bytes and written immediately following the data field during sector write operations. During subsequent sector read operations, a second CRC character is calculated from the data read and compared with the CRC character previously written to verify data integrity.	2		2	
GAP 3	Post data field gap. A program-selectable gap length that separates the previous sector's data field from the next sector's ID field. The gap length specified is dependent on the recording format and sector length (see table 3-10). Note that the gap length specified differs for a format command and read/write commands.		FF		4E
GAP 4B	Postamble gap. A variable-length gap that follows the last sector on a track. This gap is written by the FDC when the track is formatted and extends from the end of gap 3 to the index pulse.		FF		4E

Table 3-3. Recording Capacities

Drive Size	Sectors Per Track	Bytes Per Sector		Bytes Per Track (Formatted)	
		Single Density (FM)	Double Density (MFM)	Single Density (FM)	Double Density (MFM)
Standard 8-inch	26	128	256	3328	6656
	15	256	512	3840	7680
	8	512	1024	4096	8192
	4	1024	2048	4096	8192
	2	2048	4096	4096	8192
	1	4096	8192	4096	8192
5¼-inch mini	18	128	256	2304	4608
	16	128	256	2048	4096
	8	256	512	2048	4096
	4	512	1024	2048	4096
	2	1024	2048	2048	4096
	1	2048	4096	2048	4096

3-20. FDC COMMANDS

The FDC is capable of executing 12 unique commands. Of these 12 commands, all but one (the Sense Interrupt Status command) require a multibyte transfer from the host processor to initiate command execution. Following the execution of most commands, the host processor must initiate a multibyte transfer from the FDC to determine the outcome of the operation and to terminate the command. Table 3-4 lists the FDC commands and the number of bytes required to initiate and to terminate the command.

Table 3-4. FDC Commands

Command	Command Bytes	Result Bytes
Specify	3	0
Seek	3	0
Read Data	9	7
Read Deleted Data	9	7
Read ID	2	7
Read Track	9	7
Write Data	9	7
Write Deleted Data	9	7
Format Track	6	7
Recalibrate	2	0
Sense Drive Status	2	1
Sense Interrupt Status	1	2

FDC command processing consists of three phases that are entered in the following sequence:

1. Command Phase. The host processor initiates command processing by writing one or more bytes to the FDC's data register. Depending on the command to be executed, up to nine bytes may be required before the execution phase can be entered.
2. Execution Phase. The operation specified during the command phase is performed. The execution phase is entered automatically when the last command byte is received. Since the controller uses DMA for all data transfers to and from the diskette, no host processor intervention is required during the execution phase.
3. Result Phase. Following command execution, the FDC enters the result phase. With most commands, the FDC generates an interrupt to inform the host processor of the completion of the execution phase. To complete the result phase the host processor must read a series of bytes from the FDC's data register.

During the command and result phases, the main status register must be read by the host processor to determine when the FDC is ready to provide or accept the next command or result byte to be written or read from the data register as described in section 3-13. Note that during multibyte transfers to or from the FDC's data register, a delay interval must be inserted between each byte read or written to allow time for the FDC to update the main status register (see "wait" routines in the sample drivers in Appendix A).

During the execution phase of commands that transfer data to or from the diskette, the FDC generates a DMA request for each byte transferred. The DMAC responds to the DMA request with a DMA acknowledge to reset the DMA request. When the transfer is complete (terminal count received from the DMAC), the FDC generates an interrupt to indicate the beginning of the result phase. When the host processor reads the first byte from the FDC's data register (status byte ST0), the FDC automatically clears the interrupt. The host processor must read all of the result bytes to complete the command; the FDC will not accept a new command until the current command is completed.

The FDC contains five status registers. The main status register, as previously mentioned, is read by the host processor during the command and result phases. The other four status registers (ST0, ST1, ST2 and ST3) are read directly from the FDC's data register during the result phase. The status registers presented are determined by the FDC command executed. Table 3-5 defines the contents of the four result status registers.

The writing and reading of the command and result bytes to and from the FDC's data register must be performed in the order shown by the command format tables given in the individual FDC command descriptions. After the last command byte is written to the FDC, the execution phase starts automatically. During the execution phase of commands that write data to or read data from the diskette, the number of bytes transferred is determined by the word-count value loaded into the DMAC. The DMAC signals the FDC when the programmed number of bytes have been transferred; the FDC then stops the transfer, interrupts the host processor, and enters the result phase. When the host processor reads the last result byte from the FDC's data register, the command is completed and the FDC is prepared to accept a new command.

Table 3-6 defines the mnemonics used in the command format tables for the individual commands.

Table 3-5. Result Phase Status Registers

Status Register 0 (ST0)			
Bit(s)	**Name**	**Symbol**	**Description**
D7,D6	Interrupt Code	IC	D7=0 and D6=0 Normal Termination of Command. Command execution was completed successfully. D7=0 and D6=1 Abnormal Termination of Command. Command execution was initiated, but was not completed successfully. D7=1 and D6=0 Invalid Command Issued. Command execution was not initiated. D7=1 and D6=1 Ready Change. During command execution the drive went Not Ready.
D5	Seek End	SE	Set (D5=1) when the FDC completes execution of a Seek command. Note that there is no result byte associated with a Seek command; the host processor must issue a Sense Interrupt Status command to access the ST0 status byte. When parallel (overlapped) seeks are performed on multiple drives, this bit is set by the first drive to complete its seek (the drive completing its seek is identified by unit select bits D0 and D1). When performing overlapped seeks, the main status register must be examined to determine when the other drives have completed their seeks.
D4	Equipment Check	EC	Set when the addressed drive activates its FAULT/ signal to the controller or when during execution of a Recalibrate command, a TRACK 0/ signal is not received from the addressed drive after 77 STEP/ pulses have been issued. Note that if a FAULT/ signal is received during command execution, the operation is terminated immediately.
D3	Not Ready	NR	Set when a command that accesses the drive is issued and the drive is in a not-ready state or when a diskette read or write command, which specifies side 1 of a single-sided drive, is issued. The command issued is not executed.
D2	Head Address	HD	Indicates the state of the head (side selected) when the interrupt was generated (0 = side 0, 1 = side 1).
D1,D0	Unit Select	US	Indicates the drive unit number of the drive addressed when the interrupt was generated. D1=0, D0=0 = DRIVE UNIT 0 D1=0, D0=1 = DRIVE UNIT 1 D1=1, D0=0 = DRIVE UNIT 2 D1=1, D0=1 = DRIVE UNIT 3

Table 3-5. Result Phase Status Registers (Cont'd.)

Status Register 1 (ST1)			
Bit(s)	**Name**	**Symbol**	**Description**
D7	End of Cylinder	EN	Set (D7=1) during a multisector transfer when the starting sector number and the number of bytes to be transferred exceeds the last logical sector on the track or cylinder (multitrack transfers). The data transfer is terminated by the FDC when the data from the last logical sector on the track or cylinder is transferred.
D6	Not Used		This bit is always 0 (reset).
D5	Data Error	DE	Set when the FDC detects a CRC error in either the ID field or data field.
D4	Overrun	OR	Set when the DMAC did not respond to a data request within the allotted time interval to prevent loss of data. When an overrun condition occurs, the operation is terminated immediately.
D3	Not Used		This bit is always 0 (reset).
D2	No Data	ND	Set during execution of diskette read/write commands when the currently-addressed sector cannot be located within one full revolution of the diskette (i.e., two index pulses encountered). This bit usually indicates an improperly formatted diskette or, when the WC bit in status register 2 also is set, indicates that the head is positioned over the wrong track. During execution of a Read Track command, the ND bit is set when the FDC cannot locate the ID field of the first sector following the index mark.
D1	Not Writable	NW	Set during execution of a Write Data, Write Deleted Data or Format Track command if the WRITE PROTECT signal from the drive is active. The Write operation is immediately aborted, and no data is written on the diskette.
D0	Missing Address Mark	MA	Set during execution of diskette read/write operations if an ID address mark cannot be found within one revolution of the diskette (usually indicates that an unformatted diskette has been installed in the drive). This bit also is set during diskette read operations if the addressed sector's data address mark (or deleted data address mark) is not encountered (the MD bit in status register 2 also will be set).
Status Register 2 (ST2)			
D7	Not Used		This bit is always 0 (reset).
D6	Control Mark	CM	Set during execution of a Read Data command when a deleted data address mark is encountered or set during execution of a Read Deleted Data command when a (normal) data address mark is encountered.
D5	Data Error in Data Field	DD	Set during diskette read operations when a CRC error is detected in the data field. Note that since this bit is only set after the sector is read, the data transferred must be considered invalid.
D4	Wrong Cylinder	WC	Set when the cylinder (track) address specified does not match the cylinder address byte read from a sector's ID field. Note that the ND bit in status register 1 also will be set.
D3,D2	Reserved		These bits are 0 (reset).
D1	Bad Cylinder	BC	Set when a diskette read/write operation is attempted on a defective "bad" track (bad tracks are designated by writing a byte of FFH in the cylinder address byte of each ID field on the defective track using the format track command). Note that the WC bit and the ND bit in status register 1 also will be set.
D0	Missing Address Mark in Data Field	MD	Set during diskette read operations when the addressed sector's data address mark (or deleted data address mark) is not encountered before the next sector's ID address mark is read. Note that the MA bit in status register 1 also will be set.

Table 3-5. Result Phase Status Registers (Cont'd.)

Status Register 3 (ST3)			
Bit(s)	**Name**	**Symbol**	**Description**
D7	Fault	FT	Set when the FAULT/ signal from the addressed drive is active.
D6	Write Protected	WP	Set when the WRITE PROTECT/ signal from the addressed drive is active.
D5	Ready	RDY	Set when the READY/ signal from the addressed drive is active.
D4	Track 0	T0	Set when the TRACK0/ signal from the addressed drive is active.
D3	Two Sided	TS	Set when the TWO SIDED/ signal from the addressed drive is active.
D2	Head Address	HD	Indicates the state of the SIDE SELECT/ signal to the drive (0 = side 0, 1 = side 1).
D1,D0	Unit Select	US	Indicates the drive unit number of the drive addressed by the Sense Drive Status command: D1=0, D0=0 = DRIVE UNIT 0 D1=0, D0=1 = DRIVE UNIT 1 D1=1, D0=0 = DRIVE UNIT 2 D1=1, D0=1 = DRIVE UNIT 3

Table 3-6. Command Mnemonics

Symbol	Name	Description
C	Cylinder Number	The cylinder (track) number. In the result phase, C defines the current location (track address) of the drive's read/write head(s).
D	Data	The data pattern (filler byte) to be written into each sector's data field when the track is formatted.
D7-D0	Data Bus	The 8-bit data bus to/from the FDC's data register. Bit D0 is the least-significant bit.
DS0,DS1	Drive Select	The drive unit addressed by the command: DS1=0, DS0=0 = DRIVE UNIT 0 DS1=0, DS0=1 = DRIVE UNIT 1 DS1=1, DS0=0 = DRIVE UNIT 2 DS1=1, DS0=1 = DRIVE UNIT 3
DTL	Data Length	Specifies non-standard data transfer length for diagnostic use only. For normal diskette transfer operations, the DTL value specified must be FFH.
EOT	End of Track	The sector number of the last logical sector on a track.
GPL	Gap Length	The number of bytes to be written into Gap 3 (see table 3-10).
H	Head Address	The head (side) selected (0 = side 0, 1 = side 1). In the command phase, H and HDS are the same value; in the result phase, H reflects the state of the side select signal on interrupt.
HDS	Head Select	The read/write head addressed by the command (0 = head 0, 1 = head 1).
HLT	Head Load Time	The head load time interval in the Specify command. The HLT value specified corresponds to the drive's head load time specification (see table 3-9).
HUT	Head Unload Time	The head unload time interval in the Specify command. The HUT value specifies the time interval that the head remains loaded following a read or write operation (see table 3-7).
MFM	Mode Select	The recording mode selected (0 = FM mode, 1 = MFM mode).
MT	Multitrack	When set (MT=1), permits multisector read/write operations on the two tracks at the same cylinder address (i.e., an operation that begins on a track on side 0 can be continued on the same track on side 1).

Table 3-6. Command Mnemonics (Cont'd.)

Symbol	Name	Description
N	Number	A number (N) that represents the length of a sector; the sector length (number of data bytes) is equal to 128 x 2^N.
NCN	New Cylinder Number	The cylinder address value specified during the command phase of a Seek command.
PCN	Present Cylinder Number	The current position (cylinder address) of the read/write head(s).
R	Record	The record (sector) number. During the command phase, R specifies the (first) sector to be accessed; during the result phase, R indicates the number of the next logical sector number to be accessed.
R/W	Read/Write	The direction of the I/O transfer between the host processor and the FDC's data register.
SC	Sector Count	The number of sectors to be formatted on a track.
SK	Skip	When set (SK=1) during a Read Data command, causes the FDC to "skip over" any sectors with a deleted data address mark (when SK=0, the deleted sector is transferred and the command is terminated). When set during a Read Deleted Data command, causes the FDC to "skip over" any sectors with a (normal) data address mark (when SK=0, the "non-deleted" sector is transferred and the command is terminated).
SRT	Step Rate Time	The step rate time interval in the Specify command. The SRT value specified corresponds to the drive's step rate specification (see table 3-8).
ST0 ST1 ST2 ST3	Status Register 0 Status Register 1 Status Register 2 Status Register 3	One of the four status registers that are read by the host processor during the result phase to determine the outcome of command execution. The status registers can only be read after a command has been executed (i.e., after an interrupt) and contain information relevant only to the command executed.

3-21. SPECIFY COMMAND

The specify command requires three bytes to load the command and command data. Since this command only loads information into the FDC for future commands, there is no execution or result phase.

PHASE	R/W	DATA BUS								Remarks
		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
Command	W	0	0	0	0	0	0	1	1	Command Codes
	W	— SRT →				← HUT —				
	W	— HLT →							0	

The specify command sets the initial values for the three internal timers that define the drive's head load time and step rate characteristics to the FDC and the FDC's head unload time interval. Accordingly, a Specify command must be executed prior to executing any command that accesses the drive. The Head Unload Time (HUT) value 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 3-7. The Step Rate Time (SRT) value defines the time interval between step pulses sent to the drive from the FDC. This timer is programmable from 1 to 16 ms in 1 ms increments as shown in table 3-8. The SRT value must be set to 1 ms greater than the minimum desired step rate interval. The Head Load Time (HLT) value defines the time between activation of the HEAD LOAD signal and the 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 3-9.

Table 3-7. HUT Values

D_3	D_2	D_1	D_0	HEX	HUT TIME
0	0	0	1	1	16 ms
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

The time intervals mentioned in the previous paragraph are a direct function of the clock frequency. The times indicated are for an 8 MHz clock; if the clock frequency is reduced to 4 MHz (minifloppy applications), all time intervals are increased by a factor of two.

Table 3-8. SRT Values

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 must be set to 1 ms greater than the minimum desired step interval time.

Table 3-9. HLT Values

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	0	02	2 ms
0	0	0	0	0	1	0	0	04	4 ms
0	0	0	0	0	1	1	0	06	6 ms
0	0	0	0	1	0	0	0	08	8 ms
1	1	1	0	1	0	0	0	E8	232 ms
1	1	1	0	1	0	1	0	EA	234 ms
1	1	1	0	1	1	0	0	EC	236 ms
1	1	1	0	1	1	1	0	EE	238 ms
1	1	1	1	0	0	0	0	F0	240 ms
1	1	1	1	0	0	1	0	F2	242 ms
1	1	1	1	0	1	0	0	F4	244 ms
1	1	1	1	0	1	1	0	F6	246 ms
1	1	1	1	1	0	0	0	F8	248 ms
1	1	1	1	1	0	1	0	FA	250 ms
1	1	1	1	1	1	0	0	FC	252 ms
1	1	1	1	1	1	1	0	FE	254 ms

3-22. SEEK COMMAND

The Seek command requires three bytes to load the command and command data.

A Seek command must be executed prior to the execution of any command that accesses data on a track other than the track currently positioned under the read/write head(s).

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
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

During execution of a Seek command, the read/write head(s) in the addressed drive (DS1, DS0) are moved (stepped) from cylinder to cylinder by the FDC. The FDC compares the Present Cylinder Number (PCN), which is the current head position, with the New Cylinder Number (NCN) specified in the command and performs the following operation when there is a difference:

- When NCN is less than PCN, the FDC sets the DIRECTION signal to a "1" (step out toward track 0) and issues STEP pulses to the drive.
- When NCN is greater than PCN, the FDC sets the DIRECTION signal to a "0" (step in toward spindle) and issues STEP pulses to the drive.

The step rate is determined by the SRT value in the Specify command. After each STEP pulse is issued, NCN is compared with PCN. When NCN equals 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 processor is activated; the host processor must perform a Sense Interrupt Status command to determine if the seek was successful.

During the command phase, the FDC is in the busy state, but during the execution phase, the FDC is in its non-busy state. When the FDC is not busy, a Seek command may be issued to another drive. In this manner, parallel (overlapped) seek operations may be performed on up to four drives.

If the drive's READY signal is not active at the beginning of the command execution phase or if it goes inactive during the seek operation, the NR bit in Status Register 0 is set and the command is terminated.

3-23. READ DATA

The Read Data command requires nine bytes to load the command and command data. Following command execution, the host processor must read seven bytes from the FDC to complete the result phase.

PHASE	R/W	DATA BUS D7 D6 D5 D4 D3 D2 D1 D0	Remarks
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 from diskette to host memory
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	

After the last command byte is received, the FDC loads the read/write heads (if they are not loaded) at the current cylinder location, waits the specified head-settling time and then begins reading sector ID fields to locate the addressed sector (the sector number specified in byte 5 of the command). When the addressed sector is located, the FDC remains in the read mode to locate the data address mark at the beginning of the data field. After reading the data address mark, the FDC assembles the serial data from the data field into 8-bit bytes that are transferred, under direction of the DMAC, from the FDC's data register to host memory.

The number of bytes transferred is dependent on the word count value loaded into the DMAC. If the word count specified is less than a full sector, only the number of bytes specified are transferred; the FDC always reads a complete sector (to access the data field CRC bytes) in order to verify the data transfer. Conversely, if the word count value specified is greater than a sector, the FDC internally increments the sector number (R+1) and begins reading from the next logical sector on the track. The FDC will continue to read sectors until the word count is satisfied (terminal count reached) or until the last logical sector on the track is reached. If the MT (Multitrack) bit is set in the command byte, a transfer beginning on side 0 of a double-sided drive can extend to the last sector on side 1 (of the same cylinder). Table 3-10 outlines the EOT and GPL command byte values for the various sector sizes in both the FM and MFM recording modes.

When the transfer is complete (or if the transfer cannot be completed), the FDC interrupts the host processor, initiates the head-unload timeout, and enters the result phase. The host processor must then read the seven result bytes from the FDC's data register. The first three bytes read are the ST0, ST1, and ST2 status register bytes that indicate the outcome of the operation. The last four bytes reflect the updated values for C, H, R, and N (result phase ID information) when the FDC completes command execution. Table 3-11 specifies the ID information for normal (no error detected) command termination.

During normal read operations (SK bit in command byte not set), if a deleted data address mark is encountered at the beginning of a sector's data field, the data from the deleted sector is transferred to host memory and the read operation is terminated (irrespective of the word count specified). The CM

Table 3-10. Command Byte Values

Mode	Bytes Per Sector (decimal)	Sector Size (N)	Standard 8-inch Drives: Sectors Per Track (decimal)	Last Sector (EOT)	Gap3 Length (GPL) R/W	Gap3 Length (GPL) Format	5¼-inch Mini Drives: Sectors Per Track (decimal)	Last Sector (EOT)	Gap3 Length (GPL) R/W	Gap3 Length (GPL) Format
Single Density (MFM=0)	128	00	26	1A	07	1B	18	12	07	09
	128	00	—	—	—	—	16	10	10	19
	256	01	15	0F	0E	2A	8	08	18	30
	512	02	8	08	1B	3A	4	04	46	87
	1024	03	4	04	47	8A	2	02	C8	FF
	2048	04	2	02	C8	FF	1	01	C8	FF
	4096	05	1	01	C8	FF	—	—	—	—
Double Density (MFM=1)	256	01	26	1A	0E	36	18	12	0A	0C
	256	01	—	—	—	—	16	10	20	32
	512	02	15	0F	1B	54	8	08	2A	50
	1024	03	8	08	35	74	4	04	80	F0
	2048	04	4	04	99	FF	2	02	C8	FF
	4096	05	2	02	C8	FF	1	01	C8	FF
	8192	06	1	01	C8	FF	—	—	—	—

Unless otherwise specified, all values are in hexadecimal.

(Control Mark) bit in Status Register 2 will be set, and the R (sector number) result byte will contain the number of the deleted sector. During read operations with the SK bit set (SK=1), if a deleted data address mark is encountered, the FDC skips over the deleted sector and reads the next logical (non-deleted) sector. When the operation is complete (i.e., when terminal count is reached), the CM bit will be set to indicate that a deleted sector was encountered during the transfer, and the R result byte will be incremented to the next sequential sector number.

When a Read Data command cannot be completed due to an error condition, the FDC sets the Interrupt Code (IC) bits in Status Register 0 to indicate abnormal termination of the command (D7 = 0, D6 = 1) and sets specific bits in Status Registers 1 and 2 to indicate the nature of the error.

During DMA transfers between the controller and host memory, the FDC must be serviced within 27 us in the FM mode or within 13 us in the MFM mode to prevent data from being overwritten. If the FDC is not serviced within the above time limits (usually caused by a bus contention problem), the FDC terminates the transfer (abnormal termination) and sets the OR (Overrun) bit in Status Register 1.

3-24. READ DELETED DATA

The Read Deleted Data command, like the Read Data command, requires nine bytes to load the command and command data. Following command execution, the host processor must read seven bytes from the FDC to complete the result phase.

Execution of a Read Deleted Data command is identical to the Read Data command description in the previous section except for the handling of data field address marks. During a read deleted data operation in the non-skip mode (SK bit = 0 in the command byte), if a (normal) data address mark is encountered at the beginning of the sector's data field, the data from the sector is transferred and the read operation is terminated (irrespective of the word count specified). The CM bit in Status Register 2 will be set, and the R result byte will contain the number of the

Table 3-11. Result Phase ID Information

MT	EOT	Last Sector Transferred	Result Phase ID Information			
			C	H	R	N
0	1A 0F 08 04 02	Sector 1 thru 25 (Side 0 or 1) Sector 1 thru 14 (Side 0 or 1) Sector 1 thru 7 (Side 0 or 1) Sector 1 thru 3 (Side 0 or 1) Sector 1 (Side 0 or 1)	NC*	NC	R+1	NC
	1A 0F 08 04 02 01	Sector 26 (Side 0 or 1) Sector 15 (Side 0 or 1) Sector 8 (Side 0 or 1) Sector 4 (Side 0 or 1) Sector 2 (Side 0 or 1) Sector 1 (Side 0 or 1)	C+1	NC	R=01	NC
1	1A 0F 08 04 02	Sector 1 thru 25 (Side 0 or 1) Sector 1 thru 14 (Side 0 or 1) Sector 1 thru 7 (Side 0 or 1) Sector 1 thru 3 (Side 0 or 1) Sector 1 (Side 0 or 1)	NC	NC	R+1	NC
	1A 0F 08 04 02 01	Sector 26 on Side 0 Sector 15 on Side 0 Sector 8 on Side 0 Sector 4 on Side 0 Sector 2 on Side 0 Sector 1 on Side 0	NC	H=01	R=01	NC
	1A 0F 08 04 02 01	Sector 26 on Side 1 Sector 15 on Side 1 Sector 8 on Side 1 Sector 4 on Side 1 Sector 2 on Side 1 Sector 1 on Side 1	C+1	H=00	R=01	NC

*NC = no change

PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	Remarks
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 infor-
	W	H								mation prior to
	W	R								Command execution
	W	N								
	W	EOT								
	W	GPL								
	W	DTL								
Execution										Data transfer from diskette to host memory
Result	R	ST 0								Status information
	R	ST 1								after Command
	R	ST 2								execution
	R	C								
	R	H								Sector ID infor-
	R	R								mation after Command
	R	N								execution

non-deleted sector. During execution of a Read Deleted Data command with the SK bit set in the command byte, if a (normal) data address mark is encountered, the FDC skips over the "non-deleted" sector and reads the next logical deleted sector. When the operation is complete (i.e., when terminal count is reached), the CM bit will be set to indicate that a non-deleted sector was encountered during the transfer and the R result byte will be incremented to the next sequential sector number.

3-25. READ ID

The READ ID command requires two bytes to load the command and command data. Following command execution, the host processor must read seven bytes from the FDC to complete the result phase.

PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Command	W	0	MFM	0	0	1	0	1	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
Execution										The first correct ID information on the track is stored in Data Register
Result	R	ST 0								Status information
	R	ST 1								after Command
	R	ST 2								execution
	R	C								
	R	H								Sector ID infor-
	R	R								mation during Execution
	R	N								Phase

The Read ID command allows the host processor to verify the current position of the drive's read/write heads without initiating a data transfer. During command execution, the FDC loads the heads and waits the specified head-settling time (if the heads are unloaded) and begins searching for a sector ID field. When the first valid ID field is read, the FDC interrupts the host processor, initiates the head-unload timeout, and enters the result phase. The host processor must read the seven result bytes to complete the command; the C, H, R, and N result bytes contain the corresponding ID field byte values read from the ID field (i.e., the value returned in the C result byte indicates the current track address).

During the command execution phase, if an ID field address mark cannot be found within one full revolution of the diskette (e.g., if the track is not formatted), the FDC sets the MA (missing address mark) bit in Status Register 1 and sets the interrupt code bits in Status Register 0 to indicate abnormal termination.

3-26. READ TRACK

The Read Track command requires nine bytes to load the command and command data. Following command execution, the host processor must read seven bytes from the FDC to complete the result phase.

PHASE	R/W	DATA BUS D7	D6	D5	D4	D3	D2	D1	D0	Remarks
Command	W	0	MFM	0	0	0	0	1	0	Command Codes
	W	0	0	0	0	0	HDS	DS1	DS0	
	W	C								Sector ID infor-
	W	H								mation prior to
	W	R								Command execution
	W	N								
	W	EOT								
	W	GPL								
	W	DTL								
Execution										Data transfer from diskette to host memory. FDC transfers all sectors on track beginning with first sector following index
Result	R	ST 0								Status information
	R	ST 1								after Command
	R	ST 2								execution
	R	C								
	R	H								Sector ID informa-
	R	R								tion after Execu-
	R	N								tion Phase for this command.

The Read Track command operates similar to the Read Data command except that all sector data fields on the addressed track are read, beginning with the first sector following index, in their order of physical appearance on the track. During the transfer the FDC ignores ID and data field CRC errors and deleted data address marks (i.e., sectors with errors and deleted sectors are transferred). Note that multitrack and skip operations are not permitted with the Read Track command.

3-27. WRITE DATA

The Write Data command requires nine bytes to load the command and command data. Following command execution, the host processor must read seven bytes from the FDC to complete the result phase.

PHASE	R/W	DATA BUS								Remarks
		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
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	GPL								
	W	EOT								
	W	DTL								
Execution										Data transfer from host memory to diskette
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								

The command data bytes for the Write Data command are identical to the command data bytes for the Read Data command (see section 3-23). After the last command byte is received, the FDC loads the read/write heads, waits the specified head-settling time, and then begins reading sector ID fields to locate the addressed sector (the sector number specified in byte 5). After the addressed sector is located, the FDC switches the drive to the write mode during the post ID field gap and updates (writes) the sector's sync field and data field address mark. Immediately after writing the address mark, the FDC begins writing the data bytes received at its data register onto the diskette as a serial bit stream.

The number of bytes written is dependent on the word count value loaded into the DMAC. If the word count specified is less than a full sector, only the number of bytes specified are transferred to the FDC; the FDC fills the remainder of the data field with zeros. After the last data field bit is written, the FDC writes the 16-bit data field CRC character. If the word count specified is greater than a sector, the FDC internally increments the sector number and, after locating (reading) the next sector's ID field, begins writing the sector data field. The FDC will continue to write sectors until the word count is satisfied (terminal count reached) or until the last logical sector on the track is written. If the Multitrack bit is set in the command byte, a write operation beginning on side 0 of a double-side drive can extend to the last sector on side 1 (of the same cylinder).

When the transfer is complete (or if the transfer cannot be completed), the FDC interrupts the host processor, initiates the head-unload timeout, and enters the result phase. The host processor must then read the seven result bytes from the FDC's data register. As described under the Read Data command, the first three bytes indicate the outcome of the operation, and the last four bytes reflect the updated values for C, H, R, and N.

During DMA transfers between the controller and host memory, the FDC must receive the data byte to be written within 31 us in the FM mode or within 15 us in the MFM mode to prevent data from being underwritten. If the FDC is not serviced within the above time limits (usually caused by a bus contention problem), the FDC terminates the operation (abnormal termination) and sets the OR (Overrun) bit in Status Register 1.

3-28. WRITE DELETED DATA

The Write Deleted Data command is identical to the Write Data command previously described except that a deleted data address mark is written at the beginning of the sector's data field in place of a data address mark.

PHASE	R/W	DATA BUS								Remarks
		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
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	GPL								
	W	EOT								
	W	DTL								
Execution										Data transfer from host memory to diskette
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								

3-29. FORMAT TRACK

The Format Track command requires six bytes to load the command and command data. Following command execution, the host processor must read seven bytes from the FDC to complete the result phase.

PHASE	R/W	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
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				Filler Byte
Execution										FDC formats an entire track
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				

The Format Track command formats or "initializes" a track by writing the ID field, gaps, sync bytes, and address marks for each sector. The track to be formatted is determined by the position of the read/write heads on the diskette. Prior to command execution, a table in memory containing the ID field values (track address, head address, sector number and sector size) for each sector on the track must be prepared. (During command execution, the FDC uses the values from the table to write the individual ID fields.) Referring to the track format illustration (figure 3-3), address marks are written automatically by the FDC. The track (C) and head (H) addresses, sector number (R) and sector size (N) byte values to be written into the ID field are taken, in order, from the table. The ID field CRC character is derived from the ID address mark and ID field data, and is written immediately following the ID field. Gaps and sync fields are written automatically by the FDC; the length of the post data field gap (gap 3) is determined by the GPL command data byte value. The number of data bytes per sector and the number of sectors per track are determined by the N and SC command data byte values; the data pattern written into each byte of each sector's data field is determined by the D command data byte value. The data field CRC character is derived from the data address mark and the data written in the sector's data field.

The order of sector number assignment on the track is taken directly from the formatting table in memory. Four entries are required for each sector: a track address, a head address, the sector number and a sector size. Note that the order of sector number entries in the table is the sequence in which sector numbers appear on the track when it is formatted. The number of 4-byte entries in the table must equal the number of sectors on the track. Caution must be exercised when creating the formatting table since entries are not verified by the FDC and it is possible to format a track with an illegal, redundant, or missing sector number.

Since the sector number is taken directly from the formatting table, tracks can be formatted either sequentially (the first sector following the index mark is assigned sector number 1, the next adjacent sector is assigned sector number 2, and so on) or sector numbers can be "interleaved" on a track.

The sequential sector format optimizes sector access times during multisector transfers by permitting a number of sectors (up to an entire track) to be transferred within a single revolution of the diskette. Sector interleaving is used when a number of logically-consecutive sectors are to be transferred individually and the processing time between adjacent sectors is greater than the time required to access the next sector.

As an example of sector interleaving, assume that a number of consecutive sectors are to be transferred individually on both a sequentially formatted track and on a track that utilizes sector interleaving. On a sequentially formatted track, assuming that the amount of processing time required between sectors is greater than the time required to access the next sector, the diskette must rotate nearly a full revolution to access the next sector to be transferred. Since one diskette revolution requires approximately 167 milliseconds, to transfer an entire track of 15 sectors, 15 revolutions, or 2.5 seconds, are required. Conversely, if sector numbers are assigned with an interleaving factor of three (see figure 3-4), the processing time between logically-adjacent sectors is increased substantially and, if sufficient, allows the complete track to be transferred in three revolutions of the diskette (500 milliseconds).

The following table (table 3-12) describes the organization of the formatting table that would be used to format the diskette shown in figure 3-4.

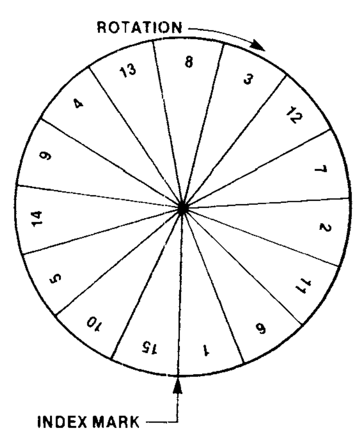


Figure 3-4. Sector Interleaving 143078-12

Table 3-12. Formatting Table

Byte	Function	Data Contents (Hexadecimal)
1	Track Address (C)	XX
2	Head Address (H)	0X
3	Sector Number 1 (R)	01
4	Sector Size (N)	0X
5	Track Address	XX
6	Head Address	0X
7	Sector Number 6	06
8	Sector Size	0X
9	Track Address	XX
10	Head Address	0X
11	Sector Number 11	0B
12	Sector Size	0X
.		
.		
.		
53	Track Address	XX
54	Head Address	0X
55	Sector Number 10	0A
56	Sector Size	0X
57	Track Address	XX
58	Head Address	0X
59	Sector Number 15	0F
60	Sector Size	0X

Following the command phase, the FDC loads the read/write heads at the current cylinder location, waits the specified head-settling time and monitors the INDEX signal from the drive. When the INDEX signal goes active (index hole detected), the FDC begins formatting the track according to figure 3-3. After writing the ID address mark for the first sector, the FDC writes the first four bytes from the format table into the ID field (the FDC initiates a DMA transfer and receives the requested bytes at its data register). The FDC writes the remainder of the sector based on the command data received during the command phase. After writing the first sector's post-data field gap, the FDC writes the sync field and ID address mark for the next sector and writes the next four bytes from the format table into the second sector's ID field. This formatting operation continues until the FDC writes the number of sectors specified by the SC command data byte. When the index mark is encountered, the FDC interrupts the host processor, initiates the head-unload timeout, and enters the result phase. The host processor must read the seven result bytes to complete the command. Note that the C, H, R, and N result bytes are irrelevant with the Format Track command.

Prior to formatting a track, the DMAC's base and current address registers (and, if required, the controller's segment address registers) must point to the first byte of the format table in memory. The DMAC word count specified must be equal to (or greater than) the number of byte entries in the format table (i.e., to format a track with 26 sectors, 104 bytes must be transferred).

3-30. RECALIBRATE

The Recalibrate command requires two bytes to load the command and command data.

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
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

The Recalibrate command positions the drive's read/write heads at a known track position and is used following power-up or a seek error (e.g., WC bit set in Status Register 2). During command execution, the FDC sets the PCN (present cylinder number) counter to zero and monitors the TRACK0/ signal from the drive. As long as the TRACK0/ signal remains inactive, the FDC holds the DIRECTION/ signal high (1) and issues up to 77 STEP/ signals to the drive (to step the read/write heads toward track 0). When the TRACK0/ signal goes active, the FDC interrupts the host processor and sets the SE (seek end) bit in status register 0. Since the Recalibrate command does not have a result phase, the host processor must issue a Seek Interrupt Status command to properly terminate the Recalibrate command and to clear the interrupt.

During the command phase, the FDC is in a busy state, but during the execution phase, the FDC is in a non-busy state. When the FDC is not busy, a recalibrate (or seek) command can be issued to another drive. In this manner, parallel (overlapped) recalibrate operations can be performed on up to four drives concurrently.

During the execution phase, if the TRACK0/ signal does not go active following 77 STEP/ pulses, the FDC sets both the EC (equipment check) and the SE bits in Status Register 0 and interrupts the host processor.

NOTE

When executing a Recalibrate command on a drive with more than 77 tracks, if the drive's read/write head is positioned on track 77 or greater when the command is executed, an abnormal termination will result (EC bit set in status register 0). A second Recalibrate command must be issued to complete the recalibrate operation and to position the read/write head over track 0.

3-31. SENSE DRIVE STATUS

The Sense Drive Status command requires two bytes to load the command and command data. Note that there is no execution phase associated with the command, and no interrupt is generated. After the command is loaded, the host processor must read one result byte to complete the result phase.

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
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 regarding selected drive

The Sense Drive Status command is used to interrogate the FDC regarding the status of the drive selected during the command phase. The result byte read (Status Register 3) contains the drive status information (see section 3-5).

3-32. SENSE INTERRUPT STATUS

The Sense Interrupt Status command requires one byte to load the command. Note that there is no execution phase associated with the command. After the command is loaded, the host processor must read two result bytes to complete the result phase.

PHASE	R/W	DATA BUS D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
Command	W	0	0	0	0	1	0	0	0	Command Codes
Result	R	ST 0								Status information following a seek or recalibrate operation
	R	PCN								

The host processor issues a Sense Interrupt Status command to effectively terminate a Seek or Recalibrate command (the PCN result byte defines the current position of the read/write head) or whenever an unexpected interrupt is received from the FDC (the Sense Interrupt Status command clears the interrupt signal). The host processor, by reading bits 5, 6, and 7 of Status Register 0, can readily identify the cause of the interrupt as follows:

Interrupt Code Bit 7	Interrupt Code Bit 6	Seek End Bit 5	Cause
1	1	0	READY/ signal from drive changed state.
0	0	1	Normal Termination of Seek or Recalibrate command.
0	1	1	Abnormal Termination of Seek or Recalibrate command.

3-33. INVALID COMMANDS

If the host processor issues either a Sense Interrupt Status command when no interrupts are pending or a command code not recognized by the FDC, the FDC immediately terminates the command phase and enters the result phase without generating an interrupt (i.e., the FDC enters a stand-by state or simply "goes to sleep"). To wake the FDC and to complete the result phase, the host processor must read a result byte from the FDC's data register; the ST0 result byte read will indicate that an invalid command was issued (bit 7=1, bit 6=0).

The ability of the FDC to recognize a Sense Interrupt Status command as an invalid command when no interrupt is pending allows the host processor to locate or "flush out" possible "hidden interrupts." (A hidden interrupt occurs during overlapped seek or recalibrate operations when a second drive completes its seek or recalibrate and the interrupt from the first drive is still pending or when more than one drive ready status change occurs before the first interrupt is cleared.) By continuing to issue Sense Interrupt Status commands until invalid command status is received, the host processor can be assured that all interrupts have been acknowledged.

Note that while in the stand-by state, the FDC cannot generate an interrupt. This fact provides a mechanism by which the host processor essentially can shut out the controller when a critical task is being performed and an interrupt from the controller cannot be tolerated.

3-34. SOFTWARE

The host software requirements for the controller consist of a set of Input/Output driver routines to perform the following tasks:

- Initialize the controller, including the FDC and DMAC, following power-up.
- Issue commands to the FDC and pass command data to the FDC and DMAC.
- Respond to completion interrupts and interpret results from the FDC.
- Handle errors.

Appendix A provides two example drivers for the iSBC 208 board. A PL/M-86 driver is given for 16-bit systems and an assembly language driver is given for 8080/8085 (8-bit) systems.

In the example I/O drivers in Appendix A, subroutines are written to perform disk I/O transfers. These subroutines:

- Are user-callable
- Pass command data via an I/O parameter block
- Wait if the FDC is busy

As decribed previously in this chapter, all FDC commands that transfer data to or from the diskette require multiple bytes of information from the host processor before the command can be executed. In addition to the information passed to the FDC, the DMAC requires information as to the number of bytes to be transferred, the starting location in memory for the transfer, and the direction of the transfer. Also, if memory addresses greater than 64K are required, the controller's segment address register must be programmed. The I/O driver routines communicate all of this information to the controller through a user-programmed I/O parameter block (IOPB). Figure 3-5 shows the IOPB used by the PL/M-86 sample driver in Appendix A.

Some bytes of the IOPB are dynamic (e.g., the track and sector numbers) and must be written into the IOPB by the calling program prior to the call. Other bytes remain fixed during program operation (e.g., bytes per sector, sectors per track) and can be declared when the driver is compiled. Some commands do not use all the parameter bytes. The IOPB may be located anywhere in host memory convenient to the user program.

The IOPB used by the Assembly Language sample driver is structured the same way as shown under each of the FDC command descriptions with the addition of a NSEC parameter used by some commands to specify the number of sectors to be transferred. An example using the assembly language driver is shown in Appendix A.

3-35. INITIALIZATION

Figure 3-6 depicts a typical initialization sequence. Initialization requires applying power to the controller and drives, resetting the controller (resets the FDC and DMAC), programming the operating mode of the DMAC, specifying the drive parameters to the FDC, and, once the drive parameters are specified, positioning the drive's read/write heads to a known position.

3-36. PROGRAMMING THE DMAC

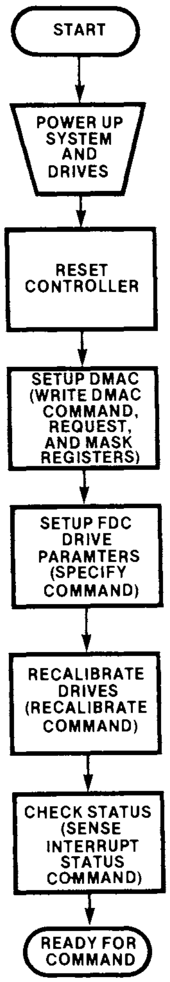
Once the DMAC has been initialized, programming the DMAC for a subsequent diskette data transfer includes loading the starting (offset) memory address (DMAC Address Register), the number of bytes to be transferred (DMAC Word-Count Register), the direction of the transfer (DMAC Mode Register), and clearing the channel 0 mask bit. Also, if memory addresses greater than 64K are to be used, the controller's segment address registers must be loaded. Note that since FDC command execution begins automatically after the last command data byte is received, the DMAC must be completely programmed before the last command byte is output to the FDC.

3-37. PROGRAMMING THE FDC

Figure 3-7 shows a generalized program flow chart for the FDC's command phase. The twelve commands are broken down into Specify, Sense Drive Status, Seek, Recalibrate, and the six data transfer commands. The remaining comand, Sense Interrupt Status, only is issued in response to an interrupt.

15 — 8	7 — 0	
TRACK ADDRESS	INSTRUCTION NUMBER	WORD 1
SECTOR NUMBER	HEAD AND DRIVE ADDRESSES	WORD 2
MT/MFM/SK BYTE	SECTORS PER TRACK	WORD 3
BYTES PER SECTOR	GAP3 LENGTH	WORD 4
NUMBER OF BYTES TO BE TRANSFERRED		WORD 5
SEGMENT ADDRESS		WORD 6
OFFSET ADDRESS		WORD 7

Figure 3-5. I/O Parameter Block 143078-13



143078-14

Figure 3-6. Initialization Flow Chart

Figure 3-8 shows the detailed steps of the command phase. The "serial" commands (e.g., Read Data) require 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 being executed, but must wait for the FDC to become idle. The two entry points "Command FDC Serial" and "Command FDC Parallel" are used in the flowchart in figure 3-8.

The Specify command establishes the timing intervals for the FDC's three internal timers and typically is issued only during initialization. The Specify command has neither an execution phase nor a result phase and does not generate a completion interrupt.

The Sense Drive Status command is issued between other commands to obtain the status of any particular drive. The status of a drive is available immediately; the Sense Drive Status command does not have an execution phase and does not generate a completion interrupt.

The Seek command is used to position the addressed drive's read/write heads over the desired track location prior to issuing a subsequent data transfer command; the Recalibrate command is used to position a drive's read/write heads over a known track position (track 0) and is issued following system initialization or a seek error. During the execution phase of these commands, the FDC enters a non-busy state, and concurrent seek or recalibrate operations can be initiated on the other drives. Following command execution, the FDC generates a completion interrupt; a Sense Interrupt Status command must be issued in response to the interrupt to complete a Seek or Recalibrate command.

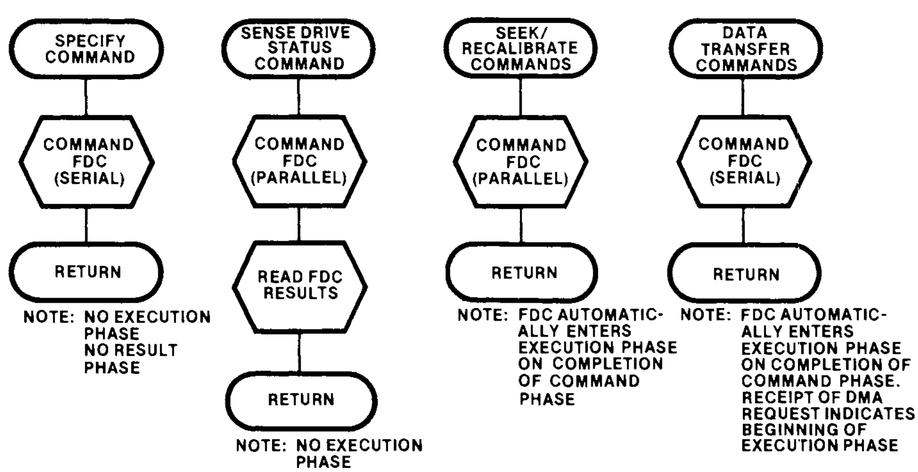


Figure 3-7. FDC Command Phase Flow Chart

143078-15

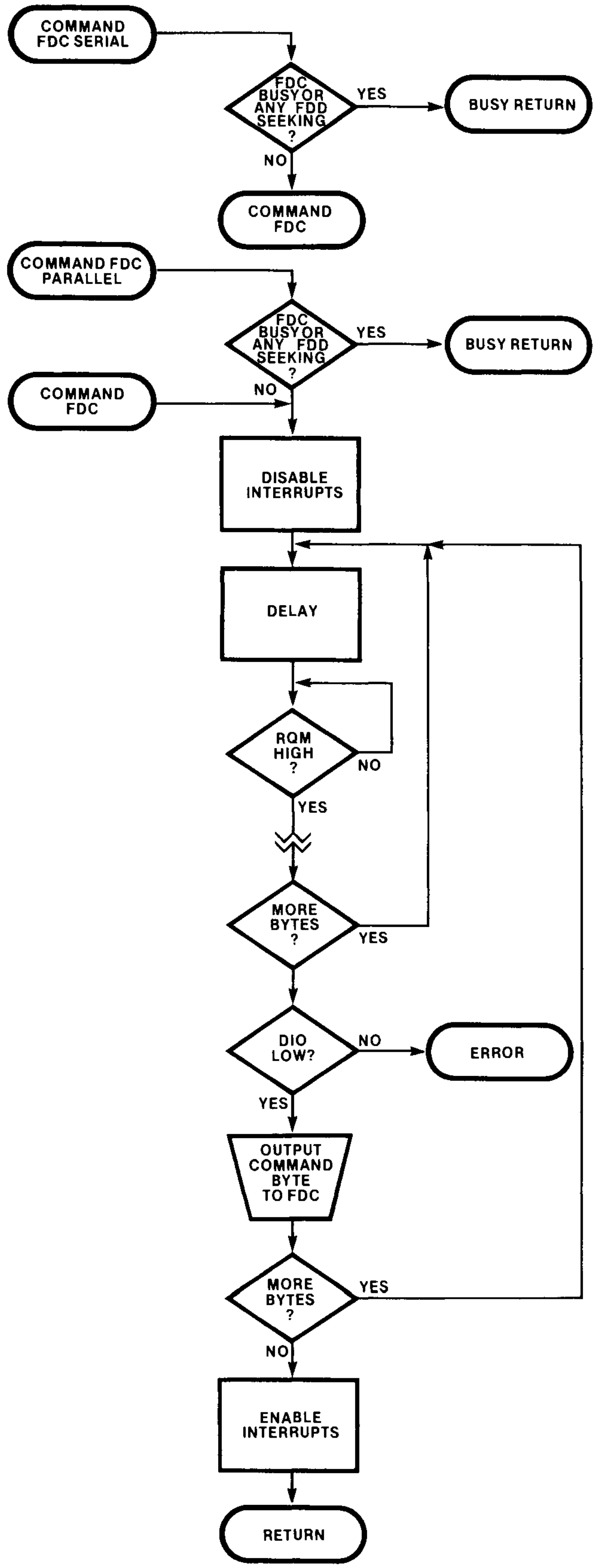


Figure 3-8. Serial/Parallel Command Phase Flow Chart 143078-16

The seven data transfer commands (Read Data, Read Deleted Data, Write Data, Write Deleted Data, Read Track, Read ID, and Format Track) all have an execution phase and a result phase, and all generate a completion interrupt.

Figure 3-9 shows the detailed steps necessary to complete a command's result phase. When a data transfer command terminates, the FDC generates an interrupt and begins the result phase. The host processor must read a series of result bytes from the FDC's data register to complete the result phase; the interrupt is cleared automatically when the last result byte is read. As shown in figure 3-9, the host processor does not need to count the number of result bytes read; the host processor can determine when the result phase is complete by checking the FDC Busy bit in the main status register. (When the Busy bit goes inactive, all result bytes have been read and the FDC is ready for a new command.)

3-38. INTERRUPT PROCESSING

When a data transfer command is completed, the FDC interrupts the host processor and enters the result phase. The host processor must then read a series of result bytes from the FDC's data register to complete the command; the interrupt is cleared automatically by the FDC when the last result byte is read.

When an interrupt results from the completion of a Seek or Recalibrate command, the host processor must issue a Sense Interrupt Status command to

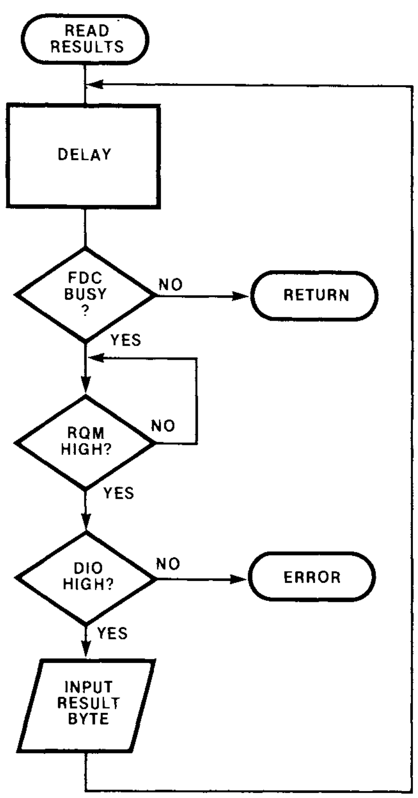


Figure 3-9. Result Phase Flow Chart 143078-17

properly terminate the command (Seek and Recalibrate commands do not have a result phase and rely on the result phase of the Sense Interrupt Status command to complete the operation).

Unexpected interrupts (i.e., interrupts that result from a change in a drive's ready status) also are cleared by a Sense Interrupt Status command. Note that if a Sense Interrupt Status command is issued when no interrupts are pending, Status Register 0 will indicate that an invalid command was issued. Accordingly, when servicing an unexpected interrupt or an interrupt resulting from the execution of a Seek or Recalibrate command, the host processor should continue to issue Sense Interrupt Status commands until an Invalid Command status is received to ensure that all "hidden" interrupts are serviced. (Hidden interrupts occur when a second interrupt is received before the first interrupt is cleared.)

Figure 3-10 illustrates the processing of FDC interrupts.

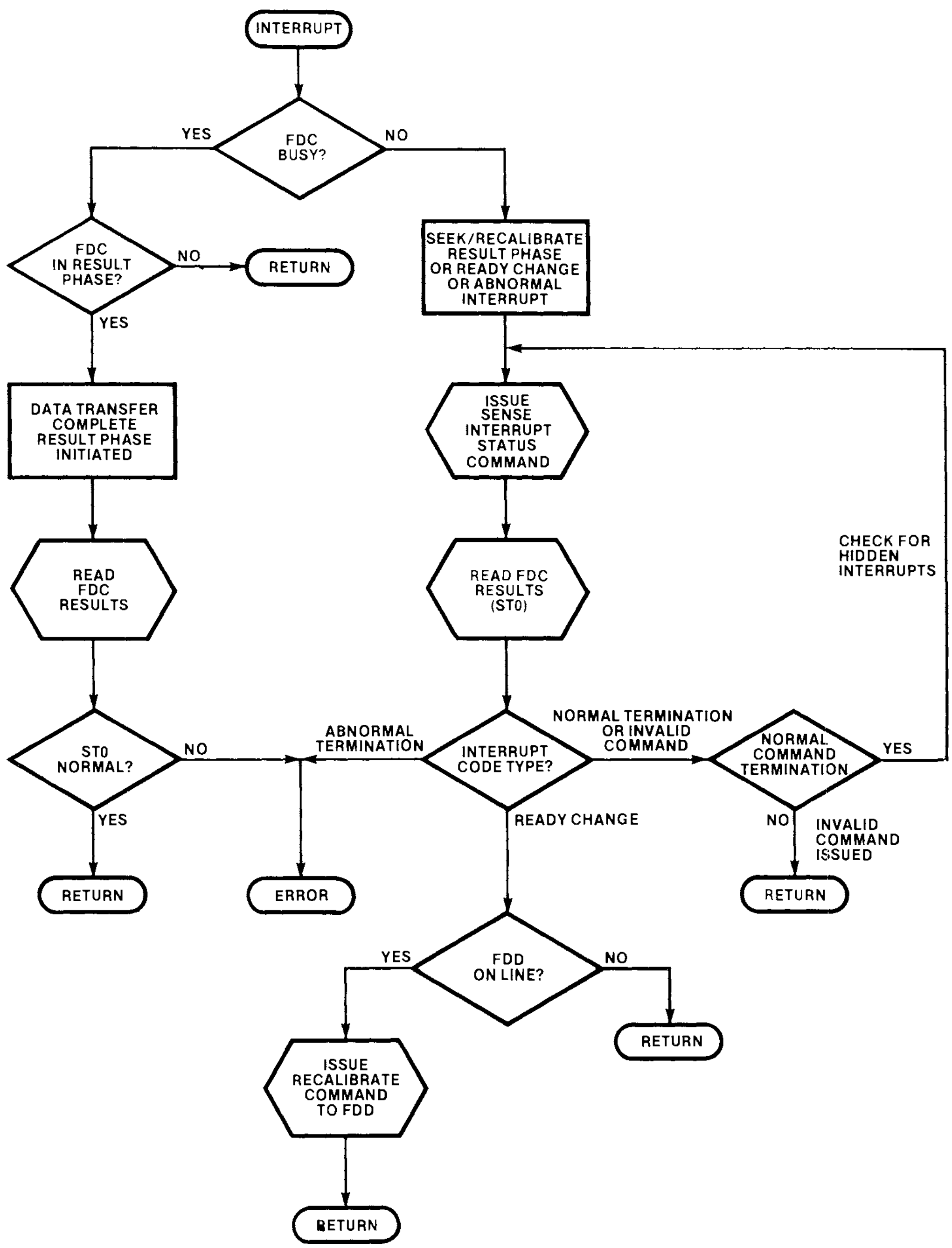


Figure 3-10. Interrupt Processing Flow Chart 143078-18

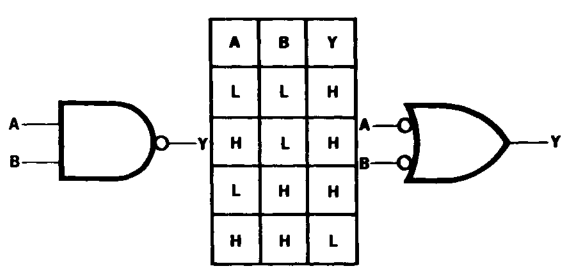
4-1. INTRODUCTION

This chapter explains the circuit operation of the iSBC 208 controller board. The level of the following discussion assumes that the reader has a working knowledge of digital electronics and has access to the individual component descriptions of all integrated circuits employed on the board. As a prerequisite, the reader should be familiar with the programming conventions outlined in Chapter 3 of this manual and the functional operation of both the host processor and the Multibus interface. Familiarity with the diskette drive interface specifications and operation also will prove beneficial in the comprehension of controller operation.

4-2. SCHEMATIC INTERPRETATION

The controller PC board schematic consists of seven individual sheets that are labeled Sheet 1 of 7, Sheet 2 of 7, etc. These drawings (figure 5-2) and the PC board assembly drawing (figure 5-1) are located in Chapter 5.

Schematic logic symbols follow active-state conventions in the positioning of the inversion symbol. A gate with an inversion symbol at its output is active in its low state, and a gate without an inversion symbol is active in its high state. Logic gating symbols are drawn according to their circuit function rather than by manufacturer's definition. For example, the gate shown in figure 4-1, depending on its application, would be drawn in either of the two configurations shown.



A	B	Y
L	L	H
H	L	H
L	H	H
H	H	L

Figure 4-1. Logic Conventions 143078-19

The gate configuration on the left (positive NAND), in figure 4-1, indicates that the required low-level output results from a logic high level at both inputs (AND function), and the gate configuration on the right (negative OR) indicates that the required high-level output results from a logic low level at either (or both) input (OR function).

In addition to the inversion symbol convention, signal nomenclature also follows an active state convention. When a signal (or level) is active in its logic low state, the signal mnemonic is followed by a slash (e.g., RST/). Conversely, when a signal is active in its high state, the slash is omitted from the signal mnemonic (e.g., RST).

4-3. FUNCTIONAL DESCRIPTION

The following subsections describe the operation of the individual circuit modules or blocks that compose the iSBC 208 controller. Figure 4-4, located at the conclusion of this chapter, is the functional block diagram of the controller and shows the interrelationship of the various blocks.

4-4. CLOCK AND TIMING CIRCUITRY

All timing and clock signals generated by the controller originate from 8-MHz crystal oscillator G1 (6ZD7). The buffered output of G1 drives timing generator U50 (3ZD6), provides a comparator frequency to the VCO frequency indicator on sheet 2, supplies the 8-MHz clock for the FDC, and drives flip-flop U67 (6ZD7). Flip-flop U67 is used as a frequency divider to provide the 4-MHz clock signal. This signal also drives write shift register U3 (7ZB5) and binary counter U69 (6ZD6). U69 produces frequencies of 2 MHz (U69-11), 1 MHz (U69-10), 0.5 MHz (U69-9), and 0.25 MHz (U69-8). Three of these frequencies (1, 0.5, and 0.25 MHz) are input to data selectors U70 (6ZD4) and U71 (6ZD3). The data selectors provide the correct clocks to the various circuits as determined by the MFM signal from the FDC and the MINI/ signal from the jumper matrix. (The MINI/signal is low for 5¼-inch drives and is high for 8-inch drives; the MFM signal is low for single-density operation and is high for double-density operation.) Table 4-1 shows the resultant output from U71 for the different states of the MINI/ and MFM signals.

Table 4-1. IC U71 Output

MINI/Low MFM Low	MINI/High MFM High	MINI/Low MFM High	MINI/High MFM Low
1Y = 2200 ns pulse 2Y = 0.25 MHz 4Y =0.25 MHz	1Y = 550 ns pulse 2Y = 1 MHz 4Y = 1 MHz	1Y = 1100 ns pulse 2Y = 0.5 MHz 4Y = 0.5 MHz	1Y = 1100 ns pulse 2Y = 0.5 MHz 4Y = 0.5 MHz

The output from U71-12 clocks flip-flop U54 at the selected frequency. The flip-flop is cleared by the 2-MHz input at U54-1 to create a 250 ns write clock pulse at the selected frequency. Table 4-2 lists the various write clock frequencies. The outputs from U71-4 and U71-7 are used in the data separator circuits (see paragraph 4-12).

Table 4-2. Write Clock Frequency

Mode	Drive Size	
	8inch	5¼-inch
Single density Double density	0.5 MHz 1.0 MHz	0.25 MHz 0.5 MHz

4-5. MULTIBUS INTERFACE

The bidirectional Multibus interface data lines are buffered by U64 (1ZC6). This circuit is enabled for both I/O and DMA type operations by U14 (1ZB6). The BDSEL/ signal enables the data driver for I/O transfers, and the DMAC/ signal enables the drivers for DMA transfers. The IOR/ signal controls the direction of the data buffer. When the IOR/ signal is active (low), U64 is in the output mode (I/O READ or DMA WRITE operations) and when the IOR/ signal is inactive (high), U64 is in the input mode (I/O WRITE or DMA READ operations).

Bidirectional address buffering for ADR0/ - ADRF/ is provided by U62 and U63 (5ZC2). These buffers are always enabled. Normally, the ADEN (5ZA7) signal is low to allow the address to enter the board (I/O transfers). When ADEN is high, the address buffers place the address on the Multibus interface (DMA operations only). Address lines ADR10/ - ADR13/ are buffered by U57 (5ZB2) and are used only for DMA operations.

The iSBC 208 controller I/O address decode circuitry decodes either 8- or 16-bit I/O addresses and occupies either 22 or 38 ports. This complex I/O mapping is performed by the Intel 3625 PROM at U16 (2ZC4). The PROM provides a board enable signal (SEL/) and an encoded number (0-7) for I/O circuit selection. The encoded number is decoded into eight individual chip select signals by U27 (2ZC2). Table 4-3 lists the base addresses and shows the resulting chip select signal.

A comparison between the I/O address and the base address jumpers is performed by comparators U28, U41, and U40 (2ZC6, 2ZB6). Since the A = B output from U28 is true for both 8- and 16-bit I/O addresses, this output is used to enable the PROM's CS2 input (the CS1 input is permanently enabled by a resistor to ground). For 8- or 16-bit address decode selection, the PROM ignores the A = B output (U40-6) when the PROM's A9 input is high (16-bit mode). When the PROM's A9 input is low (8-bit mode), a high output from comparator U40-6 is required in order to generate the SEL/ signal. When the MPST/ signal is high (no Multimodule installed), the PROM only generates a SEL/ signal when inputs A5 and A6 match and produces the 22 ports that reside on 32-port boundaries. With an iSBX multimodule installed, MPST/ is held low to cause the PROM to ignore its A6 input and to produce the 38 ports on 64-port boundaries.

Bus control signals IORC/ and IOWC/ are inverted by U58 (3ZC7) and then ORed together by U26 (3ZD6) to generate a common command signal. This signal removes the clear input to shift registers U49 (3ZD5) and U50 (3ZD6). After 16 clock pulses (eight at 8 MHz and eight at 2 MHz), both registers are filled with "ones." The resistor outputs provide three delays that are used during I/O accesses. The first delay of 500 μs at U50-10(3ZD6) delays the I/O commands to allow time for the I/O address signals to propagate through the decode circuitry and additional time to meet the DMA controller's recovery

Table 4-3. Chip Select Coding

I/O Base Address (Hex)	U27 Output	Circuit Enabled
00 thru 0F	CS0/	DMAC
10, 11	CS1/	FDC
12	CS2/	Auxiliary Port
13	CS3/	Software Reset
14	CS4/	Low byte of Seg. Reg.
15	CS5/	High byte of Seg. Reg.
20 thru 27 28 thru 2F	MCS0/ MCS1/	iSBX (when installed)

time between active read/write pulses. This delayed signal, together with the common command and SEL/ signals, enables four three-state gates (U56) that in turn, pass IOR/ or IOW/ to the selected circuitry. Dual buffering is used on IOW/ and IOR/ for loading purposes. The second delay (U50-13) provides the acknowledge timing for all I/O accesses except the software reset. A third delay of approximately 4 μs is provided to meet the reset timing requirements of the 8272 FDC. The proper XACK/ timing is determined by the level of CS3/ through gates U47 and U65 (3ZD4).

4-6. DMA CONTROLLER (DMAC)

DMA controller U18 (4ZC6) mediates the flow of data between both the disk drive (through the FDD interface and the FDC) or the iSBX module (if installed) and the Multibus interface memory.

The following DMA controller modes of operation are not supported and cannot be used on the iSBC 208 controller:

- Cascade Mode
- Memory to Memory transfers
- Compressed Timing

A one-byte transfer from memory to the FDC will be used to describe a DMA operation. Since the timing for all DMA operations is basically the same, only differences from the one byte transfer will be described where appropriate.

Assuming the host processor has set up the DMAC and the FDC, the DMA process starts when the DMAC receives a request (DREQ0) from the FDC. The DMAC resolves priority among simultaneous requests and activates its HRQ output pin (U18-10) to inform the 8218 bus controller (U39) to acquire the system bus. The bus controller then activates BREQ/ and deactivates BPRO/ and then waits for BUSY/ to go inactive. When BUSY/ goes inactive, and if BPRN/ is active, the bus controller takes control of the system bus by activating BUSY/. The bus controller then notifies the DMAC to continue the transfer by activating the ADEN/ signal which, through inverter U15 (4ZA5), activates the HLDA input at U18-7.

The ADEN/ signal also inhibits the I/O address decoder via U28-3 (2ZC6) to prevent the generation of false chip-select signals caused by a match between a memory address and the I/O base address jumpers. The ADEN signal conditions the bidirectional system bus drivers (U62, U63) on sheet 5 to place the address on the Multibus interface.

After receiving HLDA, the DMAC proceeds with the actual data transfer by activating the following signals in the order listed:

1. DAEN (U18-9) is ANDed with ADEN at U66 (5ZB5) to enable three-state address buffers U46 (5ZC3), U57 (5ZB2), and U61 (5ZB3).
2. The high-order memory address byte is output on the on-board data bus as D0-D7 (U18), buffered by U33 (1ZB5) and latched into register U31 (5ZC6) by the address strobe (ASTB) signal. At the same time, the low-order memory address byte is output as A0-A3 and DA4-DA7 (U18).
3. DACK0/ (U18-25) is sent to enable the FDC (4ZD6).
4. MEMR (U18-3) is sent to the bus controller (U39) which, in turn, activates MRDC (U39-12) to produce a memory read command on the system bus. System memory now responds with a byte of data that passes through data buffer U64 (1ZC6) that has been enabled by the DMAC/ signal from the bus controller.
5. The IOWC/ signal from the Multibus interface is now sent to the FDC (U17-3 on sheet 7) to enable the FDC to accept the data byte.

Acknowledgment from the system memory (XACK/) is synchronized with DCLK at latch U22 (3ZD2). The REDY signal from U22 is passed to the DMAC (U18-6) to allow the DMAC to complete the transfer cycle. Completion of the transfer cycle takes place in the reverse order (see figure 4-2 for DMA transfer timing).

An end of process (EOP) signal is generated by the DMAC (U18-36) when its word count register reaches zero. The EOP signal is ANDed with DACK0 and sent as a terminal count (TC) signal (EOP0/) to inform the FDC that the last transfer has occurred.

4-7. DMA ADDRESSING

When the iSBC 208 controller is operating as a bus master and performing DMA transfers, it has the capability of addressing up to 16 megabytes (24 address bits) of system memory. The controller can address the full 1-megabyte address space as specified by the Multibus interface by using the 20 address lines provided on Multibus connector P1 and, in addition, can generate four additional address lines to select between sixteen unique 1-megabyte pages. The four additional address lines, ADR14 through ADR17, are routed onto the P2 connector. Since the DMA chip only generates a 16-bit address, circuits on the board are used to latch additional address bits and then to add these bits to the DMA address when a DMA transfer occurs. The 16-bit address from the

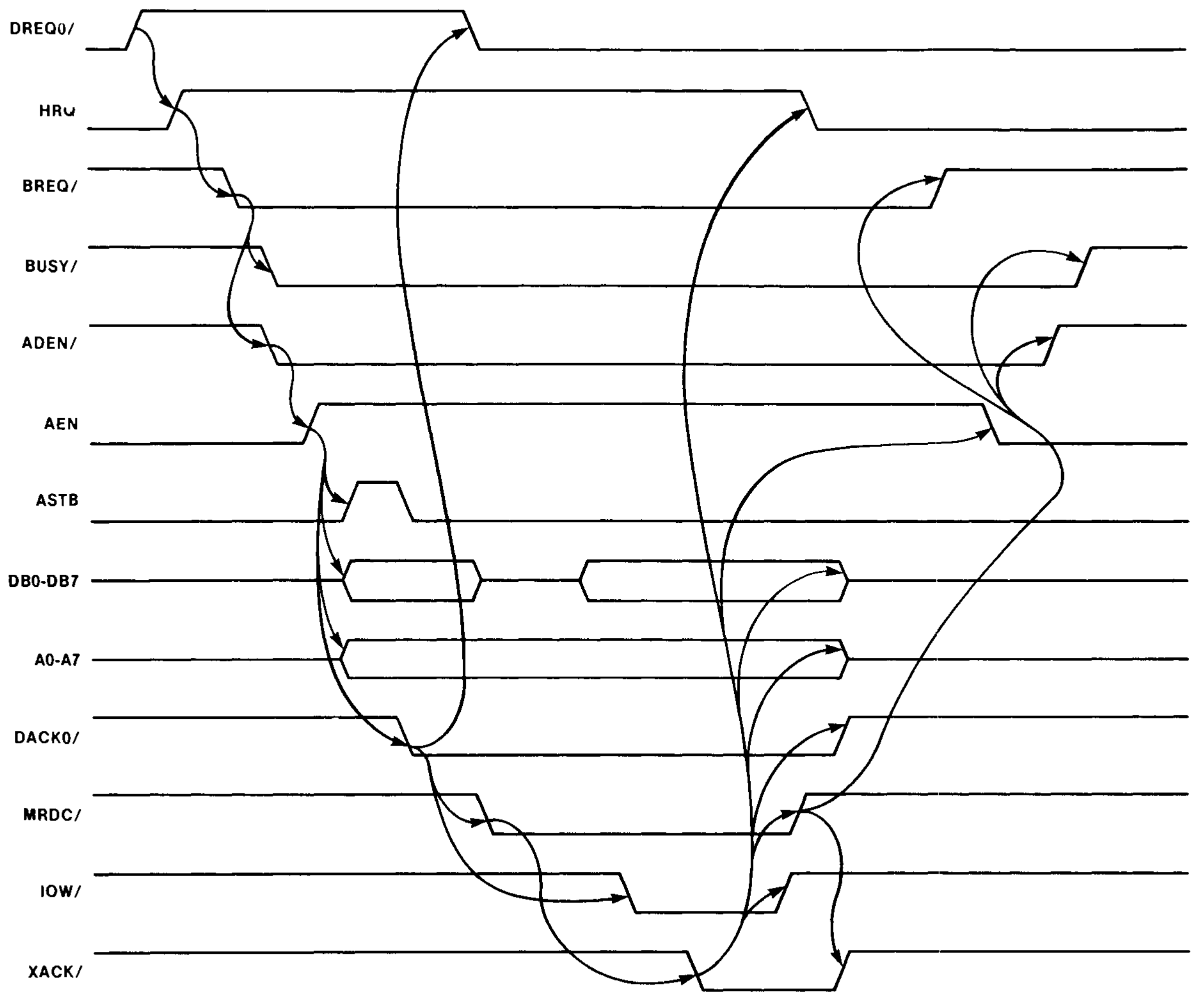


Figure 4-2. DMA Transfer Timing 143078-20

DMA chip (offset address) is added to a 16-bit segment address from the Multibus to form a 20-bit memory address. The 20-bit memory address is appended to the four high-order paging bits.

Prior to the start of a DMA transfer, segment registers U30 (5ZB6) and U32 (5ZC6) must be loaded, the starting (offset) address and word count must be loaded into the DMA chip, and then the four high-order paging bits of the memory address must be loaded into the AUX port. The low-order byte of the segment register is loaded into U32 and latched when the CS4/ (5ZC8) and IOWB/ (5ZC8) signals are active. The high-order byte of the segment register is loaded into U30 and latched when the CS5/ (5ZB8) and IOWB/ (5ZC8) signals are active. The DMA chip, in turn, loads the upper half of the offset address in offset register U31 (5ZC6) when ASTB (5ZB8) is active. The high-order paging bits (ADR 14 - ADR 17) are loaded into the AUX port (3ZB5) when CS2/ (3ZB8) and IOWB/ (3ZB6) are active. The outputs from the segment registers, the offset register, and bits DA4 through DA7 of the offset address from the DMAC are input to a set of four adders, U42 through U45 (5ZD-B4). The sum of these inputs plus offset address bits A0-A3 from the DMAC and the four high-order bits from the AUX port form the 24-bit memory address.

4-8. FLOPPY DISK CONTROLLER

Central to all disk operations is the floppy disk controller (FDC), an Intel 8272. The FDC interprets all read, write, and seek instructions and, via the floppy disk drive interface, commands the selected FDD to perform the requested operation. The FDC operates in either single-density (FM) or double-density (MFM) mode and supports an IBM sector format in both modes. Additionally, the FDC is compatible with both single-sided and double-sided media; the controller includes the circuitry for interfacing up to four drives and is capable of performing concurrent seek or recalibrate operations on four drives.

The FDC performs parallel-to-serial and serial-to-parallel data conversions; the FDC passes reassembled parallel data to the DMA controller during diskette read operations and provides serial data from the DMA controller to the selected drive during write operations. The FDC also generates an interrupt to signal the host processor that a requested operation has been completed; the FDC's status registers provide the host processor with both normal- and failure-mode status information.

4-9. FDD INTERFACE

The FDD interface consists principally of buffers, head-load and drive-select decoders, head load timers, special-function latches, and data-clock and data separation circuitry.

4-10. DRIVE AND HEAD SELECTION. Drive select decoding is performed by half of U24 (7ZB5); the other half of U24 provides radial head-load signals by decoding the drive select signal from the FDC (7ZC6). The radial head-load signals drive timers U13 (7ZC3) and U23 (7ZC3) which, in turn, provide extended HEAD LOAD signals to the drives. The timers are wired as one shots with retriggerable operation provided by transistors Q1 through Q4. Each timer produces a head-load signal for approximately one second after the FDC inactivates its HDL signal.

Half of U21 (7ZD5) provides demultiplexing for the FAULT RESET/, STEP/, LOW CURRENT/, and DIRECTION/ signals, and the other half of U21 (7ZB7) multiplexes the signals WRITE PROTECT/, TWO SIDED/, FAULT/, and TRACK 0/ signals, all under control of the RW/Seek signal from the FDC.

4-11. WRITE PRECOMPENSATION. U3 (7ZB5) is connected as a shift register and is clocked with a 4-MHz signal. This signal timing causes write data from the FDC to arrive at the shift register outputs in increments of 250 ns. Compared to the data at U3-10, the data at U3-7 is 250 ns early and the data at U3-15 is 250 ns late. The normal, early, or late data is selected by U4 (7ZB3) to provide write precompensation under control of the FDC. AND gates U14-3 and U14-6 gate the pre-shift control signals (PS0,PS1) to allow precompensation only on the inner tracks (tracks 43-77) as determined by the low current signal from U8-2 (7ZD4).

4-12. DATA SEPARATOR. The purpose of the data separator circuit is to generate a data window signal (RDWN) that enables the FDC to separate the data bits from the clock bits in the serial data stream received from the drive. The actual determination of a data "1" bit is performed by the FDC.

The data and clock pulses must be kept in their respective windows in the presence of data clock pulse jitter and frequency variations. Pulse jitter is overcome by a window-extender feature, while frequency variation is minimized by the use of a phase-lock-loop circuit. Figure 4-3 is a timing diagram of the data seperator circuit. While this circuit is designed to run at three data rates, only one rate, double density on an 8-inch drive, is used in the following description (a comparison between the three data rates is shown in table 4-4).

At the double-density, 8-inch data rate, clock and data pulses can be 2, 3, or 4 microseconds apart. This timing separation requires a 1-microsecond, on-time clock for the comparison. The on-time clock signal originates from the VCO and is generated by divider U69-5 (6ZC4) and selected by data selector U71-7 (6ZD3). Flip-flop U67-5 stays set since its K input is held low and allows the undivided on-time clock to be fed directly into U69-1. An on-time clock of 1 MHz is selected by U71.

The data from the drive (READ DATA/) is delayed and shaped by one-shot U35-7 (6ZC7) into a series of 550 ns-wide pulses and fed to the 4B input of U70 (6ZD4). Since the MINI/ signal is inactive, the 550 ns pulses are multiplexed to U71; the resultant output at U71-4 is input to the clock inputs of U53-9 (6ZA5), U54-9 (6ZB5), and U37-5 (6ZB5). The digital phase comparator consisting of U37 (6ZA5) is prevented from making false comparisons by flip-flop U53-9. The leading edge of each delayed data pulse from U71-4 clocks U53-9 to enable U37 to make a comparison between the trailing edge of the delayed data pulse at U37-3 and the rising edge of the on-time clock at U37-11.

The phase comparator controls transistors Q5 and Q6 (6ZA3) that form a current pumping circuit to increase or decrease the charge across C46. The resultant voltage on C46 controls the operating frequency of VCO U38 (6ZA2); an increase in voltage causes a corresponding increase in VCO output frequency.

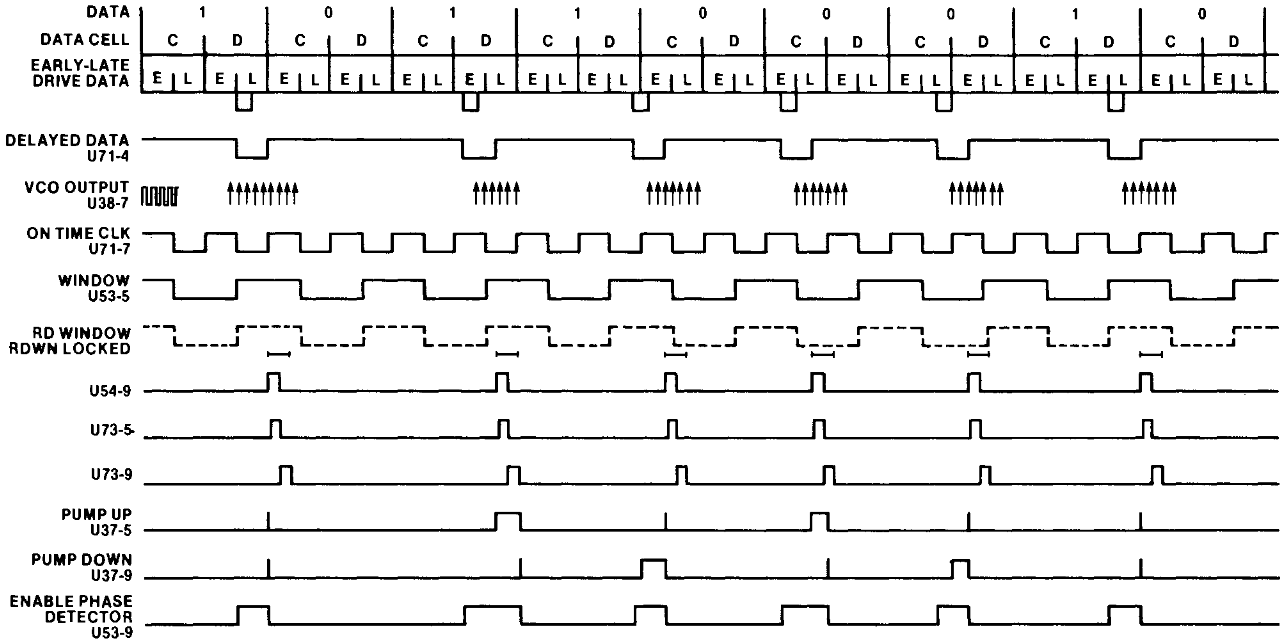


Figure 4-3. Data Recovery Timing 143078-21

When flip-flop U37-3 (6ZB5) is clocked set by the delayed data pulse before flip-flop U37-9 is clocked set by the on-time clock, Q5 conducts (increasing the charge on C46 and increasing the VCO output frequency) until U37-9 is set. When U37-9 sets, both flip-flops are cleared immediately through U52-3 and U53-9. This action creates a pump up (or pump down) time proportional to the phase difference between the delayed data pulse and the on-time clock. (A pump-down condition occurs when the on-time clock signal arrives ahead of the delayed data pulse.)

The phase comparator is enabled by the VCO signal from the FDC when reading data from the diskette. When the VCO signal is inactive, both comparator flip-flops are held set, and both pumping transistors (Q5 and Q6) conduct to cause the control voltage to return to its nominal value of 3 volts. When the control voltage is at its nominal value, the VCO (U38) output frequency is adjusted for 8 MHz by potentiometer R1 until VCO frequency indicator DS1 is at its brightest level.

Flip-flops U54-9 (6ZB5), and U73-5 (6ZB5), and U73-9 (6ZB4) form a shift register that synchronizes the delayed data to the VCO clock and that provides a 125 ns read data (RDAT) pulse for each delayed data pulse and a read window extension signal. The basic read window signal (RDWN) is generated by flip-flop U53-5 (6ZC3) on the falling edge of every other on-time clock pulse. The outputs from U53 drive slave flip-flop U55-3 and U55-11 (6ZC2). This flip-flop follows U53 unless it is inhibited by one of the shift register outputs at OR-gate U52-8 (6ZB3). The output from U52-8 is active whenever there is a pulse in the first cell (U74-9) of the shift register or whenever a data pulse occurs near the end of the normal window (i.e., whenever U73-7 is active).

Table 4-4. On Time Clock Versus Data Rate

Data Rate (k Bits/s)	Drive Size	Encoding Mode	Bit-to-Bit Spacing (μs)	On-Time Clock (MHz)
500	8-inch	MFM	2,3,4	1.0
250	8-inch	FM	2,4	0.5
250	5¼-inch	MFM	4,6,8	0.5
125	5¼-inch	FM	4,8	0.25

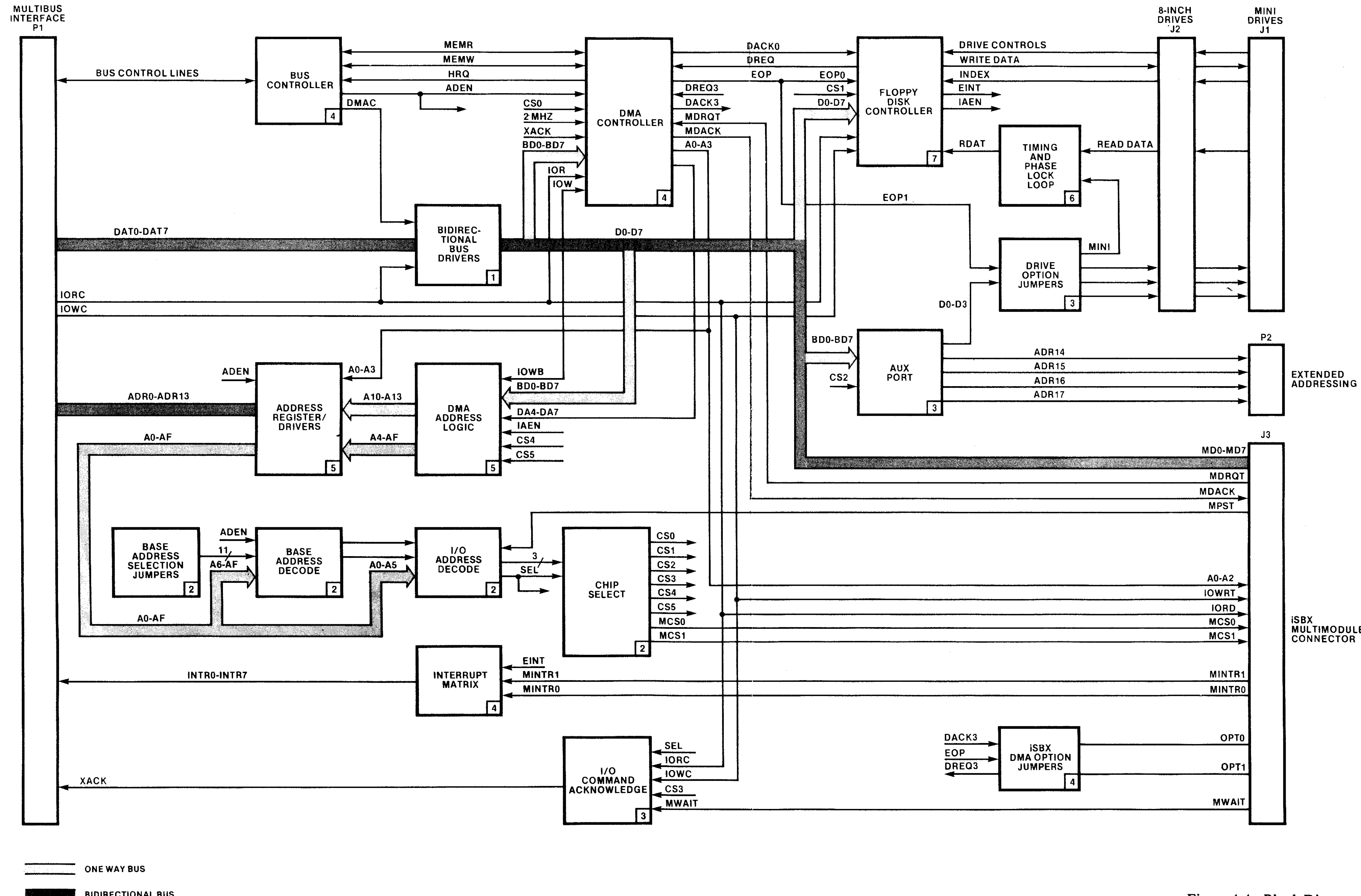


Figure 4-4. Block Diagram of Controller

CHAPTER 5
SERVICE INFORMATION

5-1. INTRODUCTION

This chapter provides the service information required for the iSBC 208 Floppy Disk Controller Board and includes a list of replaceable parts, the service diagrams, and service and repair assistance instructions.

5-2. 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. Purchase order number for billing purposes if your Intel product warranty has expired.

f. Extended warranty agreement information, if applicable.

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

Telephone:

From Alaska, Arizona, or Hawaii call—
(602) 869-4600

From all other U.S. locations call toll free—
(800) 528-0595

TWX: 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 that 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 the product in a reinforced corrugated shipping carton and label "FRAGILE" to ensure careful handling. Ship only to the address specified by Product Service Hotline personnel.

5-3. REPLACEABLE PARTS

Table 5-1 provides a list of replaceable parts for the iSBC 208 controller. Table 5-2 identifies the manufacturers specified in the MFR CODE column in table 5-1. Intel 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.

5-4. ADJUSTMENTS

The controller includes only one adjustable component, a potentiometer that sets the center frequency of the voltage-controlled oscillator (VCO). The adjustment is performed at the factory and normally is valid for the life of the controller. However, if a component is replaced within the phase-lock-loop circuit, it may be necessary to readjust the center frequency of the VCO. To perform this adjustment, insert the controller into the system, apply power, and allow the controller to "idle." While observing LED indicator DS1 (located on the front edge of the pc board towards the left), adjust potentiometer R1 (adjacent to DS1) until the LED is at its maximum brightness.

5-5. SERVICE DIAGRAMS

The parts location diagram and schematic diagram for the iSBC 208 controller are provided in figures 5-1 and 5-2, respectively. The parts location diagram is useful in locating the parts listed in table 5-1.

The schematic diagram (figure 5-2) consists of seven sheets of logic drawings that were current when the manual was printed. Minor revisions and changes may have occurred since the manual was printed. If a discrepancy exists between the schematic in the manual and the schematic shipped with the controller, the schematic shipped with the controller always supersedes the schematic in the manual.

A signal on the schematic diagram that traverses from one sheet of the drawing to another is labeled with the same boxed letter reference (e.g., A) on each sheet to simplify signal tracing. The signal mnemonic and the source/destination sheet number are shown adjacent to the boxed letter reference. Generally, signal mnemonics listed on the left side of a sheet are entering the diagram, and signal mnemonics listed on the right side of a sheet are leaving the diagram. On the schematic diagram, a signal mnemonic that ends with a slash (e.g., ALE/) is active low. Conversely, a signal mnemonic without a slash (e.g., ALE) is active high.

Table 5-1. Replaceable Parts

Reference	Description	Mfr. Part No.	Mfr. Code	Qty.
U1	IC, Quad 2-input positive-NOR gate	7402	TI	1
U2,U19, U51,U61	IC, Quad bus buffer gates	74125	TI	4
U3	IC, Quad D-type flip-flops	74175	TI	1
U4	IC, Dual 4-to-1 data selector/mux	74153	TI	1
U5,U6,U9, U10,U12	IC, Quad 2-input positive-NAND buffer	7438	TI	5
U7,U34,	Not Used			
U8,U11,U58	IC, Hex Schmidt trigger inverters	7414	TI	3
U13,U23	IC, Dual timer	NE556	SIG	2
U14	IC, Quad 2-input positive-AND gate	7408	TI	1
U15,U48	IC, Hex inverters	7404	TI	2
U16	IC, PROM, 1024x4	3625	Intel	1
U17	IC, Floppy disk controller	8272	Intel	1
U18	IC, Programmable DMA controller	8237A	Intel	1
U20	IC, Quad 2-input positive-NOR gate	74S02	TI	1
U21,U57,U72	IC, Octal 3-state buffers	74LS240	TI	3
U22,U37,U54	IC, Dual D-type flip-flop	74S74	TI	3
U24	IC, Dual 2-to-4 decoder/multiplexer	74LS139	TI	1
U25	IC, Quad 2-input positive-OR gate	7432	TI	1
U26	IC, Quad 2-input positive-OR gate	74S32	TI	1
U27	IC, 3-to-8 line decoder/mux	3205	Intel	1
U28,U40,U41	IC, 4-bit magnitude comparator	74LS85	TI	3
U29,U30,U32	IC, Octal D-type flip-flops	74LS273	TI	3
U31	IC, Octal D-type latches	74LS373	TI	1
U33,U46	IC, Octal buffer/line driver/receiver	74LS244	TI	2
U35,U36	IC, Dual retriggerable multivibrator	9602	SIG	2
U38	IC, Dual voltage controller oscillator	74124	TI	1
U39	IC, Bus controller	8218	Intel	1
U42,U43 U44,U45	IC, 4-bit binary full adder	74LS283	TI	4
U47	IC, Triple 3-input positive-NAND gates	74S10	TI	1
U49,U50	IC, Serial shift register	74LS164	TI	2
U52,U55	IC, Quad 2-input positive-NAND gate	7400	TI	4
U53,U67,U73	Ic, Dual J/K flip-flops	74S112	TI	3
U56	IC, Hex bus drivers	74368	TI	1
U59	IC, Quad bus buffer gates	74LS125	TI	1
U60	IC, Quad 2-input positive-NAND gate	74S38	TI	1
U62,U63,U64 U65,U66	IC, Octal bus transceiver	8287	Intel	3
U69	IC, Dual 4-bit binary counter	74LS393	TI	1
U70,U71	IC, Quad 2-to-1 data selector/mux	74157	TI	2

Table 5-1. Replaceable Parts (Cont'd.)

Reference	Description	Mfr. Part No.	Mfr. Code	Qty.
R2,R3,R6,R7, R12,R15,R18, R20,R29,R30, R39,R40	Resistor, 4.7k ohm, 1/4W, 5*	OBD	COML	12
R4,R38	Resistor, 330 ohm, 1/4W, 5%	OBD	COML	2
R5	Resistor, 220 ohm, 1/4W, 5%	OBD	COML	1
R8	Resistor, 150 ohm, 1/4W, 5%	OBD	COML	1
R9,R10,R16,R17	Resistor, 1M ohm, 1/4W 5%	OBD	COML	3
R11,R25,R26, R27,R28	Resistor, 270 ohm, 1/4W, 5%	OBD	COML	5
R13,R14	Resistor, 33k ohm, 1/4W, 5%	OBD	COML	2
R19,R34,R41	Resistor, 1.5k ohm, 1/4W, 5%	OBD	COML	3
R21	Resistor, 4.99k ohm, 1/8W, 1%	OBD	COML	1
R22	Resistor, carbon, 8.2k, 1/4W, 5%	OBD	COML	1
R23	Resistor, 10k ohm, 1/8W, 1%	OBD	COML	1
R24	Resistor, 20k ohm, 1/8W, 1%	OBD	COML	1
R31	Resistor, 2.2k ohm, 1/4W, 5%	OBD	COML	3
R32,R35,R36	Resistor, 1k ohm, 1/4W, 5%	OBD	COML	3
R33	Resistor, 750 ohm, 1/4W 5%	OBD	COML	1
R37	Resistor, 1.3k ohm, 1/4W, 5%	OBD	COML	1
R42	Resistor, 470 ohm, 1/4W 5%	OBD	COML	1
R43,R44,R45, R46,R47	Resistor, 10k ohm, 1/4W, 5%	OBD	COML	5
RP1	Resistor pack, 220/330 ohm, SIP	4308R-103-331/221	BOUR	1
RP2,RP3	Resistor pack, 10k ohm, DIP	4114R-002-103	BOUR	2
RP4	Resistor pack, 5.6k ohm, DIP	4114R-002-562S	BOUR	1
R1	Resistor, variable, 1k, 0.5W	68XR1K	BECK	1
C66	Thermistor, 8k, 10%	IM8001-5	WEST	1
C1,C3,C5,C7,C10, C12,C16,C17,C18, C19,C21,C23,C25, C27,C28,C29,C30, C32,C34,C39,C43, C44,C47,C48,C50, C52,C54,C56,C59, C60,C62,C64,C65, C71,C73,C74,C76, C79,C80,C82,C84	Capacitor, 0.1μF, 50V, +80-20%	OBD	COML	41
C2,C4,C6,C8, C9,C11,C13,C20, C26,C31,C33,C35, C36,C37,C38,C47, C49,C51,C53,C55, C57,C58,C61,C63, C67,C70,C72,C75, C78,C81,C83	Not Used			
C14,C15,C22,C24	Capacitor, 1μF, 20V, 10%	OBD	COML	4
C34	Capacitor, 1000pF, 15V, 5%	OBD	COML	1
C40,C41,C42	Capacitor, 270pF, 100V, 5%	OBD	COML	1
C45	Capacitor, 50pF, 100V, 5%	OBD	COML	1
C46	Capacitor, 0.01μF, 50V, 20%	OBD	COML	1
C68,C77	Capacitor, 22μF, 15V, 10%	OBD	COML	1
C69	Capacitor, 390pF, 100V, 5%	OBD	COML	1
L1	Inductor, 100μH, 10%	9230-8	JWM	1
Q1,Q2,Q3,Q4,Q5	Transistor, PNP, 2N2907A	2N2907A	FAIR	5
Q6	Transistor, NPN, 2N2222A	2N2222A	FAIR	1

Table 5-1. Replaceable Parts (Cont'd.)

Reference	Description	Mfr. Part No.	Mfr. Code	Qty.
XU17,XU18	Socket, IC, 40-pin	540-AG11D	AUG	2
G1	Crystal, clock oscillator, 8MHz	208-CB	WAK	1
DS1	Light emitting diode, red	HLMP-0301	HEW	1
E1 through E89,TP1,TP2	Wirewrap posts, interconnect	87623-1	AMP	91
J1	Connector, header, 34-pin	3433-1302	MMM	1
J2	Connector, header, 50-pin	3431-1302	MMM	1
J3	Connector, Multimodule, 18/36-pin	000291-001	VIK	1
	Connector, HSG, receptacle, 2-pin	530153-2	AMP	12

Table 5-2. Manufacturer's Codes

Mfr. Code	Manufacturer	Address
AMP	AMP, Inc.	Harrisburg, PA
AUG	Augat, Inc.	Attelboro, MA
BECK	Beckman Instruments, Inc.	Cedar Grove, NJ
BOUR	Bourns, Inc.	Riverside, CA
FAIR	Fairchild Semiconductor	Santa Clara, CA
HEW	Hewlett Packard	Cupertino, CA
Intel	Intel	Santa Clara, CA
JWM	J. W. Miller Division Bell	Compton, CA
MMM	Minnesota Mining Manufacturing	Minneapolis, MN
SIG	Signetics Semiconductor	Santa Clara, CA
TI	Texas Instruments	Dallas, TX
VIK	Viking Connector, Inc.	San Diego, CA
WAK	Wakefield Engineering, Inc.	Wakefield, MA
WEST	Western Thermistor Corp.	Oceanside, CA
COML	Any commercial source; Order By Description (OBD)	

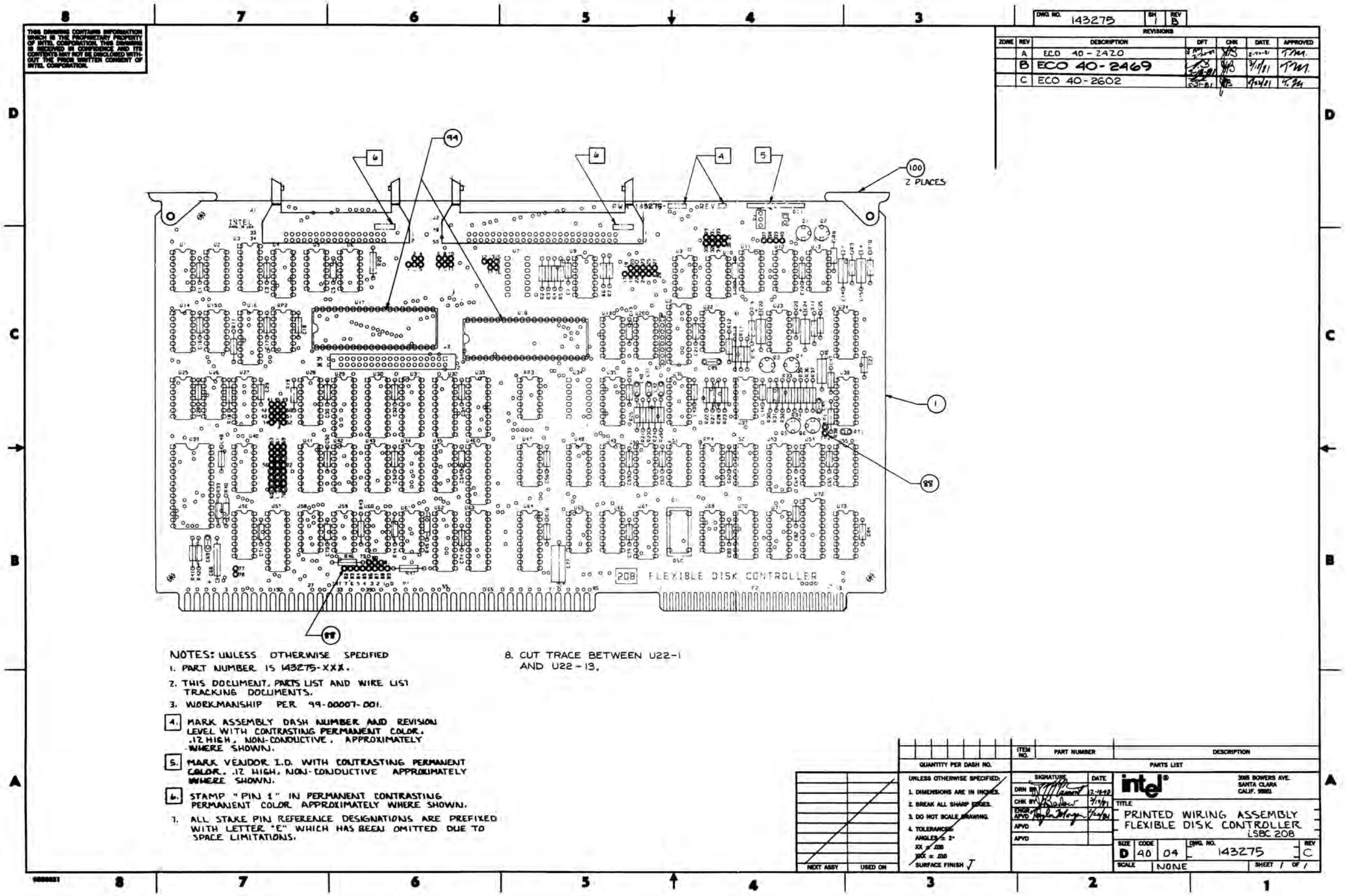
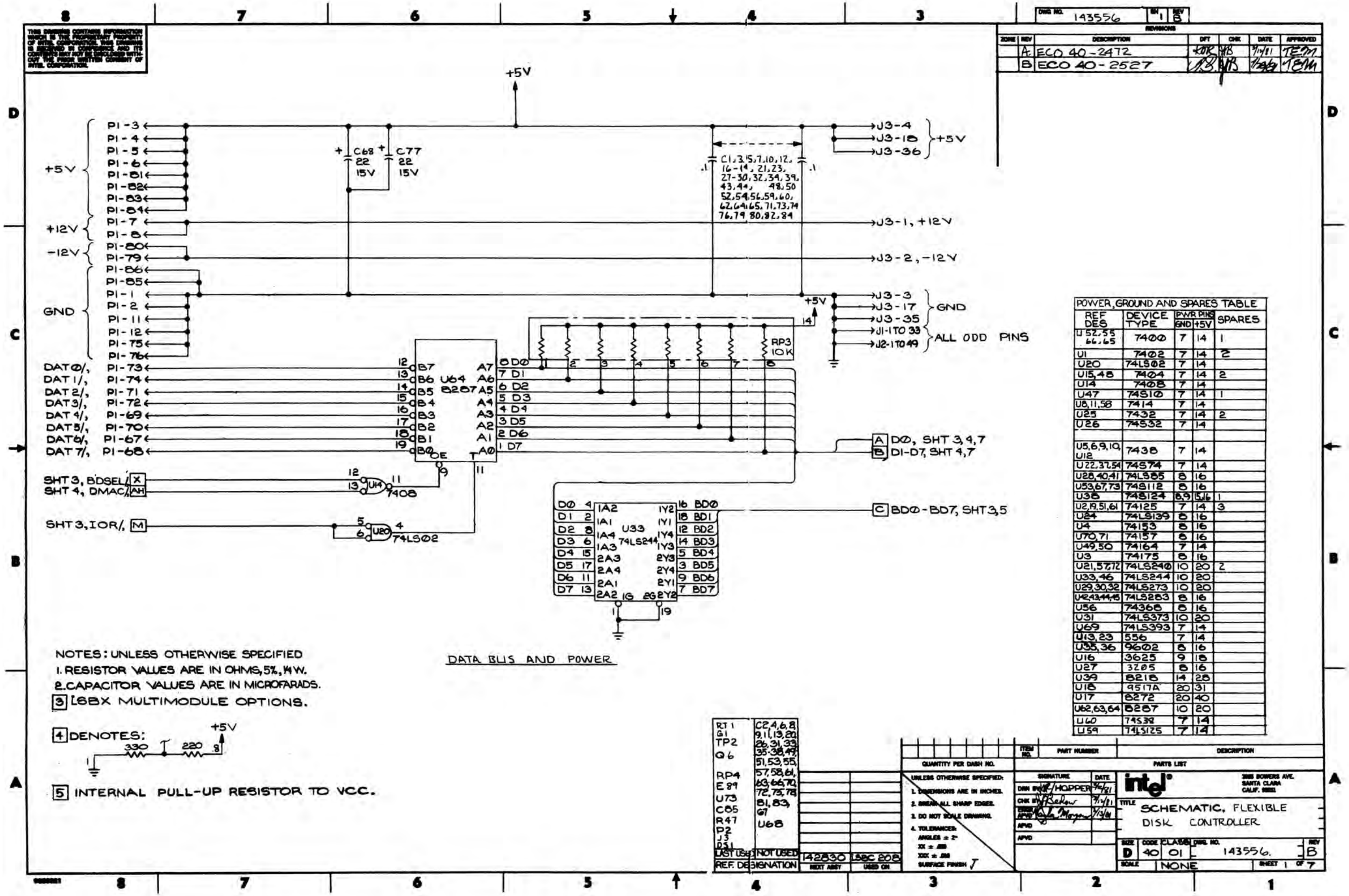


Figure 5-1. iSBC™ 208 Parts Location Diagram



POWER, GROUND AND SPARES TABLE				
REF DES	DEVICE TYPE	PWR PINS GND	PWR PINS +5V	SPARES
U52,55 66,65	7400	7	14	1
U1	7402	7	14	2
U20	74LS02	7	14	
U15,48	7404	7	14	2
U14	7408	7	14	
U47	74S10	7	14	1
U8,11,58	7414	7	14	
U25	7432	7	14	2
U26	74S32	7	14	
U5,6,9,10, U12	7438	7	14	
U22,37,54	74S74	7	14	
U28,40,41	74LS85	8	16	
U53,67,73	74S112	8	16	
U38	74S124	8,9	15,16	1
U2,19,51,61	74125	7	14	3
U24	74LS139	8	16	
U4	74153	8	16	
U70,71	74157	8	16	
U49,50	74164	7	14	
U3	74175	8	16	
U21,57,72	74LS240	10	20	2
U33,46	74LS244	10	20	
U29,30,32	74LS273	10	20	
U42,43,44,45	74LS283	8	16	
U56	74368	8	16	
U31	74LS373	10	20	
U69	74LS393	7	14	
U13,23	556	7	14	
U35,36	9602	8	16	
U16	3625	9	18	
U27	3205	8	16	
U39	8218	14	28	
U18	9517A	20	31	
U17	8272	20	40	
U62,63,64	8287	10	20	
U60	74S38	7	14	
U59	74LS125	7	14	

Figure 5-2. iSBC™ 208 Board Schematic Drawing (Sheet 1 of 7)

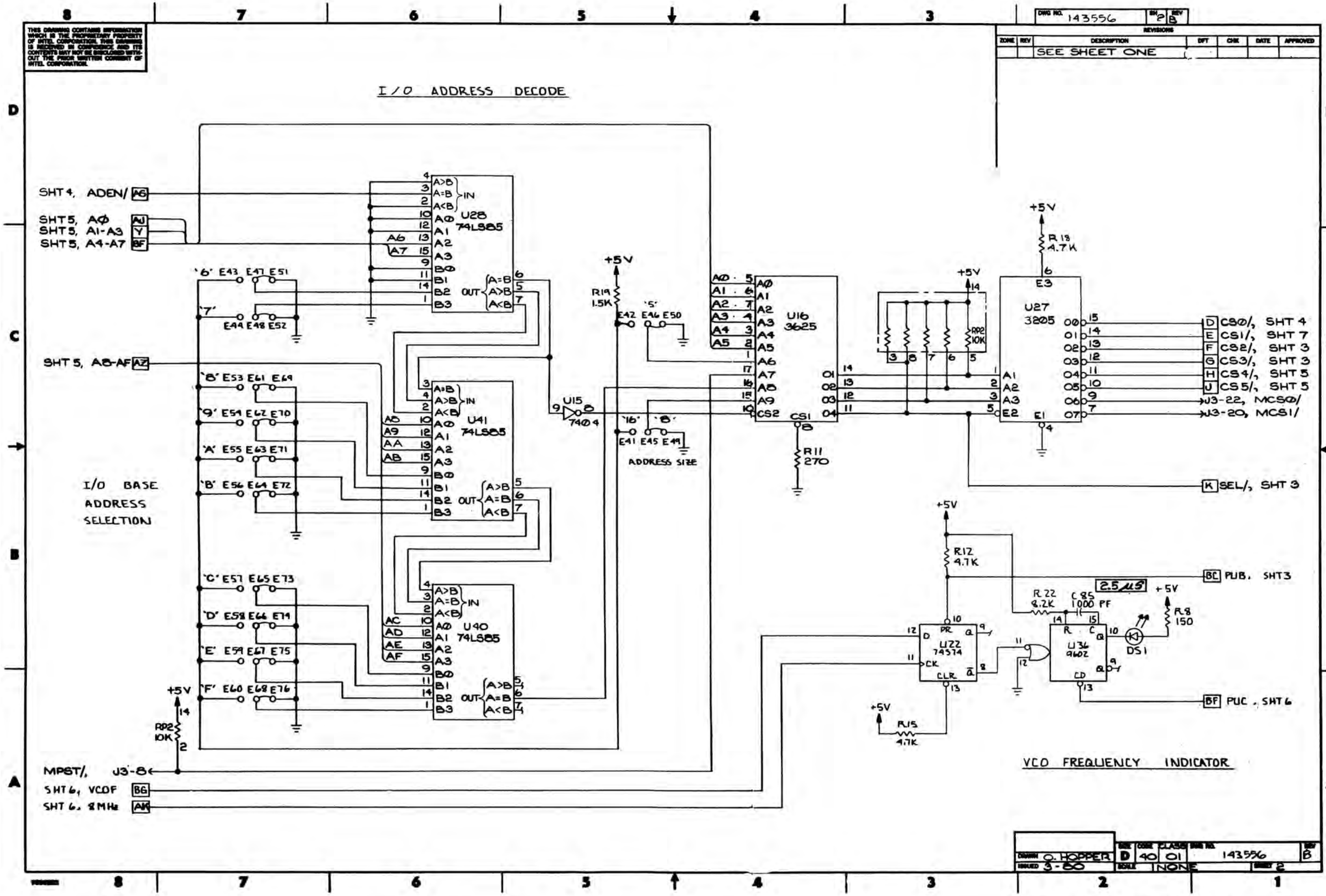


Figure 5-2. iSBC™ 208 Board Schematic Drawing (Sheet 2 of 7)

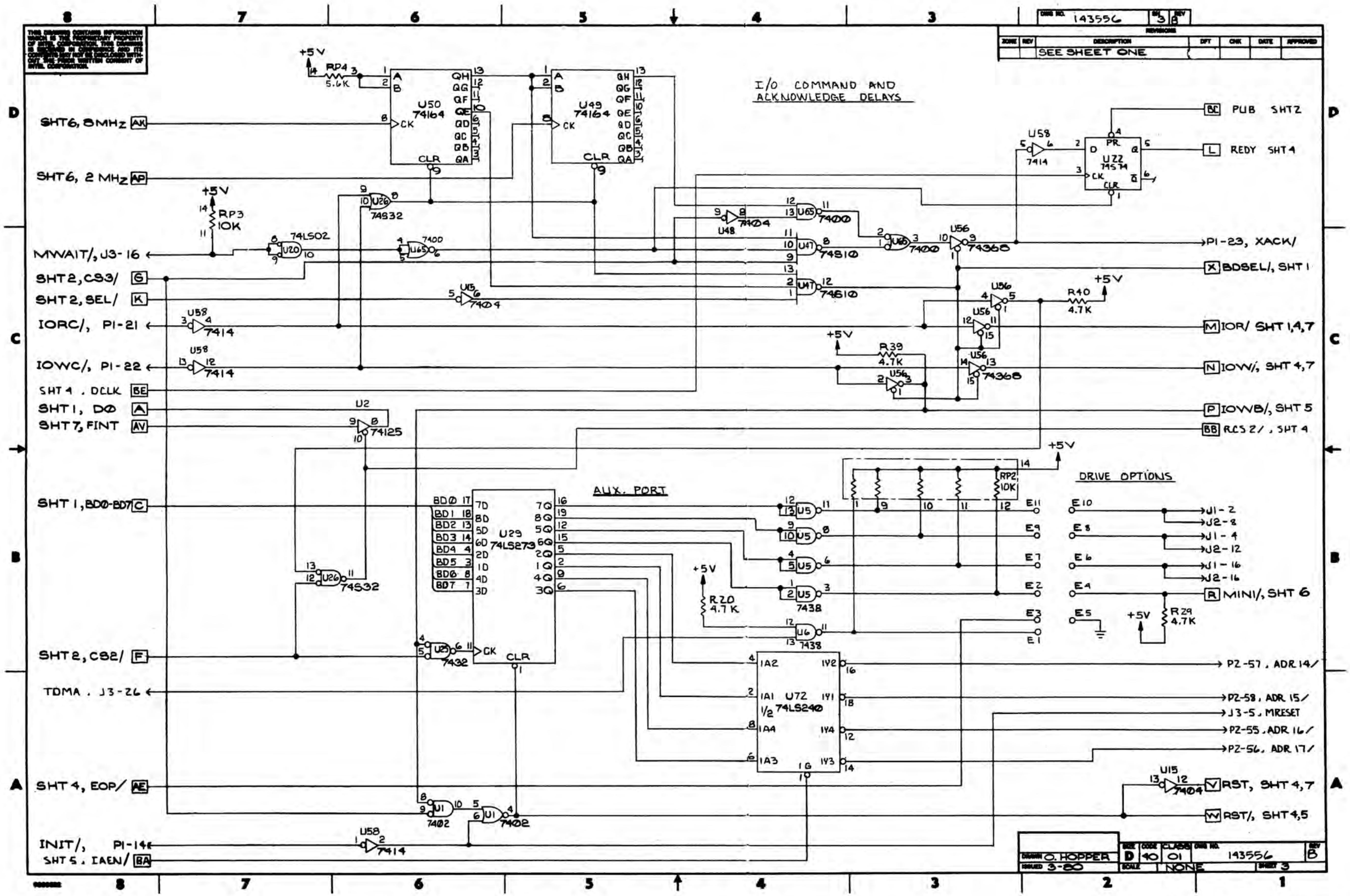


Figure 5-2. iSBC™ 208 Board Schematic Drawing (Sheet 3 of 7)

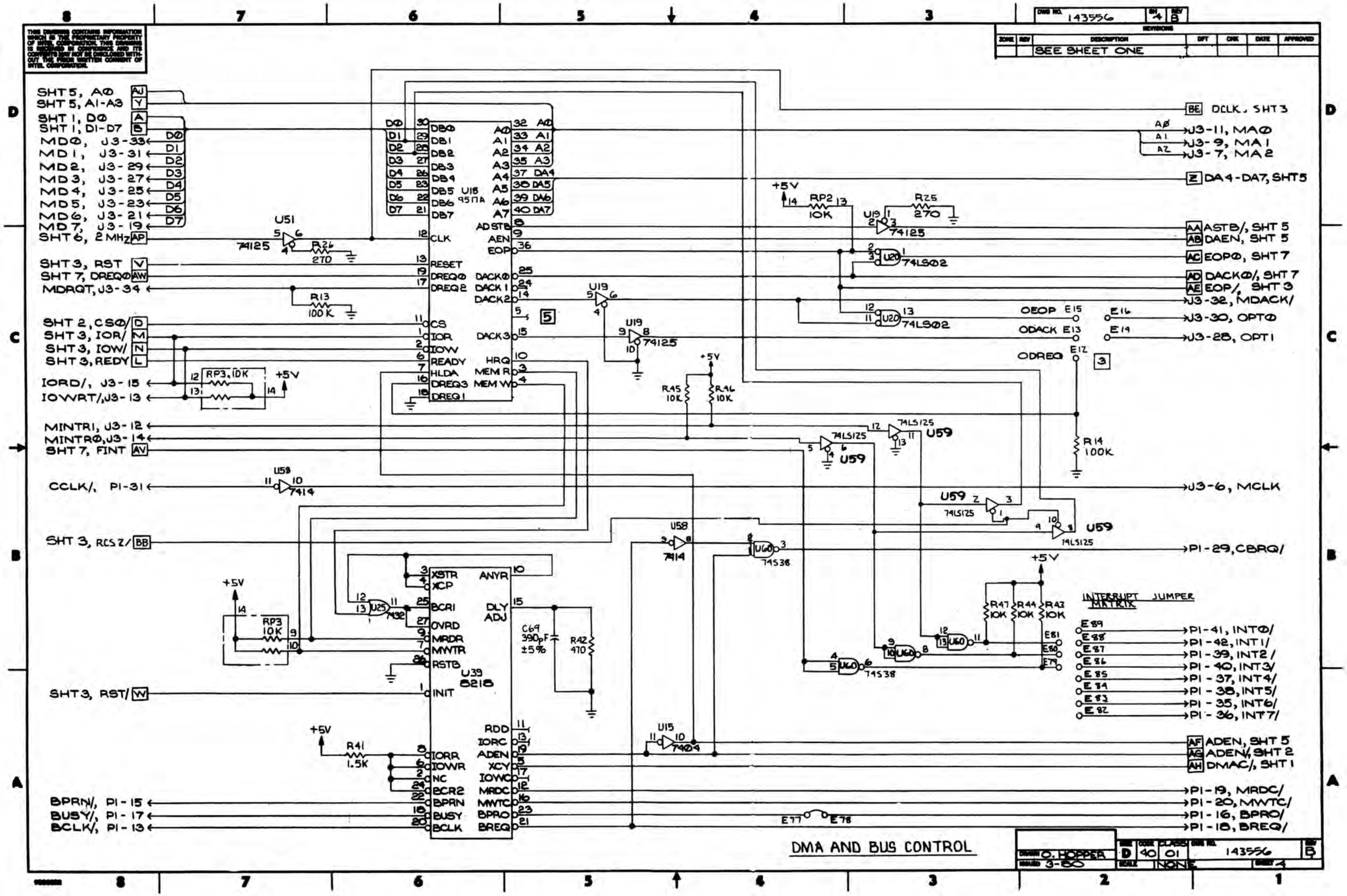


Figure 5-2. iSBC™ 208 Board Schematic Drawing (Sheet 4 of 7)

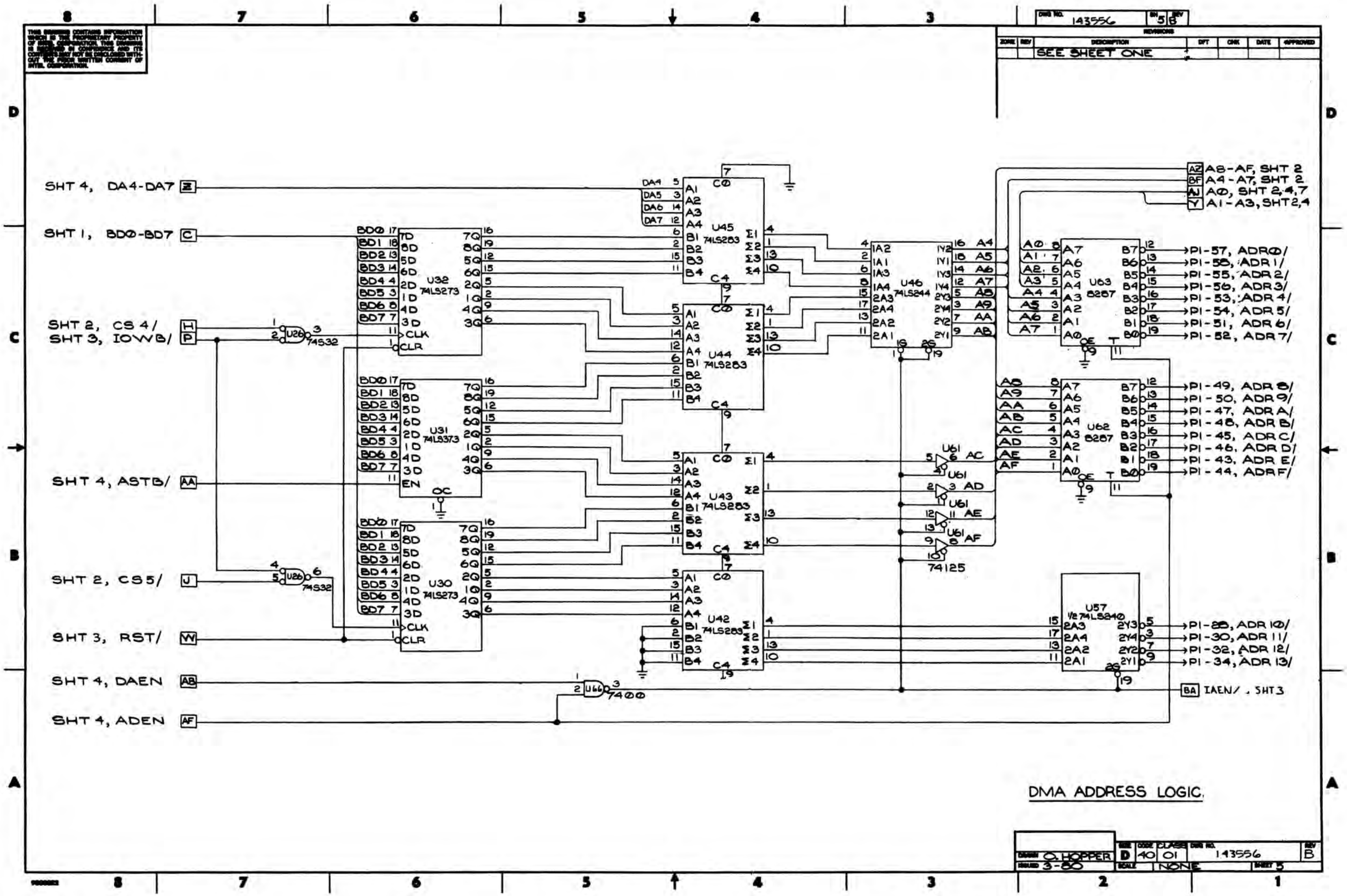


Figure 5-2. iSBC™ 208 Board Schematic Drawing (Sheet 5 of 7)

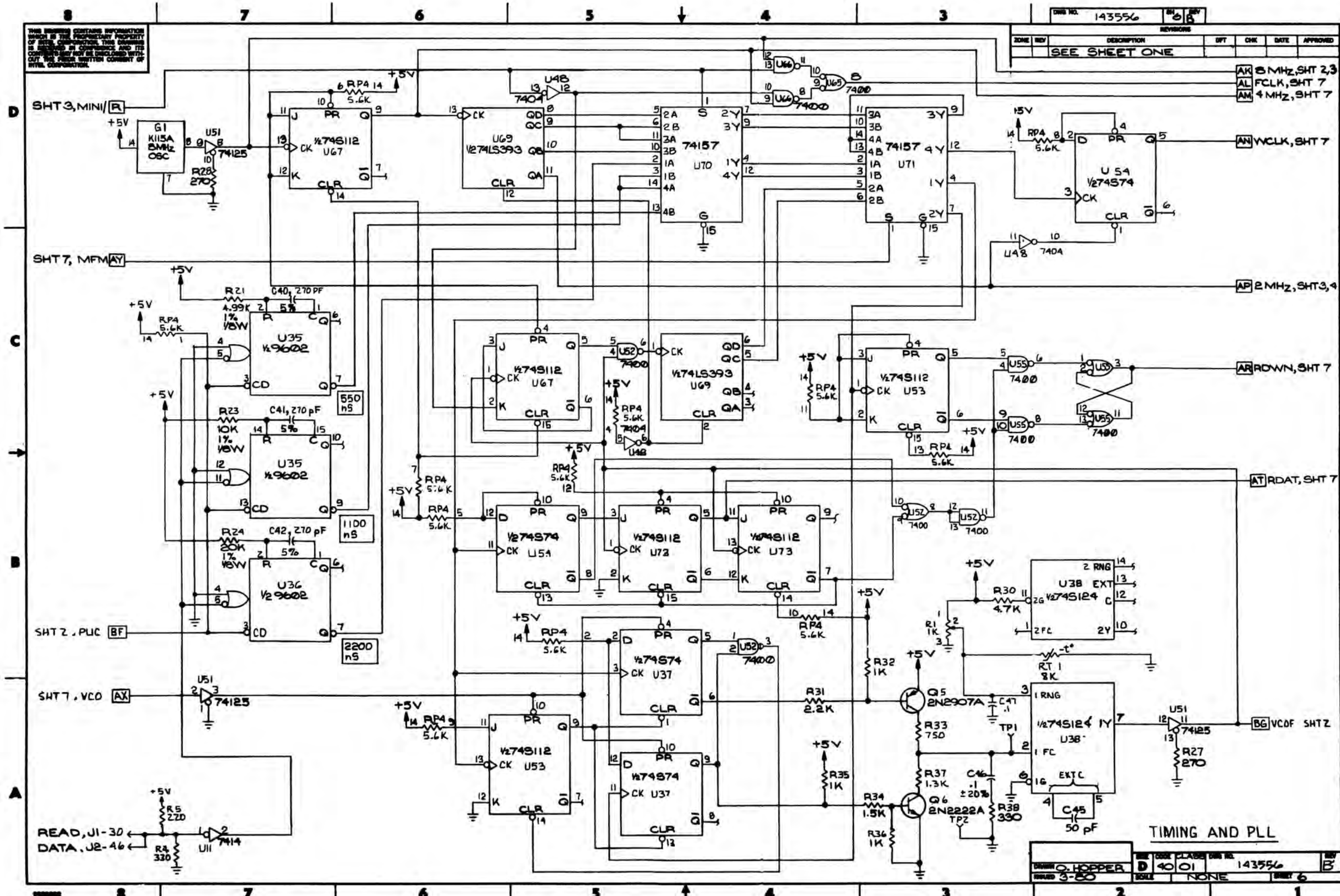


Figure 5-2. iSBC™ 208 Board Schematic Drawing (Sheet 6 of 7)

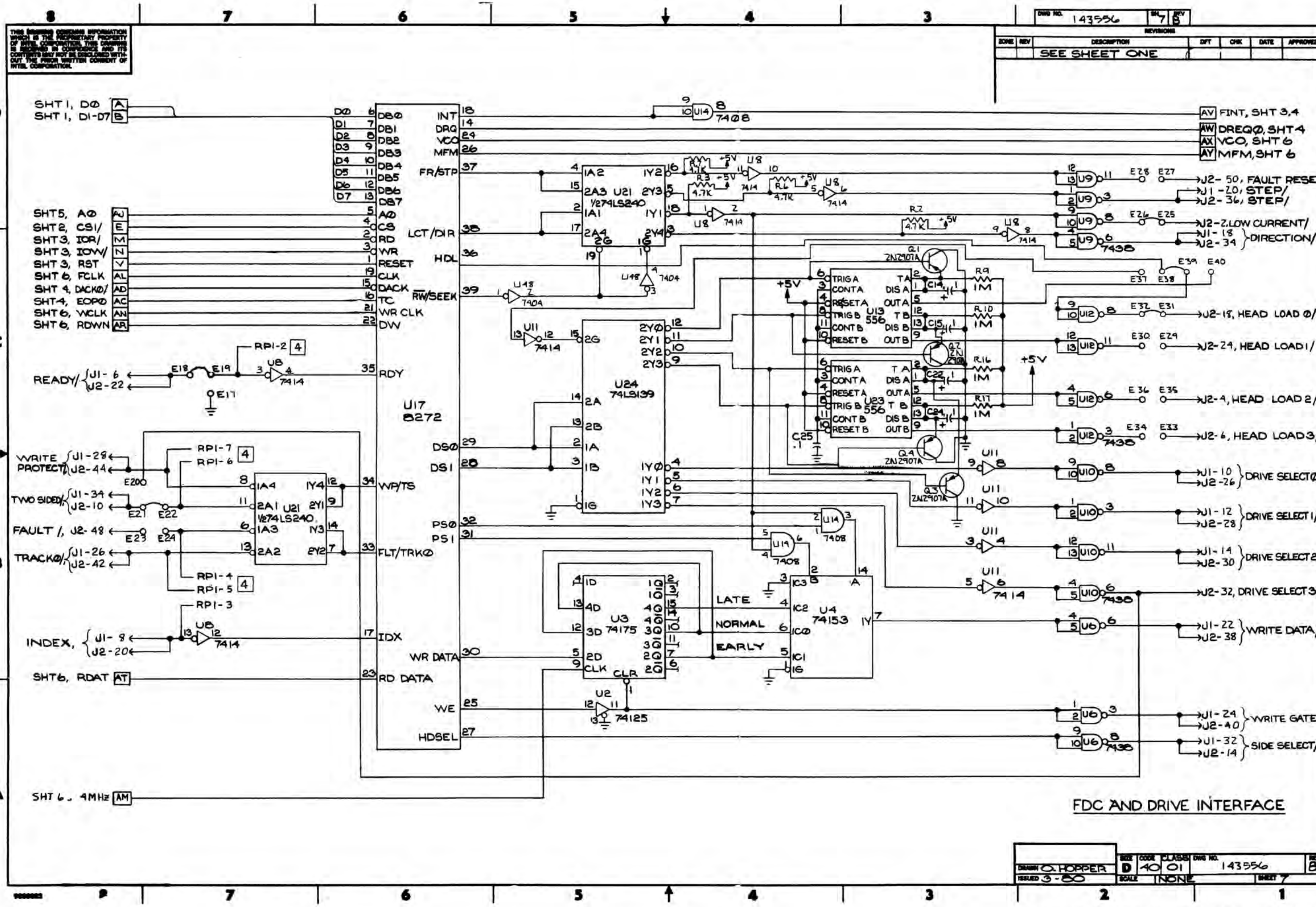


Figure 5-2. iSBC™ 208 Board Schematic Drawing (Sheet 7 of 7)

APPENDIX A
SAMPLE DRIVERS

A-1. INTRODUCTION

This appendix provides two sample drivers fcr the iSBC 208 Controller. Section A-2 describes a PL/M-86 driver for 16-bit systems. Section A-3 provides an assembly language listing for 8-bit systems.

A-2. PL/M-86 DRIVER

This is an instruction decode and driver for the iSBC 208 Disk Controller. The procedure is called from the main routine with no parameters. All of the instructions and necessary parameters are passed in an IOPB as defined below in the structure declarations. There are 12 possible instructions for the controller.

Some of the instructions require programming of the DMA controller chip on the board. The driver determines this need and either programs the chip or skips the section altogether.

The IOPB is filled with the required information for a particular instruction by the calling program. Only those locations needed for the instruction are filled, the other locations are don't cares. The IOPB is structured as follows:

15 — 8	7 — 0	
track number	instruction #	Word 1
sector number	head and drive	Word 2
mt-mf-sk byte	sectors/track	Word 3
bytes/sector	gap-3 length	Word 4
number of bytes to be transferred		Word 5
segment - memory location		Word 6
offset - memory location		Word 7

Since the IOPB is a public data structure, the address need not be passed by the calling routine.

The driver makes some assumptions and performs no error checking on the data in the IOPB. Care must be taken by the calling routine to ensure correct data for the instruction being requested.

The instructions available and their instruction numbers are as follows:

INSTRUCTION #	INSTRUCTION
1	Read Data
2	Write Data
3	Write Deleted Data
4	Read Deleted Data
5	Read a Track
6	**NOT USED***
7	**NOT USED***
8	**NOT USED***
9	Format a Track
10	Read ID
11	Recalibrate
12	Sense Interrupt Status
13	Specify
14	Sense Drive Status
15	Seek

For any instruction, the calling program is responsible for placing the correct values necessary for the instruction in the IOPB.

Status is returned to the calling procedure via a STATUS structure that also is declared public. The status is pulled directly from the FDC on completion of the instruction and is only reported; no action is taken on the status values. The PASS_FAIL byte is set to pass if status is received properly.

The STATUS structure is as follows:

15 8	7 0	
status byte 0	status byte 1	Word 1
status byte 2	track number	Word 2
head address	sector number	Word 3
pass-fail byte	N byte	Word 4

```
execute$module:
DO;

/*  This structure is used as an IOPB device to pass information through-
    out the software regarding the current instruction to be executed.
    The structure is filled with information by the calling program and
    is taken apart by the 8272 drivers.  It is declared public. */

 DECLARE iopb STRUCTURE(instruction BYTE,
                        track$no    BYTE,
                        head$drive  BYTE,
                        sector$no   BYTE,
                        mt$mf$sk    BYTE,
                        sectors$per BYTE,
                        n           BYTE,
                        gap$3       BYTE,
                        byte$cnt    WORD,
                        buffer      POINTER) PUBLIC;

/*  This structure is used to return status information from the 8272
    drivers to the calling program.  It also is declared public. */

    DECLARE status STRUCTURE(zero        BYTE,
                             one         BYTE,
                             two         BYTE,
                             track$no    BYTE,
                             head$addr   BYTE,
                             sector$no   BYTE,
                             pass$fail   BYTE,
                             n           BYTE)  PUBLIC;

/*  The following are variables and literals used by the program */

    DECLARE base$addr WORD PUBLIC;  /* Base address of iSBC 208 */
    DELCARE (temp1,temp2,temp3) BYTE ; /* Temporary byte variables */
    DELCARE (segval,offsetval) WORD PUBLIC; /* Used to split pointers */
    DELCARE port LITERALLY 'base$addr + 11'; /* data I/O port on the iSBC 208*/

/* literal declarations */
DELCARE fail LITERALLY '0ffh';
DELCARE pass LITERALLY '0';
DELCARE failure LITERALLY 'status.pass$fail=fail; return ';

/* external and public procedure declarations */
DELCARE (retry,retry$number) BYTE; /* used for soft error retrys */

/*  SPLIT is used to get around a limitation of PLM/86.  The routine is
    called with a pointer parameter and returns two words in the variables
    SEGVAL and OFFSETVAL which are the segment and offset derived from the
    pointer.  The routine can be found right after this listing. */

split:
  PROCEDURE (buff$ptr) EXTERNAL;
  DECLARE buff$ptr POINTER;
END split;
```

```
/*  The following two routines are used to delay the programming of the
    FDC between bytes to prevent overrunning the part.  This delay is
    required by the 8272.  Wait$8272 is used for writing out to the FDC
    and WAIT1$8272 is used to read status from the FDC, as they use
    different bit sequences.  */

wait$8272:
 PROCEDURE PUBLIC;
 DECLARE (pp,i) BYTE;

 /* initialize counters */
 i = 0; pp = 0;

/* no delay. The FDC insists on this delay before reading busy bits */
 DO WHILE pp<20; pp = pp+1; END;

   /* we have delayed enough, now loop on the busy status bits */
    DO WHILE ((INPUT(base$addr+10) AND 0c0h <> 80h);

    /* we do not want to hang up, so only attempt this 100 times */
     i = i+1; IF i = 100 THEN RETURN;
 END;
 RETURN;
END wait$8272;

wait1$8272:
 PROCEDURE PUBLIC;
 DECLARE (pp,i) BYTE;

 /* initialize counters */
 i = 0; pp = 0;

 /* set up a delay */
 DO WHILE pp<20; pp = pp+1; END;

 /* now set up a loop to read the busy status */
 DO WHILE ((INPUT(base$addr+10) AND 0c0h) <> 0c0h);
  i = i+1; IF i = 100 THEN RETURN;
 END;
 RETURN;
END wait1$8272;
```

```
/ ***********************************************************************************

 read$data:
    This module contains the code to program the 8272 for a read data
    instruction. It accepts one parameter which determines if the
    instuction is for read data or read deleted data.

***********************************************************************************/

read$data:
 PROCEDURE(del) PUBLIC;
 DECLARE del BYTE;

/* determine the type of read and set up the parameters */
 IF del=0 THEN temp2=06h; ELSE IF del=1 THEN temp2=0ch; ELSE temp2=02h;

 /* byte 1 */
 OUTPUT(port)=shl(iopb.mt$mf$sk,5) OR temp2;
 CALL wait$8272;

 /* byte 2 */
 temp1=(shr((iopb.head$drive AND 10h,2) + (iopb.head$drive AND 03h));
 CALL wait$8272;
 OUTPUT(port)=temp1;

 /* byte 3 */
 CALL wait$8272;
 OUTPUT(port)=iopb.track$no;

 /* byte 4 */
 CALL wait$8272;
 OUTPUT(port)=shr((iopb.head$drive AND 10h),4);

 /* byte 5 */
 CALL wait$8272;
 OUTPUT(port)=iopb.sector$no;

 /* byte 6 */
 CALL wait$8272;
 OUTPUT(port)=iopb.n;

 /* byte 7 */
 CALL wait$8272;
 OUTPUT(port)=iopb.sectors$per;

 /* byte 8 */
 CALL wait$8272;
 OUTPUT(port)=iopb.gap$3;

 /* byte 9 */
 CALL wait$8272;
 OUTPUT(port)=0ffh;

 RETURN;
 END read$data;
```

```
/ *************************************************************************

 WRITE$DATA:
    This module contains the code to program the 8272 for a write data
    instruction. It accepts one parameter which determines if the
    instuction is for write data or write deleted data.

*************************************************************************/

write$data:
 PROCEDURE(del) PUBLIC;
 DECLARE del BYTE;

/* determine if write data or deleted data */
 IF del=0 THEN temp2=05h; ELSE temp2=09h;

 /*byte 1 */
 CALL wait$8272;
 OUTPUT(port)=shl(iopb.mt$mf$sk,5) OR temp2;
 CALL wait$8272;

 /*byte 2 */
 temp1=(shr((iopb.head$drive AND 10h),2) + (iopb.head$drive AND 03h));
 CALL wait$8272;
 OUTPUT(port)=temp1;

 /*byte 3 */
 CALL wait$8272;
 OUTPUT(port)=iopb.track$no;

 /*byte 4 */
 CALL wait$8272;
 OUTPUT(port)=shr((iopb.head$drive AND 10h),4);

 /*byte 5 */
 CALL wait$8272;
 OUTPUT(port)=iopb.sector$no;

 /*byte 6 */
 CALL wait$8272;
 OUTPUT(port)=iopb.n;

 /*byte 7 */
 CALL wait$8272;
 OUTPUT(port)=iopb.sectors$per;

 /*byte 8 */
 CALL wait$8272;
 OUTPUT(port)=iopb.gap$3;

 /*byte 9 */
 CALL wait$8272;
 OUTPUT(port)=0ffh;

 RETURN;
 END write$data;
```

```
/**************************************************************************

 FORMAT$COMMAND:
    This module contains the code to program the 8272 for a format track
    instruction.

**************************************************************************/

format$command:
 PROCEDURE PUBLIC;

 temp1=iopb.mt$mf$sk AND 02h;

 /*byte 1 */
 CALL wait$8272;
 OUTPUT(port)=(shl(temp1,5) OR 0dh);
 CALL wait$8272;

 /*byte 2 */
 temp2=(shr((iopb.head$drive AND 10h),2) + (iopb.head$drive AND 03h));
 OUTPUT(port)=temp2;

 /*byte 3 */
 CALL wait$8272;
 OUTPUT(port)=iopb.n;

 /*byte 4 */
 CALL wait$8272;
 OUTPUT(port)=iopb.sectors$per;

 /*byte 5 */
 CALL wait$8272;
 OUTPUT(port)=iopb.gap$3;

 /*byte 6 */
 CALL wait$8272;
 OUTPUT(port)=4eh;

 RETURN;
 END format$command;
```

```
/*****************************************************************************

 READ$ID:
    This module contains the code to program the 8272 for a read id
    instruction.

*****************************************************************************/

read$id:
 PROCEDURE PUBLIC;

 /*byte 1 */
 temp1=shl(iopb.mt$mf$sk AND 02h,5);
 temp2=temp1 OR 0ah;
 CALL wait$8272;
 OUTPUT(port)=temp2;

 /*byte 2 */
 temp1=(shr((iopb.head$drive AND 10h),2) + (iopb.head$drive AND 03h));
 CALL wait$8272;
 OUTPUT(port)=temp1;

 RETURN;
 END read$id;
```

```
/ ***************************************************************************

SPECIFY:
    This module contains the code to set up the iSBC 208 to the desired
    disk controller parameters.  The parameters to be INPUT to the FDC
    are: Head Unload Time, the time to wait before removing the
    read/write head from the disk; Step Rate, the time to wait between
    step pulses on seeks; and Head Load Time, the time to wait before
    loading the read/write head onto the disk.  The module will program
    the 8272 with the correct values.  This module features full error
    checking for INPUT boundaries.

***************************************************************************/

specify:
PROCEDURE PUBLIC;
DECLARE (byte$2,byte$3) BYTE;
DECLARE (step$rate,hd$unld$tm,stp$rate,hd$ld$tm) BYTE;

/* reset the iSBC 208 before beginning the SPECIFY command */
  OUTPUT(base$addr + 13h)=0ffh; /* board reset */

/* set the head unload time to 10 */
  hd$unld$tm=10;

/* set the step rate to 3 */
  step$rate=3;

/* set the head load time to 70 */
  hd$ld$tm=70;

/* this section puts the info in the form that the FDC wants */
  byte$2=hd$unld$tm + shl(((step$rate - 1) XOR 0ffh),4);
  stp$rate=(step$rate - 1) XOR 0ffh;

 /* set up the byte for the FDC */
  byte$3=shl(hd$ld$tm,1) AND 0feh;

 /* now start spitting the bytes to the FDC as needed */
  CALL wait$8272;
  OUTPUT(port)=03h; /* specify command */
  CALL wait$8272;
  OUTPUT(port)=byte$2; /* head unload time and step rate */
  CALL wait$8272;
  OUTPUT(port)=byte$3; /* Head load time */

 /* ok we are done, now return */
RETURN;
END specify;
```

```
/   this is the main driver routine */

execute$instruction:
 PROCEDURE PUBLIC;

/* first set the retry counter to 0 */
 retry$number=0;

/* now enter the section to take the IOPB apart */

skip:
 retry=0;  /* reset retrys */

 IF iopb.instruction<= 9/* for all instructions less than 9, DMA is needed */
 THEN DO:
 /* DMA is required */
 CALL split (iopb.buffer); /* split pointer into 2 words */
 OUTPUT(base$addr + 14h)=LOW(seg$val); /* low byte of segment register */
 OUTPUT(base$addr + 15h)=HIGH(seg$val); /* programming segment reg on 208 */
 OUTPUT(base$addr + 0ch)=0; /* clear first/last flip-flip */
 OUTPUT(base$addr + 0dh)=0; /* master clear the DMA chip */
 OUTPUT(base$addr + 0bh)=00h; /* mode reg for channel 0 */
 OUTPUT(base$addr + 0bh)=01h; /* mode reg for channel 1 */
 OUTPUT(base$addr + 0bh)=02h; /* mode reg for channel 2 */
 OUTPUT(base$addr + 0bh)=03h; /* mode reg for channel 3 */
 OUTPUT(base$addr + 0ch)=0;   /* first/last flop */
 iopb.byte$cnt=iopb.byte$cnt-1;
/* PROGRAM CHANNEL 0 OF 8237 */
   OUTPUT(base$addr)=LOW(offset$val); /* base address of memory buffer */
   OUTPUT(base$addr)=HIGH(offset$val);
   OUTPUT(base$addr + 0ch)=0; /* clear first/last flip-flop */

 /* CHANNEL 0 */
  OUTPUT(base$addr + 01h)=LOW(iopb.byte$cnt); /* number of bytes to move */
  OUTPUT(base$addr + 01h)=HIGH(iopb.byte$cnt);
  OUTPUT(base$addr + 08h)=0; /* DMA chip command */

  /* determine if read or write memory - DMA direction - */
  IF (iopb.instruction=1) or (iopb. instruction=4)
  OR (iopb.instruction=5)
   THEN temp3=44h; /* disk to memory transfer */
  ELSE temp3=48h;  /* memory to disk transfer */

  OUTPUT(base$addr + 0bh)=temp3; /* DMA mode register */
  OUTPUT(base$addr + 0fh)=0fah;  /* DMA mask register */

 END; /* end of DMA section */$
```

```
/*  now we program the 8272 chip for the operation selected.  To do this, we
    use a case statement to either call the correct procedure or execute the
    proper statements. */

DO CASE iopb.instruction - 1; /* between 0 and 14 */

/* case 0 */ CALL read$data(0);
/* case 1 */ CALL write$data(0);
/* case 2 */ CALL write$data(1);
/* case 3 */ CALL read$data(1);
/* case 4 */ CALL read$data(2);
/* case 5 */ RETURN; /* not used */
/* case 6 */ RETURN; /* not used */
/* case 7 */ RETURN; /* not used */
/* case 8 */ CALL format$command;
/* case 9 */ CALL read$id;

/* case 10 */ DO; CALL wait$8272;
                  OUTPUT(port)=07h;
                  CALL wait$8272;
                  OUTPUT(port)=iopb.head$drive AND 03h;
              END;

/* case 11 */ DO; CALL wait$8272;
                  OUTPUT(port)=08h;
              END;

/* case 12 */ CALL specify;

/* case 13 */ DO; CALL wait$8272;
                  OUTPUT(port)=04h;
                  temp1=(shr((iopb.head$drive AND 10h),2)
                       + (iopb.head$drive AND 03h));
                  CALL wait$8272;
                  OUTPUT(port)=temp1;
              END;

/* case 14 */ DO; CALL wait$8272;
                  OUTPUT(port)=0fh;
                  temp1=(shr((iopb.head$drive AND 10h),2)
                       + (iopb.head$drive AND 03h));
                  CALL wait$8272;
                  OUTPUT(port)=temp1;
                  CALL wait$8272;
                  OUTPUT(port)=iopb.track$no;
              END;

END; /* end of do case block */
```

```
IF iopb.instruction=15 THEN RETURN; /* return now if seek */
IF (iopb.instruction=13) OR (iopb.instruction=11) THEN RETURN;
IF (iopb.instruction=14) THEN GOTO over;

/* loop on FINT bit from 8272 to detect DONE interrupt */

temp1=0feh;
DO WHILE (INPUT(base$addr + 12h) OR temp1) = temp1;
END;

/*  Now get the status from the 8272 and fill the STATUS block.  The number of
    status bytes to be read is determined by the instruction.  Most
    instructions require 9 bytes be read, but some only need 1 or 2. */

over:
 status.pass$fail=pass; /* we made it to here ok */
 IF iopb.instruction<=10
  THEN DO;
   CALL wait1$8272;
   status.zero=INPUT(port);
   CALL wait1$8272;
   status.one=INPUT(port);
   CALL wait1$8272;
   status.two=INPUT(port);
   CALL wait1$8272;
   status.n=INPUT(port);
   CALL wait1$8272;
   status.head$addr=INPUT(port);
   CALL wait1$8272;
   status.sector$no=INPUT(port);
   CALL wait1$8272;
   status.n=INPUT(port);
```

```
                              /  see if a retry is needed */

 IF (status.zero AND 0c0h)=0 THEN retry=0;
 ELSE DO;
      retry$number=retry$number+1;
      IF retry$number>3 THEN retry=0;
      ELSE retry=1;
      END;
 IF retry=1 THEN GOTO skip;

 RETURN;
END;

IF (iopb.instruction=14)
 THEN DO; CALL wait1$8272;
          status.zero=INPUT(port);
          RETURN ;
      END;

IF iopb.instruction=12
 THEN DO; CALL wait1$8272;
          status.zero=INPUT(port);
          CALL wait1$8272;
          status.track$no=INPUT(port);
          RETURN;
      END;

 IF iopb.instruction>=16 THEN RETURN;

 status.pass$fail=fail;
 RETURN;
END execute$instruction;
END execute$mode; /*jbe*/
```

```
; ****************************************************************************
;
; SPLIT:
;
; ACCEPTS: ONE POINTER IN THE STACK POINTED TO BY THE BP REG
;
; DESTROYS: SI,DI
;
; USES: TWO MEMORY LOCATIONS TO PLACE THE SEGMENT AND OFFSET INTO
;       WHICH ARE DECLARED PUBLIC IN STRUCT.
;
; FUNCTION: TO SPLIT THE POINTER INTO TWO WORDS, A SEGMENT AND AN OFFSET,
;           SO THAT PLM/86 CAN PROGRAM THEM INTO THE 208 BOARD. THE TWO
;           WORDS ARE RETURNED IN THE LOCATIONS SEG_VAL AND OFFSET_VAL.
;
;****************************************************************************
NAME      POINTER_SPLITTER

PUBLIC SPLIT

DGROUP   GROUP   DATA
CGROUP   GROUP   CODE
ASSUME CS:CODE,DS:DATA


DATA     SEGMENT WORD     PUBLIC 'DATA'

   EXTRN SEGVAL:WORD,OFFSETVAL:WORD

DATA ENDS

CODE     SEGMENT WORD     PUBLIC 'CODE'

SPLIT    PROC  FAR     ;FAR CALL TO AID IN THE LINK

         PUSH  BP
         MOV   BP,SP
         MOV   DI,[BP+8] ;GET THE SEGMENT OUT OF THE STACK
         MOV   SI,[BP+6]  ;GET THE OFFSET OUT OF THE STACK
         MOV   SEGVAL,DI ;PUT THE SEGMENT INTO MEMORY
         MOV   OFFSETVAL,SI  ;PUT THE OFFSET INTO MEMORY
         POP   BP
         RET   4                     ;RETURN TO CALLER

SPLIT    ENDP
CODE     ENDS
```

A-3. ASSEMBLY LANGUAGE DRIVER

```
ASM80 DR208.SOR


ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    1

  LOC  OBJ         LINE        SOURCE STATEMENT

                         ;*******************************************************
                         ;*******************************************************
                         ; SBC 208 FLEXIBLE DISK CONTROLLER I/O SUBROUTINES
                         ; MAY 1981
                         ;*******************************************************
                         ;*******************************************************
  0010                   VER     EQU     10H     ;VERSION 1.0
                         ;*******************************************************
                         ; THERE ARE TWO LEVELS OF SUBROUTINES PROVIDED.
                         ;
                         ; LEVEL 1 SUBROUTINES ARE THE PRIMITIVE SUBROUTINES.
                         ; WHEN USING THESE SUBROUTINES, IT IS POSSIBLE TO WRITE
                         ; SPECIAL USER ORIENTED SUBROUTINES TO PERFORM DISKETTE
                         ; I/O OPERATIONS
                         ;
                         ; THE LEVEL 1 SUBROUTINES:
                         ;       - ARE USER CALLABLE THRU THE JUMP TABLE
                         ;       - USE ALL 8085 REGISTERS
                         ;       - RETURN IMMEDIATELY IF FDC IS BUSY
                         ;       - FOLLOW PL/M CONVENTIONS FOR PARAMETER PASSING
                         ;
                         ;
                         ; LEVEL 2 SUBROUTINES PERFORM DISKETTE I/O BY CALLING
                         ; THE PROPER SUBROUTINE.
                         ;
                         ; THE LEVEL 2 SUBROUTINES:
                         ;       - ARE USER CALLABLE THRU THE JUMP TABLE
                         ;       - PASS PARAMETERS VIA IOPB (INPUT/OUTPUT
                         ;         PARAMETER BLOCK)
                         ;       - SAVE AND RESTORE ALL REGISTERS EXCEPT CARRY
                         ;               CARRY=0 FOR NORMAL RETURN
                         ;               CARRY=1 INDICATES THAT A COMMAND HAND
                         ;                       SHAKING ERROR HAS OCCURED (BITS
                         ;                       6 AND/OR 7 OF THE MAIN STATUS
                         ;                       REGISTER ARE IN THE WRONG STATE)
                         ;       - WAIT IF FDC IS BUSY
                         ;       - FOLLOW PL/M CONVENTIONS FOR PARAMETER PASSING
                         ;       - AUTOMATICALLY SEEK TO CYLINDER NO. IN IOPB.
                         ;       - USE NSEC (NUMBER OF SECTORS) FOR MULTIPLE
                         ;         SECTOR I/O
                         ;       - WHEN  RETURNING  FROM THE  INTERRUPT SERVICE
                         ;         ROUTINE, THE "D" REGISTER INDICATES THE TYPE
                         ;         OF TERMINATION THAT WAS COMPLETED.
                         ;               D = 0 NORMAL
                         ;               D = 1 ABNORMAL, EXAMINE THE RESULT BYTES
                         ;                     TO DETERMINE WHY THE OPERATION WAS
                         ;                     NOT COMPLETED SUCCESSFULLY.
                         ;       - THE  FORMAT  OF  EACH  IOPB IS  SHOWN  AT THE
                         ;         BEGINNING OF EACH ROUTINE.  NOTE THAT COMMAND
                         ;         BITS ARE NOT  REQUIRED IN THE FIRST PARAMETER
                         ;         AS THEY ARE INSERTED  BY THE SPECIFIC ROUTINE
                         ;         CALLED.
                         ;       - EXTENSIVE ERROR MESSAGES ARE PROVIDED TO HELP
                         ;         DURING  PROGRAM  DEBUGGING.  THE ROUTINES AND
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    2

 LOC  OBJ         LINE        SOURCE STATEMENT

                        ;          SOURCE LINES  USED TO  PROVIDE THESE MESSAGES
                        ;          ARE MARKED  AT THE END OF THE LINE WITH A "#"
                        ;          SIGN. THEY CAN BE DELETED TO SAVE MEMORY SPACE
                        ;          WHEN DEBUGGING IS COMPLETE.
                        ;*****************************************************
                        ; 8259 EQUATES
 00DA                   ICCP     EQU     0DAH     ;INTERRUPT CONTROLLER CMND PORT
 0020                   EOIC     EQU     020H     ;END OF INTERRUPT COMMAND WORD
                        ;*****************************************************
 0040                   DMAMD    EQU     40H      ;DMA (8237) MODE
                                                  ; CH0, VERIFY,AUTO-DISABLED,
                                                  ;  ADDRESS INC, SINGLE
 0080                   RD       EQU     80H      ;8237 READ BIT
 0040                   WR       EQU     40H      ;8237 WRITE BIT
                        ; I/O PORT EQUATES
                        PUBLIC  BASE,DMAR0,DMWC0,DMAR1,DMWC1,DMAR2,DMWC2,DMAR3
                        PUBLIC  DMWC3,DMSR,DMRQ,DMKSR,DMODE,DCLFL,DMCLR,DMTR
                        PUBLIC  DMASK,FDCST,FDCDT,AUXP,SFTRS,SEGLO,SEGHI,READ
                        PUBLIC  RDDD,WRITE,WRTDD,RDTRK,RDID,FRMTK,AUXRST,AUXSET
                        PUBLIC  AUXADR,SCNEQ,SCNLE,SCNHE,RECAL,SEEK,SPCFY,SNSDS
                        PUBLIC  SNSIS,INIT,INT20,PRSLT
 0000                   BASE     EQU     0H       ;BASE PORT ADDRESS FOR FDC BOARD
 0000                   DMAR0    EQU     BASE+0   ;DMAC CH.0 ADDRESS REG
 0001                   DMWC0    EQU     BASE+1   ;DMAC CH.0 WORD COUNT REG
 0002                   DMAR1    EQU     BASE+2   ;DMAC CH.1 ADDRESS REG
 0003                   DMWC1    EQU     BASE+3   ;DMAC CH.1 WORD COUNT REG
 0004                   DMAR2    EQU     BASE+4   ;DMAC CH.2 ADDRESS REG
 0005                   DMWC2    EQU     BASE+5   ;DMAC CH.2 WORD COUNT REG
 0006                   DMAR3    EQU     BASE+6   ;DMAC CH.3 ADDRESS REG
 0007                   DMWC3    EQU     BASE+7   ;DMAC CH.3 WORD COUNT REG
 0008                   DMSR     EQU     BASE+8   ;DMAC STATUS REGISTER
 0009                   DMRQ     EQU     BASE+9   ;DMAC REQUEST REGISTER
 000A                   DMKSR    EQU     BASE+0AH         ;DMAC MASK SET/RESET REG
 000B                   DMODE    EQU     BASE+0BH         ;DMAC MODE REG.
 000C                   DCLFL    EQU     BASE+0CH         ;DMAC CLEAR 1ST/LAST F/F
 000D                   DMCLR    EQU     BASE+0DH         ;DMAC MASTER CLEAR
 000E                   DMTR     EQU     BASE+0EH         ;DMAC READ TEMP. REG.
 000F                   DMASK    EQU     BASE+0FH         ;DMAC WRITE MASK REG.
 0010                   FDCST    EQU     BASE+10H         ;FDC MAIN STATUS REG.
 0011                   FDCDT    EQU     BASE+11H         ;FDC DATA REGISTER
 0012                   AUXP     EQU     BASE+12H         ;AUXILIARY PORT
 0013                   SFTRS    EQU     BASE+13H         ;SOFTWARE RESET
 0014                   SEGLO    EQU     BASE+14H         ;LOWER BYTE SEGMENT REG.
 0015                   SEGHI    EQU     BASE+15H         ;UPPER BYTE SEGMENT REG.
                        ;
                        ; ENTRY POINT JUMP TABLE
 0800                   ORG      800H
 0800 C35508                     JMP     INT10    ;SINGLE LEVEL INTERRUPT (NO 8259)
 0803 C3D108                     JMP     INT20    ;INTERRUPT WITH 8259 SERVICE
 0806 C31809                     JMP     INIT     ;INITIALIZE DMA CONTROLLER
                        ;*****************************************************
                        ; LEVEL 1 ROUTINES
 0809 C3030B                     JMP     CMNDS    ;COMMAND SERIAL OPERATION
 080C C30B0B                     JMP     CMNDP    ;COMMAND PARALLEL OPERATION
 080F C3300B                     JMP     RSULT    ;RESULT PHASE
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0         MODULE   PAGE    3

  LOC  OBJ          LINE        SOURCE STATEMENT

 0812 C3DF0A                  JMP     DMARD   ;DMA READ, I/O WRITE
 0815 C3E40A                  JMP     DMAWR   ;DMA WRITE, I/O READ
                         ;*****************************************************
                         ; LEVEL 2 ROUTINES
 0818 C32A09                  JMP     READ    ;READ
 081B C33309                  JMP     RDDD    ;READ DELETED DATE
 081E C33C09                  JMP     WRITE   ;WRITE
 0821 C34509                  JMP     WRTDD   ;WRITE DELETED DATA
 0824 C34E09                  JMP     RDTRK   ;READ TRACK
 0827 C36509                  JMP     RDID    ;READ ID
 082A C36E09                  JMP     FRMTK   ;FORMAT TRACK
 082D C38809                  JMP     SCNEQ   ;SCAN EQUAL
 0830 C39109                  JMP     SCNLE   ;SCAN LOW OR EQUAL
 0833 C39A09                  JMP     SCNHE   ;SCAN HIGH OR EQUAL
 0836 C3A309                  JMP     RECAL   ;RECALIBRATE
 0839 C3BA09                  JMP     SEEK    ;SEEK
 083C C3D309                  JMP     SPCFY   ;SPECIFY
 083F C3EA09                  JMP     SNSDS   ;SENSE DRIVE STATUS
 0842 C3040A                  JMP     SNSIS   ;SENSE INTERRUPT STATUS
                         ;*****************************************************
                         ; AUXILIARY PORT ROUTINES
 0845 C3AC0A                  JMP     AUXRST  ;RESET-A-BIT (DRIVE CONTROL
                                              ; FUNCTIONS)
 0848 C3B90A                  JMP     AUXSET  ;SET-A-BIT (DRIVE CONTROL
                                              ; FUNCTIONS)
 084B C3D10A                  JMP     AUXADR  ;1 MEGABYTE PAGE ADDRESS BITS
                         ;*****************************************************
 7F80                    USTACK  EQU     7F80H   ;STACK POINTER
 7F30                    REGF    EQU     USTACK-50H      ;RESERVE 80 LOCATIONS
                                                         ; FOR THE STACK
 084E 297F               ARSBF:  DW      REGF-7  ;RESERVE 7 LOCATIONS FOR RESULT
                                                 ; BYTES
 0850 C34300             CONO:   JMP     43H     ;CONSOLE OUTPUT             #
                                                 ; PASS DATA BYTE IN C
                                                 ; ASSUMES B, DE, HL PRESERVED
 0853 20                 DELAY:  DB      20H     ;DELAY FOR FDC STATUS (>100US)
 0854 10                 VERSION:DB      VER     ;VERSION NUMBER
                         ;*****************************************************
                         ; INTERRUPT SERVICE ROUTINE (NON-8259 SYSTEM)
                         ;
                         ;       CALLING SEQUENCE
                         ;       CALL    INT10
                         ;
                         ;
                         ; REGS: AF, BC, D, HL
                         ; STK PRS: 4+CONO
 0855 1600               INT10:  MVI     D,0     ;CLEAR ABNORMAL TERMINATION FLAG
 0857 DB10                       IN      FDCST   ;A = FDC STATUS
 0859 47                         MOV     B,A     ;SAVE STATUS
 085A E610                       ANI     10H     ;FDC BUSY?
 085C C2B108                     JNZ     IT010   ;YES, IS A READ, RDDD, WRITE,
                                                 ; WRTDD, RDTRK, RDID, FRMTK,
                                                 ; SCNEQ, SCNLE, OR SCNHE INT.
                         ; SEEK/RECALIBRATE RESULT PHASE
                         ; OR ATTENTION, ABNORMAL, INTERRUPT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0          MODULE   PAGE    4

 LOC  OBJ          LINE        SOURCE STATEMENT

 085F 21C30B        165          LXI     H,MSG30 ;"FDC SEEK/ATTN INT"              #
 0862 CD720B        166          CALL    MESSG   ;PRINT MESSAGE                    #
 0865 CD5B0B        167 IT002:   CALL    RDYC    ;FDC READY FOR COMMAND?
 0868 D8            168          RC              ;NO"FDC ERR DIO HI IN CMD PHASE"
 0869 3E08          169          MVI     A,08H   ;YES
 086B D311          170          OUT     FDCDT   ;SENSE INTERRUPT STATUS
                    171                          ; (CLEARS INTERRUPT FROM 8272)
 086D CD300B        172          CALL    RSULT   ;READ FDC STATUS
 0870 D8            173          RC              ;ERROR"DIO ERR LO IN RSLT PHASE"
 0871 CDE108        174          CALL    PRSLT   ;PRINT RESULT BYTES               #
 0874 2A4E08        175          LHLD    ARSBF
 0877 7E            176          MOV     A,M     ;A = STATUS REGISTER 0
 0878 47            177          MOV     B,A     ;SAVE STATUS
 0879 E6C0          178          ANI     0C0H    ;EXAMINE UPPER TWO BITS
 087B FE80          179          CPI     80H     ;INVALID COMMAND? (10)
 087D C8            180          RZ              ;YES, RETURN
 087E B7            181          ORA     A       ;NORMAL TERMINATION? (00)
 087F CA6508        182          JZ      IT002   ;YES,CHECK FOR HIDDEN INTERRUPTS
 0882 FEC0          183          CPI     0C0H    ;NO, ATTENTION INTERRUPT? (11)
 0884 CA9508        184          JZ      IT008   ;YES
 0887 1601          185          MVI     D,1     ;SET ABNORMAL TERMINATION FLAG
 0889 21E50B        186          LXI     H,MSG50 ;NO, ABNORMAL TERMINATION (01) #
 088C CD720B        187          CALL    MESSG   ;"FDC SEEK ERR"                   #
 088F CDE108        188          CALL    PRSLT   ;PRINT RESULT BYTES               #
 0892 C36508        189          JMP     IT002   ;CHECK FOR HIDDEN INTERRUPTS
                    190 ; ATTENTION INTERRUPT
 0895 3E08          191 IT008:   MVI     A,08H   ;A=FDD READY MASK FOR ST0
 0897 A0            192          ANA     B       ;IS THE FDD READY?
 0898 C26508        193          JNZ     IT002   ;NO, CHECK FOR HIDDEN INTERRUPTS
                    194 ;RECALIBRATE FOR NEWLY MOUNTED DISK
 089B 3E03          195          MVI     A,3H    ;A=UNIT SELECT MASK
 089D A0            196          ANA     B       ;A = US?
 089E 47            197          MOV     B,A     ;SAVE UNIT SELECT
 089F CD5B0B        198          CALL    RDYC
 08A2 D8            199          RC              ;ERROR "DIO ERR"
 08A3 3E07          200          MVI     A,07H
 08A5 D311          201          OUT     FDCDT   ;RECALIBRATE
 08A7 CD5B0B        202          CALL    RDYC
 08AA D8            203          RC              ;ERROR "DIO ERR"
 08AB 78            204          MOV     A,B
 08AC D311          205          OUT     FDCDT   ;OUTPUT UNIT SELECT
 08AE C36508        206          JMP     IT002   ;CHECK FOR HIDDEN INTERRUPTS
                    207 ; I/O INTERRUPT
 08B1 21D70B        208 IT010:   LXI     H,MSG40 ;"FDC I/O INT"                    #
 08B4 CD720B        209          CALL    MESSG   ;PRINT MESSAGE                    #
 08B7 CD300B        210          CALL    RSULT   ;GET RESULT BYTES
                    211                          ; AND RESET 8272 INTERRUPT
 08BA D8            212          RC              ;ERROR "DIO ERR"
 08BB CDE108        213          CALL    PRSLT   ;PRINT RESULT BYTES               #
 08BE 2A4E08        214          LHLD    ARSBF
 08C1 7E            215          MOV     A,M     ;A=STATUS REGISTER 0
 08C2 E6C0          216          ANI     0C0H    ;NORMAL TERMINATION?
 08C4 C8            217          RZ              ;YES, "INTERRUPT CODE IS ZERO"
 08C5 21F40B        218          LXI     H,MSG60 ;NO, ERROR                        #
 08C8 CD720B        219          CALL    MESSG   ;"FDC I/O ERR"                    #
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    5

 LOC  OBJ         LINE        SOURCE STATEMENT

08CB CDE108                CALL    PRSLT    ;PRINT RESULT BYTES          #
08CE 1601                  MVI     D,1      ;SET ABNORMAL TERMINATION FLAG
08D0 C9                    RET
                  ;*****************************************************
                  ; INTERRUPT SERVICE ROUTINE (SYSTEMS WITH 8259)
                  ;        SAVES ALL REGISTERS, PROCESSES INT FROM 8272
                  ;        RESTORES ALL REGISTER, AND RETURNS
                  ;
                  ; REGS: NONE
                  ; STK PRS: 9+CONO (NOTE: INTERRUPT USES 1 PAIR)
08D1 CD190A       INT20:   CALL    SAVER    ;SAVE REGISTERS
08D4 3E20                  MVI     A,EOIC   ;A=END OF INTERRUPT COMMAND
08D6 D3DA                  OUT     ICCP     ;RESET INTERRUPT CONTROLLER
                  ; NOTE: 8259 MUST USE EDGE TRIGGER MODE
08D8 CD5508                CALL    INT10    ;PROCESS INTERRUPT
08DB F1                    POP     PSW      ;RESTORE
08DC C1                    POP     B        ; ALL
08DD D1                    POP     D        ;  REGISTERS
08DE E1                    POP     H        ;   AND
08DF FB                    EI               ;    ENABLE INTERRUPTS
08E0 C9                    RET              ;     RETURN
                  ;*****************************************************
                  ; PRINT RESULT BYTES (THIS  ROUTINE  CAN  BE  ELIMINATED
                  ;                   WHEN ERROR MESSAGES ARE  NO  LONGER
                  ;                   REQUIRED)
                  ;
                  ;        C= # OF BYTES
                  ;        CALL   PRSLT
                  ;
                  ; REGS: AF, HL
                  ; STK PRS: 2+CONO
08E1 C5           PRSLT:   PUSH    B        ;SAVE BC
08E2 41                    MOV     B,C      ;B=BYTE COUNT
08E3 2A4E08                LHLD    ARSBF    ;ADR OF RESULT BUFFER
08E6 0E20         PR005:   MVI     C,' '    ;PRINT
08E8 CD5008                CALL    CONO     ; SPACE
08EB 7E                    MOV     A,M      ;PRINT
08EC CD0409                CALL    PR010    ; MSN
08EF 7E                    MOV     A,M      ;PRINT
08F0 CD0809                CALL    PR020    ;LSN
08F3 23                    INX     H        ;BUMP POINTER
08F4 05                    DCR     B        ;DONE?
08F5 C2E608                JNZ     PR005    ;NO
08F8 0E0D                  MVI     C,0DH    ;YES, PRINT
08FA CD5008                CALL    CONO     ;CR
08FD 0E0A                  MVI     C,0AH    ;PRINT
08FF CD5008                CALL    CONO     ; LF
0902 C1                    POP     B        ;RESTORE BC
0903 C9                    RET
                  ; PRINT MOST SIGNIFICANT NIBBLE
0904 0F           PR010:   RRC              ;MOVE
0905 0F                    RRC              ; LEFT
0906 0F                    RRC              ;  NIBBLE
0907 0F                    RRC              ;   TO RIGHT
                  ; PRINT LEAST SIGNIFICANT NIBBLE
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0          MODULE   PAGE    6

 LOC  OBJ           LINE        SOURCE STATEMENT

 0908 E60F                PRO20:  ANI     0FH     ; 0 TO 15
 090A D60A                        SUI     0AH     ; -10 TO 5
 090C 0E3A                        MVI     C,3AH   ;ASCII DISPLACEMENT FOR 0 TO 9
 090E DA1309                      JC      PRO30
 0911 0E41                        MVI     C,41H   ;ASCII DISPLACEMENT FOR A TO F
 0913 81                  PRO30:  ADD     C       ;CONVERT BINARY TO ASCII
 0914 4F                          MOV     C,A     ;PASS CHAR IN C
 0915 C35008                      JMP     CONO    ;PRINT CHAR
                          ;****************************************************
                          ; INITIALIZE 8237 DMA CONTROLLER
                          ;
                          ; REGS: F (CARRY = 0)
                          ; STK PRS: 4
 0918 F3                  INIT:   DI              ;DISABLE INTERRUPTS WHILE
                                                  ;INITIALIZING BOARD
 0919 CD190A                      CALL    SAVER
 091C 3E40                        MVI     A,DMAMD ;SELECT DMA MODE
 091E D30B                INIO:   OUT     DMODE   ;SET 8237 MODE
 0920 3C                          INR     A       ;SELECT NEXT CHANNEL
 0921 FE44                        CPI     DMAMD+4 ;LAST CHANNEL
 0923 C21E09                      JNZ     INIO    ;NO, SET ALL CHANNELS
 0926 FB                          EI              ;RE-ENABLE INTERRUPTS
 0927 C31E0A                      JMP     RSTOR   ;RETURN TO USER, CARRY=0
                          ;****************************************************
                          ; LEVEL 2 ROUTINES
                          ;       USER CALLABLE
                          ;       SAVE ALL REGISTERS EXCEPT CARRY
                          ;****************************************************
                          ; READ
                          ;
                          ;       CALLING SEQUENCE
                          ;       BC=ADR(IOPB)
                          ;               IOPB:   MT.MF,SK,X ;X=SPACE FOR COMMAND
                          ;                       HD,US    ;HEAD, UNIT
                          ;                       C        ;CYLINDER
                          ;                       H        ;HEAD
                          ;                       R        ;RECORD
                          ;                       N        ;SECTOR SIZE
                          ;                       EOT      ;END OF TRACK
                          ;                       GPL      ;GAP LENGTH
                          ;                       DTL      ;DATA LENGTH
                          ;                       NSEC     ;NUMBER OF SECTORS
                          ;       DE=ADR(DATA)
                          ;        CALL   READ
                          ;       NORMAL RETURN, CARRY=0 (NC)
                          ;       ERROR RETURN, CARRY=1 (C)
                          ;
                          ; REGS: CARRY
                          ; STK PRS: 13+CONO
 092A CD190A              READ:   CALL    SAVER   ;SAVE REGISTERS
 092D 21E649                      LXI     H,(WR+9) SHL 8 + 0E0H + 06H ;DMA WRITE,
                                                  ; 9 BYTES, MTMFSK MASK, COMMAND
 0930 C32D0A                      JMP     DTRN1   ;DATA TRANSFER COMMAND
                          ;****************************************************
                          ; READ DELETED DATA
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0         MODULE   PAGE    7

 LOC  OBJ          LINE        SOURCE STATEMENT

                         ;
                         ;       CALLING SEQUENCE
                         ;       BC=ADR(IOPB)
                         ;               IOPB:    MT,MF,SK,X;X=SPACE FOR COMMAND
                         ;                        HD,US     ;HEAD, UNIT SELECT
                         ;                        C         ;CYLINDER
                         ;                        H         ;HEAD
                         ;                        R         ;RECORD
                         ;                        N         ;SECTOR SIZE
                         ;                        EOT       ;END OF TRACK
                         ;                        GPL       ;GAP LENGTH
                         ;                        DTL       ;DATA LENGTH
                         ;                        NSEC      ;NUMBER OF SECTORS
                         ;       DE=ADR(DATA)
                         ;       CALL    RDDD
                         ;       NORMAL RETURN, CARRY=0 (NC)
                         ;       ERROR RETURN,  CARRY=1 (C)
                         ;
                         ; REGS: CARRY
                         ; STK PRS: 13+CONO
 0933 CD190A             RDDD:   CALL    SAVER    ;SAVE REGISTERS
 0936 21EC49                     LXI     H,(WR+9) SHL 8 + 0E0H + 0CH ;DMA WRITE,
                                                  ;9 BYTES, MTMFSK MASK, COMMAND
 0939 C32D0A                     JMP     DTRN1
                         ;*****************************************************
                         ; WRITE DATA
                         ;
                         ;       CALLING SEQUENCE
                         ;       BC=ADR(IOPB)
                         ;               IOPB:    MT,MF,X  ;X=SPACE FOR COMMAND
                         ;                        HD,US     ;HEAD, UNIT SELECT
                         ;                        C         ;CYLINDER
                         ;                        H         ;HEAD
                         ;                        R         ;RECORD
                         ;                        N         ;SECTOR SIZE
                         ;                        EOT       ;END OF TRACK
                         ;                        GPL       ;GAP LENGTH
                         ;                        DTL       ;DATA LENGTH
                         ;                        NSEC      ;NUMBER OF SECTORS
                         ;       DF=ADR(DATA)
                         ;       CALL    WRITE
                         ;       NORMAL RETURN, CARRY=0 (NC)
                         ;       ERROR RETURN, CARRY=1 (C)
                         ;
                         ; REGS: CARRY
                         ; STK PRS: 13+CONO
 093C CD190A             WRITE:  CALL    SAVER    ;SAVE REGISTERS
 093F 21C589                     LXI     H,(RD+9) SHL 8 + 0C0H + 05H ;DMA READ,
                                                  ;9 BYTES, MTMFSK MASK, COMMAND
 0942 C32D0A                     JMP     DTRN1
                         ;*****************************************************
                         ; WRITE DELETED DATA
                         ;
                         ;       CALLING SEQUENCE
                         ;       BC=ADR(IOPB)
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    8

 LOC  OBJ          LINE        SOURCE STATEMENT

                    ;               IOPB:    MT,MF,X ;X=SPACE FOR COMMAND
                    ;                        HD,US   ;HEAD, UNIT SELECT
                    ;                        C       ;CYLINDER
                    ;                        H       ;HEAD
                    ;                        R       ;RECORD
                    ;                        N       ;SECTOR SIZE
                    ;                        EOT     ;END OF TRACK
                    ;                        GPL     ;GAP LENGTH
                    ;                        DTL     ;DATA LENGTH
                    ;                        NSEC    ;NUMBER OF SECTORS
                    ;       DE=ADR(DATA)
                    ;       CALL    WRTDD
                    ;       NORMAL RETURN, CARRY=0 (NC)
                    ;       ERROR RETURN, CARRY=1 (C)
                    ;
                    ; REGS: CARRY
                    ; STK PRS: 13+CONO
 0945 CD190A        WRTDD:  CALL    SAVER   ;SAVE REGISTERS
 0948 21C989                LXI     H,(RD+9) SHL 8 +0C0H +09H ;DMA READ
                                             ;9 BYTES, MTMFSK MASK, COMMAND
 094B C32D0A                JMP     DTRN1
                    ;**************************************************
                    ; READ A TRACK
                    ;
                    ;       CALLING SEQUENCE
                    ;       BC=ADR(IOPB)
                    ;               IOPB:    MF,SK,X ;X=SPACE FOR COMMAND
                    ;                        HD,US   ;HEAD, UNIT SELECT
                    ;                        C       ;CYLINDER
                    ;                        H       ;HEAD
                    ;                        R       ;RECORD
                    ;                        N       ;SECTOR SIZE
                    ;                        EOT     ;END OF TRACK
                    ;                        GPL     ;GAP LENGTH
                    ;                        DTL     ;DATA LENGTH
                    ;                        NSEC    ;# OF SECTORS (NOT USED)
                    ;       DE=ADR(DATA)
                    ;       CALL    RDTRK
                    ;       NORMAL RETURN, CARRY=0 (NC)
                    ;       ERROR RETURN, CARRY=1 (C)
                    ;
                    ; REGS: CARRY
                    ; STK PRS: 14+CONO
 094E CD190A        RDTRK:  CALL    SAVER   ;SAVE USER REGISTERS
 0951 216249                LXI     H,(WR+9) SHL 8 + 60H + 02H  ;DMA WRITE,
                                             ;9 BYTES, MTMFSK MASK, COMMAND
 0954 0A                    LDAX    B        ;SAVE
 0955 F5                    PUSH    PSW      ; MT,MF,SK
 0956 C5                    PUSH    B        ;SAVE ADDR(1OPB)
 0957 E5                    PUSH    H        ;SAVE ADR (DATA)
 0958 D5                    PUSH    D        ;SAVE PARAMETERS
                    ; SEEK
 0959 CDC609                CALL    SK010
 095C DA380A                JC      DT005   ;JUMP IF ERROR
                    ; FORCE DTRN ROUTINE TO USE EOT INSTEAD OF NSEC TO
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE    9

 LOC  OBJ          LINE        SOURCE STATEMENT

                        ; CALCULATE BYTE COUNT
 095F 210300                    LXI     H,3       ;DISPLACEMENT FOR EOT
 0962 C3440A                    JMP     DT015     ;PICK  UP  IN  MIDDLE  OF DTRN
                                                  ; ROUTINE
                        ;*****************************************************
                        ; READ ID
                        ;       NOTE: MUST BE PRECEDDED BY A SEEK
                        ;       CALLING SEQUENCE
                        ;       BC=ADR(IOPB)J
                        ;               IOPB:   MF,X     ;X=SPACE FOR COMMAND
                        ;                       HD,US    ;HEAD, UNIT SELECT
                        ;       CALL    RDID
                        ;       NORMAL RETURN, CARRY=0 (NC)
                        ;       ERROR RETURN, CARRY=1 (C)
                        ;
                        ; REGS: CARRY
                        ; STK PRS: 11+CONO
 0965 CD190A            RDID:   CALL    SAVER     ;SAVE REGISTERS
 0968 214A02                    LXI     H,2 SHL 8 + 040H + 0AH ;NO. OF BYTES,
                                                  ;MTMFSK MASK, COMMAND
 096B C3910A                    JMP     DTRN2
                        ;*****************************************************
                        ; FORMAT A TRACK
                        ;       NOTE: MUST BE PRECEEDED BY A SEEK
                        ;       CALLING SEQUENCE
                        ;       BC=ADR(IOPB)
                        ;               IOPB:   MF,X     ;X=SPACE FOR COMMAND
                        ;                       HD,US    ;HEAD, UNIT SELECT
                        ;                       N        ;SECTOR SIZE
                        ;                       SC       ;SECTORS/TRACK
                        ;                       GPL3     ;GAP LENGTH
                        ;                       D        ;DATA
                        ;       DE=ADR(DATA)
                        ;       CALL    FRMTK
                        ;       NORMAL RETURN, CARRY=0 (NC)
                        ;       ERROR RETURN, CARRY=1 (C)
                        ;
                        ; REGS: CARRY
                        ; STK PRS: 11+CONO
 096E CD190A            FRMTK:  CALL    SAVER     ;SAVE REGISTERS
 0971 210300                    LXI     H,3
 0974 09                        DAD     B         ;HL POINTS TO SC PARAMETER
 0975 6E                        MOV     L,M
 0976 2600                      MVI     H,0       ;HL=SC
 0978 29                        DAD     H         ;DOUBLE COUNT
 0979 29                        DAD     H         ;QUADRUPLE COUNT
 097A 2B                        DCX     H         ;DECREMENT FOR 8237
 097B C5                        PUSH    B         ;SAVE ADR(IOPB)
 097C 4D                        MOV     C,L
 097D 44                        MOV     B,H       ;BC=COUNT
 097E CDDF0A                    CALL    DMARD     ;SET UP 8237
 0981 C1                        POP     B         ;RESTORE ADR(10PB)
 0982 214D06                    LXI     H,6 SHL 8 + 040H + 0DH  ;NO. OF BYTES,
                                                  ;MTMFSK MASK, COMMAND
 0985 C3910A                    JMP     DTRN2
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0         MODULE   PAGE   10

  LOC  OBJ          LINE        SOURCE STATEMENT

                         ;*****************************************************
                         ; SCAN EQUAL
                         ;
                         ;       CALLING SEQUENCE
                         ;       BC=ADR(IOPB)
                         ;               IOPB:   MR,MF,SK,X;X=SPACE FOR COMMAND
                         ;                       UD,US   ;HEAD, UNIT SELECT
                         ;                       C       ;CYLINDER
                         ;                       H       ;HEAD
                         ;                       R       ;RECORD
                         ;                       N       ;SECTOR SIZE
                         ;                       EOT     ;END OF TRACK
                         ;                       GPL     ;GAP LENGTH
                         ;                       STP     ;STEP (1 OR 2)
                         ;                       NSEC    ;NUMBER OF SECTORS
                         ;       DE=ADR(DATA)
                         ;       CALL    SCNEQ
                         ;       NORMAL RETURN, CARRY=0 (NC)
                         ;       ERROR RETURN, CARRY=1 (C)
                         ;
                         ; REGS: CARRY
                         ; STK PRS: 13+CONO
  0988 CD190A            SCNEQ:  CALL    SAVER   ;REGISTERS
  098B 21F189                    LXI     H,(RD+9) SHL 8 + 0E0H + 11H  ;DMA READ,
                                                 ;9 BYTES, MTMFSK MASK, COMMAND
  098E C32D0A                    JMP     DTRN1
                         ;*****************************************************
                         ; SCAN LOW OR EQUAL
                         ;
                         ;       CALLING SEQUENCE
                         ;       BC=ADR(IOPB)
                         ;               IOPB:   MT,MF,SK,X  ;X=SPACE FOR COMMAND
                         ;                       HD,US   ;HEAD, UNIT SELECT
                         ;                       C       ;CYLINDER
                         ;                       H       ;READ
                         ;                       R       ;RECORD
                         ;                       N       ;SECTOR SIZE
                         ;                       EOT     ;END OF TRACK
                         ;                       GPL     ;GAP LENGTH
                         ;                       STP     ;STEP (1 OR 2)
                         ;                       NSEC    ;NUMBER OF SECTORS
                         ;       DE=ADR(DATA)
                         ;       CALL    SCNLE
                         ;       NORMAL RETURN, CARRY=0 (NC)
                         ;       ERROR RETURN, CARRY=1  (C)
                         ;
                         ; REGS: CARRY
                         ; STK PRS: 13+CONO
  0991 CD190A            SCNLE:  CALL    SAVER   ;SAVE REGISTERS
  0994 21F989                    LXI     H,(RD+9) SHL 8 + 0E0H + 19H  ;DMA READ,
                                                 ;9 BYTES, MTMFSK MASK, COMMAND
  0997 C32D0A                    JMP     DTRN1
                         ;*****************************************************
                         ;SCAN HIGH OR EQUAL
                         ;
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0          MODULE    PAGE    11

 LOC  OBJ         LINE        SOURCE STATEMENT

                        ;       CALLING SEQUENCE
                        ;       BC=ADR(IOPB)
                        ;               IOPB:   MT,MF,SK,X  ;X=SPACE FOR COMMAND
                        ;                       HD,US      ;HEAD, UNIT SELECT
                        ;                       C          ;CYLINDER
                        ;                       H          ;HEAD
                        ;                       R          ;RECORD
                        ;                       N          ;SECTOR SIZE
                        ;                       EOT        ;END OF TRACK
                        ;                       GPL        ;GAP LENGTH
                        ;                       STP        ;STEP (1 OR 2)
                        ;                       NSEC       ;NUMBER OF SECTORS
                        ;       DE=ADR(DATA)
                        ;       CALL    SCNHE
                        ;       NORMAL RETURN, CARRY=0 (NC)
                        ;       ERROR RETURN, CARRY=1 (C)
                        ;
                        ; REGS: CARRY
                        ; STK PRS: 13+CONO
 099A CD190A            SCNHE:  CALL    SAVER   ;SAVE REGISTERS
 099D 21FD89                    LXI     H,(RD+9) SHL 8 + 0E0H + 1DH  ;DMA READ,
                                                ;9 BYTES, MTMFSK MASK, COMMAND
 09A0 C32D0A                    JMP     DTRN1
                        ;*****************************************************
                        ; RECALIBRATE
                        ;
                        ;       CALLING SEQUENCE
                        ;       BC=ADR(IOPB)
                        ;               IOPB:   X          ;X=SPACE FOR COMMAND
                        ;                       0,US       ;UNIT SELECT
                        ;       CALL    RECAL
                        ;       NORMAL RETURN, CARRY=0 (NC)
                        ;       ERROR RETURN, CARRY=1 (C)
                        ;
                        ; REGS: CARRY
                        ; STK PRS: 11+CONO
 09A3 CD190A            RECAL:  CALL    SAVER   ;SAVE REGISTERS
 09A6 0A                        LDAX    B       ;MT,MF,SK
 09A7 F5                        PUSH    PSW     ;SAVE MT,MF,SK
 09A8 C5                        PUSH    B       ;SAVE ADR(MR,MF,SK)
 09A9 3E07                      MVI     A,07H   ;A=CAMMAND
 09AB 02                        STAX    B       ;STORE COMMAND IN 1OPB
 09AC 1E02                      MVI     E,2     ;B=NO. OF BYTES IN COMMAND
 09AE CD0B0B            RE010:  CALL    CMNDP   ;RECALIBRATE
 09B1 DAA50A                    JC      DT060   ;QUIT IF ERROR
 09B4 C2AE09                    JNZ     RE010   ;WAIT IF FDC BUSY
 09B7 C3A50A                    JMP     DT060   ;NORMAL RETURN
                        ;*****************************************************
                        ; SEEK
                        ;
                        ;       CALLING SEQUENCE
                        ;       BC=ADR(IOPB)
                        ;               IOPB:   X          ;S=SPACE FOR COMMAND
                        ;                       HD,US      ;HEAD, UNIT SELECT
                        ;                       C          ;CYLINDER
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0          MODULE    PAGE    12

  LOC  OBJ         LINE        SOURCE STATEMENT

                         ;         CALL     SEEK
                         ;         NORMAL RETURN, CARRY=0 (NC)
                         ;         ERROR RETURN, CARRY=1 (C)
                         ;
                         ; REGS: CARRY
                         ; STK PRS: 12+CONO
 09BA CD190A             SEEK:     CALL     SAVER     ;SAVE REGISTERS
 09BD 0A                           LDAX     B         ;MT,MF,SK
 09BE F5                           PUSH     PSW       ;SAVE MT,MF,SK
 09BF C5                           PUSH     B         ;SAVE ADR(MR,MF,SK)
 09C0 CDC609                       CALL     SK010     ;ISSUE SEEK
 09C3 C3A50A                       JMP      DT060
                         ;
                         ; REGS: ALL
                         ; STK PRS: 5+CONO
 09C6 3E0F               SK010:    MVI      A,0FH
 09C8 02                           STAX     B         ;SET SEEK COMMAND IN IOPB
 09C9 1E03                         MVI      E,3       ;NO. OF BYTES IN COMMAND
 09CB CD0B0B             SK020:    CALL     CMNDP     ;ISSUE SEEK COMMAND
 09CE D8                           RC                 ;RETURN IF ERROR
 09CF C2CB09                       JNZ      SK020     ;WAIT FOR FDC READY
 09D2 C9                           RET
                         ;**************************************************
                         ; SPECIFY
                         ;
                         ;  CALLING SEQUENCE
                         ;         BC=ADR(IOPB)
                         ;                  IOPB:    X       ;X=SPACE FOR COMMAND
                         ;                           SRT,HUT ;STEP  RATE, AND  HEAD
                         ;                                   ; UNLOAD TIME
                         ;                           HLT,ND  ;HEAD  LOAD  TIME, AND
                         ;                                   ; NON-DMA MODE
                         ;         CALL     SPCFY
                         ;         NORMAL RETURN, CARRY=0 (NC)
                         ;         ERROR RETURN, CARRY=1 (C)
                         ;
                         ; REGS: CARRY
                         ; STK PRS: 11+CONO
 09D3 CD190A             SPCFY:    CALL     SAVER     ;SAVE REGISTERS
 09D6 0A                           LDAX     B         ;MT,MF,SK
 09D7 F5                           PUSH     PSW       ;SAVE MT,MF,SK
 09D8 C5                           PUSH     B         ;SAVE ADR(MT,MF,SK)
 09D9 1E03                         MVI      E,3       ;NO. OF BYTES IN COMMAND
 09DB 3E03                         MVI      A,03H     ;SET COMMAND WORD
 09DD 02                           STAX     B         ;IN 1OPB
 09DE CD030B             SPC10:    CALL     CMNDS     ;ISSUE COMMAND
 09E1 DAA50A                       JC       DT060     ;QUIT IF ERROR
 09E4 C2DE09                       JNZ      SPC10     ;WAIT FOR FDC READY
 09E7 C3A50A                       JMP      DT060     ;NORMAL RETURN
                         ;**************************************************
                         ; SENSE DRIVE STATUS
                         ;
                         ;         CALLING SEQUENCE
                         ;         BC=ADR(IOPB)
                         ;                  IOPB:    X       ;X=SPACE FOR COMMAND
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0         MODULE   PAGE   13

 LOC  OBJ          SOURCE STATEMENT

                   ;                           HD,US    ;HEAD, UNIT SELECT
                   ;          CALL    SNSDS
                   ;          NORMAL RETURN, CARRY=0 (NC),ST3 IN RESULT BUFFER
                   ;          ERROR RETURN, CARRY=1  (C)
                   ;
                   ; REGS: CARRY
                   ; STK PRS: 11+CONO
 09EA CD190A       SNSDS:     CALL    SAVER    ;SAVE REGISTERS
 09ED 0A                      LDAX    B        ;MT,MF,SK
 09EE F5                      PUSH    PSW      ;SAVE MT,MF,SK
 09EF C5                      PUSH    B        ;SAVE ADR(MT,MF,SK)
 09F0 3E04                    MVI     A,04H    ;A=COMMAND
 09F2 02                      STAX    B        ;STORE COMMAND IN IOPB
 09F3 1E02                    MVI     E,2      ;NO. OF BYTES IN COMMAND
 09F5 CD0B0B       SD010:     CALL    CMNDP    ;ISSUE COMMAND
 09F8 DAA50A                  JC      DT060    ;QUIT IF ERROR
 09FB C2F509                  JNZ     SD010    ;WAIT FOR FDC READY
 09FE CD300B                  CALL    RSULT    ;GET ST3
 0A01 C3A50A                  JMP     DT060    ;RETURN TO CALLER
                   ;****************************************************
                   ; SENSE INTERRUPT STATUS
                   ;
                   ;  CALLING SEQUENCE
                   ;          CALL    SNSIS
                   ;          NORMAL RETURN,CARRY=0 (NC), ST0 & PCN IN  RESULT
                   ;                                       BUFFER
                   ;          ERROR RETURN, CARRY=1 (C)
                   ;
                   ; REGS: CARRY
                   ; STK PRS: 9+CONO
 0A04 CD190A       SNSIS:     CALL    SAVER    ;SAVE REGISTERS
 0A07 01180A                  LXI     B,SNSIC  ;ADR(IOPB)
 0A0A 1E01                    MVI     E,1      ;NO. OF BYTES
 0A0C CD210B                  CALL    CMND     ;ISSUE COMMAND
 0A0F DA1E0A                  JC      RSTOR    ;QUIT IF ERROR
 0A12 CD300B                  CALL    RSULT    ;GET RESULTS
 0A15 C31E0A                  JMP     RSTOR    ;RETURN
 0A18 08           SNSIC:     DB      08H
                   ;****************************************************
                   ; SAVE REGISTERS ON STACK
                   ;
                   ; REGS: HL
                   ; STK PRS: 3
 0A19 E3           SAVER:     XTHL             ;SAVE HL ON STACK
                                               ;HL=ADR(CALLER)
 0A1A D5                      PUSH    D        ;SAVE DE
 0A1B C5                      PUSH    B        ;SAVE BC
 0A1C F5                      PUSH    PSW      ;SAVE AF
 0A1D E9                      PCHL             ;RETURN TO CALLER
                   ;****************************************************
                   ; RESTORE REGISTERS FROM STACK
                   ;
                   ;          NORMAL RETURN, CARRY=0
                   ;          ERROR RETURN, CARRY=1
                   ;
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0          MODULE    PAGE    14

  LOC  OBJ          LINE        SOURCE STATEMENT

                    ; REGS: CARRY
                    ; STK PRS: -4
 0A1E DA270A        RSTOR:  JC      RSTC     ;RESTORE WITH CARRY=1
 0A21 F1                    POP     PSW      ;RESTORE WITH CARRY=0
 0A22 37                    STC
 0A23 3F                    CMC
 0A24 C3290A                JMP     RSBDH    ;RESTORE B,D,H
 0A27 F1            RSTC:   POP     PSW
 0A28 37                    STC
 0A29 C1            RSBDH:  POP     B        ;RESTORE BC
 0A2A D1                    POP     D        ;RESTORE DE
 0A2B E1                    POP     H        ;RESTORE HL
 0A2C C9                    RET              ;RETURN TO USER
                    ;***************************************************
                    ; DATA TRANSFER COMMAND ROUTINE
                    ;
                    ; CALLING SEQUENCE
                    ;       CALL    SAVER
                    ;       BC=ADR(IOPB)
                    ;       DE=ADR(DATA)
                    ;       L=MMMCCCCC       WHERE MMM=MTMFSK MASK
                    ;                        CCCCC=FDC COMMAND
                    ;       H=RWNNNNNN       WHERE R=8237 RD BIT
                    ;                        W=8237 WR BIT
                    ;                        NNNNNN=NO. OF BYTES IN COMMAND
                    ;       JMP     DTRN1
                    ;       OR
                    ;       BC=ADR(IOPB)
                    ;       L=MMMCCCCC       WHERE MMM=MTMFSK MASK
                    ;                        CCCCC=NO. OF BYTES IN COMMAND
                    ;       H=XXNNNNNN       WHERE XX=DON'T CARE
                    ;                        NNNNNN= NO. OF BYTES IN COMMAND
                    ;       JMP     DTRN2
                    ;
                    ;       NORMAL RETURN, CARRY=0 (NC)
                    ;       ERROR RETURN, CARRY=1 (C)
                    ;
                    ; REGS: ALL
                    ; STK PRS: 9+CONO
 0A2D 0A            DTRN1:  LDAX    B        ;MT,MF,SK
 0A2E F5                    PUSH    PSW      ;SAVE MT,MF,SK
 0A2F C5                    PUSH    B        ;SAVE ADR(IOPB)
 0A30 E5                    PUSH    H        ;SAVE PARAMETERS
 0A31 D5                    PUSH    D        ;SAVE ADR(DATA)
                    ; SEEK
 0A32 CDC609                CALL    SK010    ;SEEK
 0A35 D2410A                JNC     DT010    ;JUMP IF OK
 0A38 E1            DT005:  POP     H        ;ERROR
 0A39 D1                    POP     D        ;IN
 0A3A C1                    POP     B        ;  SEEK
 0A3B F1                    POP     PSW      ;RESTORE
 0A3C 02                    STAX    B        ;MT,MF,SK
 0A3D 37                    STC
 0A3E C31E0A                JMP     RSTOR    ;RETURN WITH CARRY=1
                    ; CALCULATE SECTOR SIZE
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0          MODULE   PAGE   15

  LOC  OBJ            LINE        SOURCE STATEMENT

  0A41 210600              DT010:  LXI     H,6
  0A44 09                  DT015:  DAD     B           ;ADR(NSEC)
  0A45 5E                          MOV     E,M         ;SAVE NSEC
  0A46 210200                      LXI     H,2
  0A49 09                          DAD     B           ;HL=ADR(N)
  0A4A 7E                          MOV     A,M         ;A=N=(0 TO 3)
  0A4B B7                          ORA     A           ;N=0?
  0A4C C2550A                      JNZ     DT020       ;NO
  0A4F 218000                      LXI     H,80H       ;YES, SET SECTOR SIZE=128
  0A52 C3600A                      JMP     DT040
  0A55 210001              DT020:  LXI     H,256       ;HL=BASE SECTOR SIZE
  0A58 3D                  DT030:  DCR     A           ;DONE?
  0A59 CA600A                      JZ      DT040       ;YES, HL=SECTOR SIZE
  0A5C 29                          DAD     H           ;NO, DOUBLE THE VALUE
  0A5D C3580A                      JMP     DT030
  0A60 7B                  DT040:  MOV     A,E         ;RECALL NSEC
  0A61 54                          MOV     D,H         ;SAVE
  0A62 5D                          MOV     E,L         ;SECTOR SIZE
                           ; MULTIPLY SECTOR SIZE BY NSEC
  0A63 3D                  DT042:  DCR     A           ;DONE?
  0A64 CA6B0A                      JZ      DT045       ;YES
  0A67 19                          DAD     D           ;NO, ADD ANOTHER SECTOR SIZE
  0A68 C3630A                      JMP     DT042       ;CHECK AGAIN
  0A6B 2B                  DT045:  DCX     H           ;HL=(SECTOR SIZE) * NSEC-1
                           ; SET UP DMA CONTROLLER
  0A6C EB                          XCHG                ;DE=8237 WORD COUNT
  0A6D F3                          DI                  ;DISABLE INTERRUPTS WHILE
                                                       ;PROGRAMMING 8237
  0A6E D30C                        OUT     DCLFL       ;CLEAR F/L F/F
  0A70 7B                          MOV     A,E
  0A71 D301                        OUT     DMWC0       ;PROGRAM LSB OF COUNT
  0A73 7A                          MOV     A,D
  0A74 D301                        OUT     DMWC0       ;PROGRAM MSB OF COUNT
  0A76 D1                          POP     D           ;RESTORE ADR (DATA)
  0A77 7B                          MOV     A,E
  0A78 D300                        OUT     DMAR0       ;PROGRAM LSB OF ADDRESS
  0A7A 7A                          MOV     A,D
  0A7B D300                        OUT     DMAR0       ;PROGRAM MSB OF ADDRESS
  0A7D E1                          POP     H           ;HL=PARAMETERS
  0A7E 3EC0                        MVI     A,0C0H      ;MASK FOR RD, WR BITS
  0A80 A4                          ANA     H           ;A=RD,WR BIT
  0A81 1F                          RAR                 ;POSITION
  0A82 1F                          RAR                 ; RD
  0A83 1F                          RAR                 ;  WR
  0A84 1F                          RAR                 ;   BIT
  0A85 1640                        MVI     D,DMAMD     ;DMA MODE WORD
  0A87 B2                          ORA     D           ;OR RD/WR BIT WITH MODE WORD
  0A88 D30B                        OUT     DMODE       ;SET MODE
  0A8A AF                          XRA     A           ;DMAC MASK VALUE
  0A8B D30A                        OUT     DMKSR       ;ENABLE DMA TRANSFER ON CH.0
  0A8D FB                          EI                  ;RE-ENABLE INTERRUPTS
  0A8E C1                          POP     B           ;RESTORE ADR(IOPB)
  0A8F F1                          POP     PSW         ;RESTORE
  0A90 02                          STAX    B           ;MT,MF,SK
                           ; COMMAND FDC
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   16

  LOC  OBJ         LINE        SOURCE STATEMENT

                        ; REGS: ALL
                        ; STK PRS: 7 + COND
 0A91 0A                DTRN2:  LDAX    B       ;A=MT,MF,SK
 0A92 F5                        PUSH    PSW     ;SAVE MT,MF,SK
 0A93 C5                        PUSH    B       ;SAVE ADR(MT,MF,SK)
 0A94 F61F                      ORI     1FH     ;INCLUDE MASK FOR COMMAND
 0A96 A5                        ANA     L       ;MASKMT,MF,SK,COMMAND
 0A97 02                        STAX    B       ;STORE COMMAND IN IOPB
 0A98 3E3F                      MVI     A,3FH   ;MASK FOR NO. OF BYTES
 0A9A A4                        ANA     H       ;MASK OUT RD/WR BITS
 0A9B 5F                        MOV     E,A     ;NO. OF BYTES
 0A9C CD030B            DT050:  CALL    CMNDS   ;COMMAND SERIAL OPERATION
 0A9F DAA50A                    JC      DT060   ;QUIT IF ERROR
 0AA2 C29C0A                    JNZ     DT050   ;WAIT IF FDC BUSY
 0AA5 C1                DT060:  POP     B       ;RESTORE ADR(MT,MF,SK)
 0AA6 D1                        POP     D       ;RESTORE MT,MF,SK
 0AA7 7A                        MOV     A,D     ;WHILE PRESERVING
 0AA8 02                        STAX    B       ; CARRY FLAG
 0AA9 C31E0A                    JMP     RSTOR   ;NORMAL RETURN
                        ;*****************************************************
                        ;AUXILIARY PORT RESET-A-BIT SUBROUTINE
                        ;
                        ;       CALL    AUXRST
                        ;       BC=ADR (CONTROL BYTE)
                        ;       E=CONTROL BIT (0 TO 3),REMAINING BITS (4 TO 7)
                        ;                                 MUST =0
                        ;
                        ;       RETURN W/CARRY=0
                        ;
                        ; REGS: F
                        ; STK PRS: 4
 0AAC CD190A            AUXRST: CALL    SAVER   ;SAVE REGISTERS
 0AAF CDC80A                    CALL    SLECT   ;SELECT CONTROL LINE
 0AB2 2F                        CMA
 0AB3 5F                        MOV     E,A     ;RESETS
 0AB4 0A                        LDAX    B       ; SELECTED
 0AB5 A3                        ANA     E       ;  BIT
 0AB6 C3C20A                    JMP     EXRTN   ;RETURN
                        ;*****************************************************
                        ;AUXILIARY PORT SET-A-BIT SUBROUTINE
                        ;
                        ;       CALL    AUXSET
                        ;       BC=ADR (CONTROL BYTE)
                        ;       E=CONTROL BIT (0 TO 3),REMAINING BITS (4 TO 7)
                        ;                                 MUST=0
                        ;
                        ;       RETURN W/CARRY=0
                        ;
                        ; REGS: F
                        ; STK PRS: 4
 0AB9 CD190A            AUXSET: CALL    SAVER   ;SAVE REGISTERS
 0ABC CDC80A                    CALL    SLECT   ;SELECT CONTROL LINE
 0ABF 5F                        MOV     E,A     ;SAVE MASK
 0AC0 0A                        LDAX    B       ;GET CONTROL BYTE
 0AC1 B3                        ORA     E       ;SET SELECTED BIT
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0          MODULE   PAGE   17

 LOC  OBJ           SOURCE STATEMENT

 0AC2 02        EXRTN:  STAX    B       ;UPDATE CONTROL BYTE
 0AC3 D312              OUT     AUXP    ;SEND COMMAND
 0AC5 C31E0A            JMP     RSTOR   ;RESTORE REGISTERS
                ;
 0AC8 1C        SLECT:  INR     E       ;1 TO 4
 0AC9 3E80              MVI     A,80H
 0ACB 07        SL010:  RLC
 0ACC 1D                DCR     E       ;CORRECT UNIT?
 0ACD C2CB0A            JNZ     SL010   ;NO
 0AD0 C9                RET
                ;*****************************************************
                ;AUXILIARY PORT ADDRESS BITS (14H TO 17H)
                ;
                ;       BC=ADR (CONTROL BYTE)
                ;       E=PAGE NO. IN HI NIBBLE (BITS 4 TO 7),
                ;         LO NIBBLE (BITS 0 TO 3) DON'T CARE
                ;
                ;       RETURN W/CARRY=0
                ;
                ;REGS: F
                ;STK PRS: 4
 0AD1 CD190A    AUXADR: CALL    SAVER   ;SAVE REGISTERS
 0AD4 7B                MOV     A,E     ;CLEAR
 0AD5 E6F0              ANI     0F0H    ; LO
 0AD7 5F                MOV     E,A     ;  NIBBLE
 0AD8 0A                LDAX    B       ;GET CONTROL BYTE
 0AD9 E60F              ANI     0FH     ;MASK PAGE BITS
 0ADB B3                ORA     E       ;PUT PAGE INTO CONTROL BYTE
 0ADC C3C20A            JMP     EXRTN   ;RETURN
                ;*****************************************************
                ; LEVEL 1 ROUTINES
                ;       USER CALLABLE THRU JUMP TABLE
                ;       USE ALL REGISTERS
                ;
                ;
                ; DMA SET ROUTINE
                ;       BC=COUNT
                ;       DE=ADDRESS
                ;       CALL    DMARD OR DMAWR
                ;
                ; REGS: AF,B
                ; STK PRS: 0
 0ADF 3E80      DMARD:  MVI     A,RD    ;TURN ON 8237 RD BIT
 0AE1 C3E60A            JMP     DMAST
 0AE4 3E40      DMAWR:  MVI     A,WR    ;TURN ON 8237 WR BIT
 0AE6 F3        DMAST:  DI              ;DISABLE INTERRUPTS WHILE
                                        ;PROGRAMMING THE 8237
 0AE7 D5                PUSH    D       ;SAVE ADDRESS
 0AE8 1F                RAR             ;POSITION
 0AE9 1F                RAR             ; RD
 0AEA 1F                RAR             ;  WR
 0AEB 1F                RAR             ;   BIT
 0AEC 1640              MVI     D,DMAMD ;DMA MODE WORD
 0AEE B2                ORA     D       ;OR RD/WR BIT WITH MODE WORD
 0AEF D30B              OUT     DMODE   ;SET MODE
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0         MODULE   PAGE   18

 LOC  OBJ          SOURCE STATEMENT

 0AF1 D1                 POP     D         ;RESTORE ADDRESS
 0AF2 79                 MOV     A,C
 0AF3 D301               OUT     DMWCO     ;PROGRAM LSB OF COUNT
 0AF5 78                 MOV     A,B
 0AF6 D301               OUT     DMWCO     ;PROGRAM MSB OF COUNT
 0AF8 7B                 MOV     A,E
 0AF9 D300               OUT     DMARO     ;PROGRAM LSB OF ADDRESS
 0AFB 7A                 MOV     A,D
 0AFC D300               OUT     DMARO     ;PROGRAM MSB OF ADDRESS
 0AFE AF                 XRA     A         ;DMAC MASK VALUE
 0AFF D30A               OUT     DMKSR     ;ENABLE DMA TRANSFER
 0B01 FB                 EI                ;RE-ENABLE INTERRUPTS
 0B02 C9                 RET
                   ;*****************************************************
                   ; COMMAND PHASE ROUTINE
                   ;
                   ; CALLING SEQUENCE
                   ;         BC=ADR(IOPB)
                   ;         E=# OF BYTES IN COMMAND
                   ;
                   ;         CALL     CMNDS    ;COMMAND SERIAL OPERATION
                   ;           OR
                   ;         CALL     CMNDP    ;COMMAND PARALLEL OPERATION
                   ;           OR
                   ;         CALL     CMND     ;UNCHECKED COMMAND OUTPUT
                   ;
                   ;         ERROR RETURN, CARRY=1 (C)
                   ;         BUSY RETURN, ZERO FLAG=0 (NZ). CARRY=0 (NC)
                   ;                 REGS BC, E PRESERVED FOR WAIT LOOPING
                   ;         NORMAL RETURN, ZERO FLAG=1 (Z). CARRY=0 (NC)
                   ;
                   ; NOTE:  THE 8272 FDC IS EITHER IN THE READ/WRITE MODE OR
                   ;        THE SEEK MODE, AND THESE TWO MODES ARE MUTUALLY
                   ;        EXCLUSIVE.
                   ;
                   ; REGS: ALL
                   ; STK PRS: 4+CONO
                   ;
                   ; COMMAND SERIAL OPERATIONS
                   ;         I.E.    COMMANDS THAT OPERATE IN THE READ/WRITE
                   ;                 MODE  OF THE  8272 AND/OR COMMANDS THAT
                   ;                 MUST CHECK FOR FDC BUSY AND FOR ANY FDD
                   ;                 SEEKING
                   ;         E.G.    READ  DATA, READ  DELETED  DATA, WRITE
                   ;                 DATA, WRITE DELETED DATA, READ A TRACK,
                   ;                 SCAN EQUAL, SCAN LOW OR EQUAL, SCAN HIGH
                   ;                 OR EQUAL, READ ID, FORMAT A TRACK, AND
                   ;                 SPECIFY
                   ;
 0B03 DB10         CMNDS:  IN      FDCST     ;GET MAIN FDC STATUS
 0B05 E61F                 ANI     1FH       ;FDC BUSY OR FDC IN SEEK MODE?
 0B07 C0                   RNZ               ;YES, RETURN W/ZERO FLAG=0, AND
                                             ; CARRY=0
 0B08 C3210B               JMP     CMND      ;NO, START COMMAND OUTPUT
                   ;
```

ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0   MODULE   PAGE   19

```
 LOC  OBJ        LINE        SOURCE STATEMENT

                       ; COMMAND PARALLEL OPERATIONS
                       ;        I.E.     COMMANDS THAT OPERATE IN THE SEEK MODE
                       ;                 OF  THE 8272 AND/OR COMMANDS THAT MUST
                       ;                 CHECK FOR  FDC BUSY  AND FOR SPECIFIED
                       ;                 FDD SEEKING
                       ;        E.G.     SEEK, RECALIBRATE, SENSE DRIVE STATUS
                       ;
0B0B DB10              CMNDP:   IN       FDCST    ;GET MAIN FDC STATUS
0B0D 6F                         MOV      L,A      ;SAVE FDC STATUS
0B0E E610                       ANI      10H      ;FDC BUSY? (I.E. IS FDC IN
                                                  ; READ/WRITE MODE?)
0B10 C0                         RNZ               ;YES, RETURN W/ZERO FLAG=0, AND
                                                  ; CARRY=0
0B11 03                         INX      B        ;ADR (UNIT SELECT BYTE)
0B12 0A                         LDAX     B        ;A=HD,US BYTE
0B13 0B                         DCX      B        ;RESTORE POINTER
0B14 E603                       ANI      03H      ;A = US=0 TO 3
0B16 57                         MOV      D,A      ;D=US
0B17 14                         INR      D        ;D=1 TO 4
0B18 3E80                       MVI      A,80H
0B1A 07                CM010:   RLC               ;SHIFT MASK TO NEXT HIGHER UNIT
0B1B 15                         DCR      D        ;DONE?
0B1C C21A0B                     JNZ      CM010    ;NO, CONTINUE
0B1F A5                         ANA      L        ;YES, IS FDD SEEKING?
0B20 C0                         RNZ               ;YES, RETURN W/ZERO FLAG=0, AND
                                                  ; CARRY=0
                                                  ;NO, START COMMAND OUTPUT
                       ;
                       ; COMMANDS THAT DO NOT CHECK FOR FDC BUSY OR ANY
                       ; FDD SEEKING
                       ;        E.G.     SENSE INTERRUPT STATUS
                       ;
0B21 F3                CMND:    DI                ;DISABLE INTERRUPTS WHILE
                                                  ; PROGRAMMING THE 8272
0B22 CD5B0B            CM020:   CALL     RDYC     ;IS FDC READY FOR COMMAND
0B25 D8                         RC                ;NO, ERROR, CARRY=1
0B26 0A                         LDAX     B        ;YES, A=BYTE FROM IOPB
0B27 D311                       OUT      FDCDT    ;SEND BYTE TO FDC DATA PORT
0B29 03                         INX      B        ;BUMP POINTER
0B2A 1D                         DCR      E        ;DONE?
0B2B C2220B                     JNZ      CM020    ;NO, CONTINUE
0B2E FB                         EI                ;YES, RE-ENABLE INTERRUPTS
0B2F C9                         RET               ;NORMAL RETURN, CARRY=0, AND
                                                  ; ZERO FLAG=1
                       ;***************************************************
                       ; RESULT PHASE ROUTINE
                       ;
                       ;  CALLING SEQUENCE
                       ;        CALL     RSULT
                       ;
                       ;        NORMAL RETURN, CARRY=0
                       ;                 C=NO. OF BYTES FOUND
                       ;                 RESULT BYTES STORED IN BUFFER
                       ;        ERROR RETURN, CARRY=1
                       ;
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   21

  LOC  OBJ          LINE        SOURCE STATEMENT

                    1100 ; REGS: AF,HL
                    1101 ; STK PRS: 2+CONO
  0B72 C5           1102 MESSG:  PUSH     B        ;SAVE BC
  0B73 7E           1103 MS010:  MOV      A,M
  0B74 B7           1104         ORA      A        ;END OF MESSAGE?
  0B75 CA800B       1105         JZ       MS020    ;YES
  0B78 4F           1106         MOV      C,A      ;NO, OUTPUT NEXT CHAR
  0B79 CD5008       1107         CALL     CONO     ;CONSOLE OUTPUT
  0B7C 23           1108         INX      H
  0B7D C3730B       1109         JMP      MS010    ;CONTINUE
  0B80 C1           1110 MS020:  POP      B        ;RESTORE BC
  0B81 C9           1111         RET
                    1112 ;
                    1113 ;
  0B82 46444320     1114 MSG10:  DB 'FDC ERR, DIO HI IN CMND PHASE',0DH,0AH,0
  0B86 4552522C
  0B8A 2044494F
  0B8E 20484920
  0B92 494E2043
  0B96 4D4E4420
  0B9A 50484153
  0B9E 45
  0B9F 0D
  0BA0 0A
  0BA1 00
  0BA2 46444320     1115 MSG20:  DB 'FDC ERR, DIO LO IN RSULT PHASE',0DH,0AH,0
  0BA6 4552522C
  0BAA 2044494F
  0BAE 204C4F20
  0BB2 494E2052
  0BB6 53554C54
  0BBA 20504841
  0BBE 5345
  0BC0 0D
  0BC1 0A
  0BC2 00
  0BC3 46444320     1116 MSG30:  DB 'FDC SEEK/ATTN INT',0,0AH,0
  0BC7 5345454B
  0BCB 2F415454
  0BCF 4E20494E
  0BD3 54
  0BD4 00
  0BD5 0A
  0BD6 00
  0BD7 46444320     1117 MSG40:  DB 'FDC I/O INT',0,0AH,0
  0BDB 492F4F20
  0BDF 494E54
  0BE2 00
  0BE3 0A
  0BE4 00
  0BE5 46444320     1118 MSG50:  DB 'FDC SEEK ERR',0,0AH,0
  0BE9 5345454B
  0BED 20455252
  0BF1 00
  0BF2 0A
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0        MODULE   PAGE   20

 LOC  OBJ          LINE        SOURCE STATEMENT

                   1045 ; REGS: AF,BC,HL
                   1046 ; STK PRS: 3+CONO
 0B30 2A4E08       1047 RSULT:  LHLD    ARSBF   ;HL=ADR (RESULT BUFFER)
 0B33 0E00         1048         MVI     C,0     ;INITIALIZE BYTE COUNT
 0B35 3A5308       1049 RS010:  LDA     DELAY   ;ALLOW 8272 TIME
 0B38 3D           1050 RS015:  DCR     A       ; TO CHANGE
 0B39 C2380B       1051         JNZ     RS015   ;  FDC STATUS
 0B3C DB10         1052 RS017:  IN      FDCST   ;A=FDC STATUS
 0B3E 47           1053         MOV     B,A     ;SAVE FDC STATUS
 0B3F E610         1054         ANI     10H     ;MORE RESULT BYTES? STILL BUSY?
 0B41 C8           1055         RZ              ;NO, NORMAL RETURN
                   1056                         ;NOTE: CARRY=0 FROM "ANI"
 0B42 78           1057         MOV     A,B     ;YES, RESTORE STATUS
 0B43 07           1058         RLC             ;RQM (READY) HIGH?
 0B44 D23C0B       1059         JNC     RS017   ;NO, KEEP WAITING
 0B47 07           1060         RLC             ;YES, DIO=OUTPUT?
 0B48 DA530B       1061         JC      RS020   ;YES
 0B4B 21A20B       1062         LXI     H,MSG20 ;PRINT OUT "DIO LO IN RESULT    #
 0B4E CD720B       1063         CALL    MESSG   ; PHASE" ERROR MESSAGE          #
 0B51 37           1064         STC             ;SET CARRY TO
 0B52 C9           1065         RET             ; INDICATE ERROR
 0B53 DB11         1066 RS020:  IN      FDCDT   ;GET RESULT BYTE FROM FLOPPY
 0B55 77           1067         MOV     M,A     ;STORE BYTE IN MEMORY
 0B56 23           1068         INX     H       ;BUMP POINTER
 0B57 0C           1069         INR     C       ;BUMP COUNT
 0B58 C3350B       1070         JMP     RS010   ;GO BACK & CHECK FOR MORE BYTES
                   1071 ;*****************************************************
                   1072 ; READY FOR COMMAND SUBROUTINE
                   1073 ;
                   1074 ; CALLING SEQUENCE
                   1075 ;        CALL    RDYC
                   1076 ;
                   1077 ;        NORMAL RETURN, CARRY=0
                   1078 ;        ERROR RETURN, CARRY=1
                   1079 ;
                   1080 ; REGS: AF,HL
                   1081 ; STK PRS: 3+CONO
 0B5B 3A5308       1082 RDYC:   LDA     DELAY   ;ALLOW 8272 TIME
 0B5E 3D           1083 RY010:  DCR     A       ; TO CHANGE
 0B5F C25E0B       1084         JNZ     RY010   ;  FDC STATUS
 0B62 DB10         1085 RY020:  IN      FDCST   ;GET FDC STATUS
 0B64 07           1086         RLC             ;IS RQM (READY) HIGH?
 0B65 D2620B       1087         JNC     RY020   ;NO, WAIT UNTIL IT IS
 0B68 07           1088         RLC             ;YES, DIO=INPUT?
 0B69 D0           1089         RNC             ;YES, FDC READY FOR COMMAND
 0B6A 21820B       1090         LXI     H,MSG10 ;NO, ERROR                      #
 0B6D CD720B       1091         CALL    MESSG   ;"DIO HIGH"                     #
 0B70 37           1092         STC             ;SET CARRY TO
 0B71 C9           1093         RET             ; INDICATE ERROR
                   1094 ;
                   1095 ; MESSAGE SUBROUTINE
                   1096 ;        (FROM THIS POINT TO THE END OF THE PROGRAM CAN
                   1097 ;        BE ELIMINATED WHEN ERROR MESSAGES ARE NO LONGER
                   1098 ;        REQUIRED)
                   1099 ;        HL=ADR (MESSAGE),NOTE: LAST BYTE MUST EQUAL ZERO
```

```
ISIS-II 8080/8085 MACRO ASSEMBLER, V3.0          MODULE    PAGE    22

  LOC  OBJ          LINE        SOURCE STATEMENT

  0BF3 00
  0BF4 46444320     1119 MSG60:  DB 'FDC I/O ERR',0,0AH,0
  0BF8 492F4F20
  0BFC 455252
  0BFF 00
  0C00 0A
  0C01 00
                    1120 END

PUBLIC SYMBOLS
AUXADR A 0AD1   AUXP   A 0012   AUXRST A 0AAC   AUXSET A 0AB9   BASE   A 0000   DCLFL  A 000C   DMAR0  A 0000
DMAR1  A 0002   DMAR2  A 0004   DMAR3  A 0006   DMASK  A 000F   DMCLR  A 000D   DMKSR  A 000A   DMODE  A 000B
DMRQ   A 0009   DMSR   A 0008   DMTR   A 000E   DMWC0  A 0001   DMWC1  A 0003   DMWC2  A 0005   DMWC3  A 0007
FDCDT  A 0011   FDCST  A 0010   FRMTK  A 096E   INIT   A 0918   INT20  A 08D1   PRSLT  A 08E1   RDDD   A 0933
RDID   A 0965   RDTRK  A 094E   READ   A 092A   RECAL  A 09A3   SCNEQ  A 0988   SCNHE  A 099A   SCNLE  A 0991
SEEK   A 09BA   SEGHI  A 0015   SEGLO  A 0014   SFTRS  A 0013   SNSDS  A 09EA   SNSIS  A 0A04   SPCFY  A 09D3
WRITE  A 093C   WRTDD  A 0945

EXTERNAL SYMBOLS


USER SYMBOLS
ARSBF  A 084E   AUXADR A 0AD1   AUXP   A 0012   AUXRST A 0AAC   AUXSET A 0AB9   BASE   A 0000   CMND   A 0B21
CMNDP  A 0B0B   CMNDS  A 0B03   CM010  A 0B1A   CM020  A 0B22   CON0   A 0850   DCLFL  A 000C   DELAY  A 0853
DMAMD  A 0040   DMAR0  A 0000   DMAR1  A 0002   DMAR2  A 0004   DMAR3  A 0006   DMARD  A 0ADF   DMASK  A 000F
DMAST  A 0AE6   DMAWR  A 0AE4   DMCLR  A 000D   DMKSR  A 000A   DMODE  A 000B   DMRQ   A 0009   DMSR   A 0008
DMTR   A 000E   DMWC0  A 0001   DMWC1  A 0003   DMWC2  A 0005   DMWC3  A 0007   DT005  A 0A38   DT010  A 0A41
DT015  A 0A44   DT020  A 0A55   DT030  A 0A58   DT040  A 0A60   DT042  A 0A63   DT045  A 0A6B   DT050  A 0A9C
DT060  A 0AA5   DTRN1  A 0A2D   DTRN2  A 0A91   EOIC   A 0020   EXRTN  A 0AC2   FDCDT  A 0011   FDCST  A 0010
FRMTK  A 096E   ICCP   A 00DA   INI0   A 091E   INIT   A 0918   INT10  A 0855   INT20  A 08D1   IT002  A 0865
IT008  A 0895   IT010  A 08B1   MESSG  A 0B72   MSG10  A 0B82   MSG20  A 0BA2   MSG30  A 0BC3   MSG40  A 0BD7
MSG50  A 0BE5   MSG60  A 0BF4   MS010  A 0B73   MS020  A 0B80   PR005  A 08E6   PR010  A 0904   PR020  A 0908
PR030  A 0913   PRSLT  A 08E1   RD     A 0080   RDDD   A 0933   RDID   A 0965   RDTRK  A 094E   RDYC   A 0B5B
READ   A 092A   RECAL  A 09A3   REGF   A 7F30   RE010  A 09AE   RSBDH  A 0A29   RS010  A 0B35   RS015  A 0B38
RS017  A 0B3C   RS020  A 0B53   RSTC   A 0A27   RSTOR  A 0A1E   RSULT  A 0B30   RY010  A 0B5E   RY020  A 0B62
SAVER  A 0A19   SCNEQ  A 0988   SCNHE  A 099A   SCNLE  A 0991   SD010  A 09F5   SEEK   A 09BA   SEGHI  A 0015
SEGLO  A 0014   SFTRS  A 0013   SK010  A 09C6   SK020  A 09CB   SLECT  A 0AC8   SL010  A 0ACB   SNSDS  A 09EA
SNSIC  A 0A18   SNSIS  A 0A04   SPC10  A 09DE   SPCFY  A 09D3   USTACK A 7F80   VER    A 0010   VERSIO A 0854
WR     A 0040   WRITE  A 093C   WRTDD  A 0945

ASSEMBLY COMPLETE,    NO ERRORS
```

iSBX™ MULTIMODULE™ BOARD INTERFACE

B-1. INTRODUCTION

The iSBC 208 Controller is designed to accept a single- or double-wide iSBX Multimodule board. The iSBX Multimodule board installed on the controller is accessed by the host processor directly through the Multibus interface; the controller dedicates two of its DMA channels (DMAC channels 2 and 3) to the iSBX board to provide direct memory access between the iSBX board and system memory. The physical interface between the parent iSBC 208 Controller and the installed iSBX Multimodule board is provided through a 36-pin connector. For specific information on an individual iSBX board, refer to the corresponding iSBX Multimodule board hardware reference manual.

B-2. INSTALLATION

Physical installation of the selected iSBX Multimodule board on the controller is described in the corresponding iSBX Multimodule board hardware reference manual. Table B-1 defines the iSBX Multimodule board signals on the controller's J3 connector.

B-3. CONFIGURATION

As noted in table B-1, the iSBC 208 Controller includes a number of jumpers that must be installed to enable the corresponding signals on the controller's J3 connector. Note that none of the jumpers are installed at the factory. The following subsections define the jumper functions for the Multimodule board DMA channels and system interrupts.

B-4. DMA Channels

Channel 2 of the DMAC is reserved exclusively for the iSBX Multimodule board; the channel 2 DREQ (DMA Request) and DACK (DMA Acknowledge) signals are permanently routed to the J3 connector, and no jumpers are required. If an end-of-process (EOP) signal is required by the iSBX Multimodule board to indicate when the channel 2 DMA transfer is complete (i.e., DMAC channel 2 word count register decrements to zero), a jumper must be installed between jumper post E15 (OEOP) and either jumper post E16 (OPT0 signal line on J3-30) or jumper post E14 (OPT1 signal line on J3-28).

Channel 3 of the DMAC is available to the iSBX Multimodule board; the channel 3 DACK and DREQ signals must be jumpered on the controller to route

Table B-1. J3 Connector Pin Assignments

Pin	Signal	Function	Pin	Signal	Function
1	+12V	+12 volts	19	MD7	Multimodule Data Bit 7
2	−12V	−12 volts	20	MCS1/	Multimodule Chip Select 1
3	Gnd	Logic Ground	21	MD6	Multimodule Data Bit 6
4	+5V	+5 volts	22	MCS0/	Multimodule Chip Select 0
5	MRESET	Multimodule Reset	23	MD5	Multimodule Data Bit 5
6	MCLK	Multimodule Clock	24	Reserved	
7	MA2	Multimodule Address Bit 2	25	MD4	Multimodule Data Bit 4
8	MPST/	Multimodule Present	26	TDMA*	Terminate DMA
9	MA1	Multimodule Address Bit 1	27	MD3	Multimodule Data Bit 3
10	Reserved		28	OPT1*	Optional Signal 1
11	MA0	Multimodule Address Bit 0	29	MD2	Multimodule Data Bit 2
12	MINTR1*	Multimodule Interrupt 1	30	OPT0*	Optional Signal 0
13	IOWRT/	I/O Write	31	MD1	Multimodule Data Bit 1
14	MINTR0*	Multimodule Interrupt 0	32	MDACK/	Multimodule DMA Ack.
15	IORD/	I/O Read	33	MD0	Multimodule Data Bit 0
16	MWAIT/	Multimodule Wait	34	MDRQT	Multimodule DMA Request
17	Gnd	Logic Ground	35	Gnd	Logic Ground
18	+5V	+5 volts	36	+5V	+5 volts

*Signal requires jumper connection on controller board.

the signals to the J3 connector. The DACK3 signal (ODACK) appears on jumper post E13, and the DREQ3 signal (ODREQ) appears on jumper post E12. These two signals must be connected to jumper posts E16 (OPT0 signal line on J3-30) and E14 (OPT1 signal line on J3-28).

NOTE

Since DMAC channel 3 requires the use of both the OPT0 and OPT1 optional signal lines, the EOP signal from the DMAC to the iSBX Multimodule board cannot be supported.

If the iSBX Multimodule board includes logic to externally terminate a DMA transfer, a jumper must be installed between jumper post E1 (TDMA signal on J3-26) to jumper post E3 (external EOP input to the DMAC).

B-5. INTERRUPTS

There are two interrupt signals available on iSBX Multimodule board interface connector J3, MINTR0 on J3-14 and MINTR1 on J3-12. These signals are routed to the controller's interrupt jumper matrix (jumper posts E80 and E81, respectively) and must be connected to the desired Multibus interface interrupt level according to the following table.

Interrupt Signal	Jumper Post	Jumper Post	Interrupt Level
MINTR0	E80	E89 E88 E87 E86	INT0/ INT1/ INT2/ INT3/
MINTR1	E81	E85 E84 E83 E82	INT4/ INT5/ INT6/ INT7/

Note that the interrupt level selected must *not* have been previously assigned to another bus master.

B-6. PROGRAMMING INFORMATION

When an iSBX Multimodule board is installed on the controller, host processor communication is accomplished through a set of 16 I/O ports. These 16 I/O ports reference the same I/O base address as the controller and are numbered port addresses 20 through 2F (hexadecimal). To address the additional I/O ports, a 6-bit I/O port address is required (the controller only requires a 5-bit port address); the I/O base address must be located on a 64-port boundary, and I/O base address bit 5 is irrelevant.

B-7. PORT ASSIGNMENTS

The 16 I/O ports assigned to the iSBX Multimodule board are divided into two groups of eight ports by the two Multimodule chip select signals (MCS0/ and MCS1/). The individual port addressed within the group is determined by Multimodule address bits 0 through 2 (MA0-MA2). Table B-2 defines the iSBX Multimodule port assignments; refer to the corresponding iSBX Multimodule board hardware reference manual for the specific I/O port functions.

Table B-2. I/O Port Assignments

I/O Port Address (Hexadecimal)	iSBX Board Signal Levels				
	MCS1/	MCS0/	MA2	MA1	MA0
20	1	0	0	0	0
21			0	0	1
22			0	1	0
23			0	1	1
24			1	0	0
25			1	0	1
26			1	1	0
27			1	1	1
28	0	1	0	0	0
29			0	0	1
2A			0	1	0
2B			0	1	1
2C			1	0	0
2D			1	0	1
2E			1	1	0
2F			1	1	1

B-8. PROGRAMMING THE DMAC

Programming the DMAC is described in sections 3-3 through 3-12 of this manual.

APPENDIX C
DRIVE INTERFACES

C1. INTRODUCTION

The following tables (tables C-1 and C-2) define specific drive interfaces for a number of standard- and mini-sized drives that are compatible with the iSBC 208 controller. In the tables, a drive interface pin number appearing in an individual drive column indicates that the signal function and pin assignment on the controller interface connector are the same on the drive interface connector.

C-2. USING THE TABLES

As an example of how the tables are used, assume that four Micropolis 1015 mini drives are to be interfaced to the controller. Referring to the Micropolis 1015 column in table C-2, note that a (common) HEAD LOAD signal is required on pin 2, a MOTOR ON signal is required on pin 16, the drive select signal for the fourth drive is required on pin 34, and that all of the remaining interface signals are directly pin-to-pin compatible.

Referring to the controller schematic in Chapter 5, to configure the controller to provide a HEAD LOAD signal on pin 2 of connector J1, the jumper between posts E31 and E32 (see sheet 7 of the schematic) is removed, and a jumper is installed between posts E32 (the source of the HEAD LOAD signal) and E10 (pin 2 of connector J1); see sheet 3 of the schematic. Again referrring to sheet 3 of the schematic, to provide a MOTOR ON signal on pin 16, a jumper is installed between the selected auxiliary port bit (see section 2-14) on jumper post E11, E9, E7, or E2, and jumper post E6 (pin 16 of connector J1). To provide a fourth drive select signal, the factory-installed jumper between posts E21 and E22 (TWO SIDED/) is removed, and a jumper is installed between post E20 (the controller's DRIVE SELECT 3/ signal on sheet 7) and post E21 (pin 34 of connector J1).

Table C-1. Standard 8-inch Drive Interface Pin Assignments

Controller Interface Connector J2		Shugart SA800/850	Calcomp 143M	CDC 9406-3	Memorex 550/552	MFE Series 500/700	Persci 70	Persci 288	Pertec 650	Pertec 5x4	Qume Data Trak 8	Remex 2000/4000	Siemens FDD 200-8 FDD 100-8
Signal Name	Pin												
**LOW CURRENT/	2	2[1]	HEAD LOAD 2	Unassigned	Unassigned	2	MOTOR ON	Unassigned	2	2	Unassigned	2	2
*HEAD LOAD 2/	4	Unassigned	HEAD LOAD 3	Unassigned	Unassigned	Unassigned	Unassigned	Unassigned	Unassigned	Unassigned	Unassigned	Unassigned	Unassigned
*HEAD LOAD 3/	6	Unassigned	HEAD LOAD 4	Unassigned	Unassigned	POWER SAVE	Unassigned	Unassigned	Unassigned	Unassigned	Unassigned	Unassigned	Unassigned
*User Defined	8	Unassigned	TRK 43	Unassigned	Unassigned	Unassigned	Unassigned	Unassigned	WRITE BUSY	Unassigned	Unassigned	Unassigned	Unassigned
**TWO SIDED/	10	10[1]	10	10	10[2]	10[3]	SEEK COMPLETE	10	10	Unassigned	10	10[4]	ILLEGAL[5] PACK
*User Defined	12	DISK CHANGE	DISK CHANGE	DISK CHANGE	DISK CHANGE	DISK CHANGE	RESTORE	DISK CHANGE	DISK CHANGE	Unassigned	DISK CHANGE	DISK CHANGE	Unassigned
SIDE SELECT/	14	14[1]	HEAD LOAD1	14	14[2]	14[3]	REMOTE EJECT	14	14	Unassigned	14	14[4]	14[5]
*User Defined	16	IN USE	IN USE	IN USE	IN USE	IN USE	POSITION PULSES	IN USE	BUSY CONTROL	Unassigned	IN USE	IN USE	IN USE
**HEAD LOAD 0/	18	18	18	18	18	18	18	18	18	18	18	18	18
INDEX/	20	20	20	20	20	20	20	20	20	20	20	20	20
**READY/	22	22	22	22	22	22	22	22	22	22	22	22	22
*HEAD LOAD 1/	24	SECTOR	SECTOR	SECTOR	SECTOR	SECTOR	SECTOR	SECTOR	SECTOR	SECTOR	Unassigned	SECTOR	SECTOR
DRIVE SELECT 0/	26	26	26	26	26	26	26	26	26	26	26	26	26
DRIVE SELECT/	28	28	28	28	28	28	28	28	28	28	28	28	28
DRIVE SELECT 2/	30	30	30	30	30	30	30	30	30	30	30	30	30
DRIVE SELECT 3/	32	32	32	32	32	32	32	32	32	32	32	32	32
DIRECTION/	34	34	34	34	34	34	34	34	34	34	34	34	34
STEP/	36	36	36	36	36	36	36	36	36	36	36	36	36
WRITE DATA/	38	38	38	38	38	38	38	38	38	38	38	38	38
WRITE GATE/	40	40	40	40	40	40	40	40	40	40	40	40	40
TRACK 0/	42	42	42	42	42	42	42	42	42	42	42	42	42
WRITE PROTECT/	44	44	44	44	44	44	44	44	44	44	44	44	44
READ DATA/ 46	46	46	46	46	46	46	46	46	46	46	46	46	46
*FAULT/	48	SEPARATED DATA	SEPARATED DATA	SEPARATED DATA	SEPARATED DATA	SEPARATED DATA	SEPARATED DATA	SEPARATED DATA	SEPARATED DATA	SEPARATED DATA	Unassigned	SEPARATED DATA	SEPARATED DATA
*FAULT RESET/	50	SEPARATED CLOCK	SEPARATED CLOCK	SEPARATED CLOCK	SEPARATED CLOCK	SEPARATED CLOCK	SEPARATED CLOCK	SEPARATED CLOCK	SEPARATED CLOCK	SEPARATED CLOCK	Unassigned	SEPARATED CLOCK	SEPARATED CLOCK

*Requires jumper on controller
**Jumper is installed on controller

[1] 850 Only [2] 552 Only [3] 700 Only [4] 4000 Only [5] 200-8 Only

Table C-2. Mini Drive Interface Pin Assignments

Controller Interface Connector J1		Shugart SA400/450	Shugart SA410/460	BASF 6106/6108	CDC 9409	Micropolis 1015	MPI 51/52	Pertec FD200	Pertec FD250	Siemens FDD 200-5N FDD 100-5B	Tandon TM100
Signal Name	Pin										
*User Defined	2	Unassigned	Unassgined	HEAD LOAD	Unassigned	HEAD LOAD	Unassigned	Unassigned	Unassigned	Unassigned	Unassigned
*User Defined	4	IN USE[1]	IN USE	Unassigned	Unassigned	Unassigned	IN USE	Unassigned	BUSY CONTROL	IN USE[4]	Unassigned
**READY/	6	Unassigned	DRIVE SELECT 4	6	DRIVE SELECT 4	6	DRIVE SELECT 4	DRIVE SELECT 3	DRIVE SELECT 3	DRIVE SELECT 3	DRIVE SELECT 3
INDEX/	8	8	8	8	8	8	8	8	8	8	8
DRIVE SELECT 0/	10	10	10	10	10	10	10	10	10	10	10
DRIVE SELECT 1/	12	12	12	12	12	12	12	12	12	12	12
DRIVE SELECT 2/	14	14	14	14	14	14	14	14	14	14	14
*User Defined	16	MOTOR ON	MOTOR ON	MOTOR ON	MOTOR ON	MOTOR ON	MOTOR ON	MOTOR ON	MOTOR ON	MOTOR ON	MOTOR ON
DIRECTION/	18	18	18	18	18	18	18	18	18	18	18
STEP/	20	20	20	20	20	20	20	20	20	20	20
WRITE DATA/	22	22	22	22	22	22	22	22	22	22	22
WRITE ENABLE/	24	24	24	24	24	24	24	24	24	24	24
TRACK 0	26	26	26	26	26	26	26	26	26	26	26
WRITE PROTECT/	28	28	28	28	28	28	28	28	28	28	28
READ DATA/	30	30	30	30	30	30	30	30	30	30	30
SIDE SELECT/	32	32[1]	32[2]	32[3]	32	32	32	Unassigned	32	32[4]	32
**TWO SIDED/	34	Unassigned	DOOR OPEN	IN USE	Unassigned	DRIVE SELECT 4	Unassigned	Unassigned	Unassigned	Unassigned	Unassigned

*Requires jumper on controller
**Jumper is installed on controller

[1] 450 Only
[2] 460 Only
[3] 6108 Only
[4] 200-5N Only