iSBC 204™
FLEXIBLE DISKETTE CONTROLLER HARDWARE REFERENCE MANUAL

Manual Order Number: 9800568-02

Intel Corporation, 3065 Bowers Avenue, Santa Clara, California 95051

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

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

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

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

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

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

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

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

PREFACE

This manual provides information regarding the installation, programming, operation, and service of the iSBC 204 Flexible Diskette Controller. Supplementary information is provided in the following Intel publications.

- *Intel MULTIBUS Specification Manual,* Order No. 9800683.
- *MCS-80 User's Manual,* Order No. 9800153
- *MCS-86 User's Manual,* Order No. 9800722
- *Intel 8080/8085 Assembly Language Programming Manual,* Order No. 9800301
- *Intel MCS-86 Assembly Language Reference Manual,* Order No. 9800640

CONTENTS

TABLES

ILLUSTRATIONS

CHAPTER 1
GENERAL INFORMATION

1-1. SCOPE

This manual is the hardware reference for the iSBC (Intel Single Board Computer) Model 204 Flexible Diskette Controller. Note that to avoid any ambiguity with the individual Intel "controller" integrated circuits incorporated in the logic design of the controller, the iSBC 204 Flexible Diskette Controller is referred to as the "Interface" in the remainder of this manual. The manual itself is divided into five parts which describe general information, preparation for use, programming conventions, theory of operation, and service information, respectively.

1-2. INTRODUCTION

The iSBC 204 Interface is one product within a complete line of Intel iSBC System 80 and 86 expansion modules. The standard interface is capable of supporting either one double-sided or two single-sided, single density diskette drives available from a number of major drive manufacturers. Optionally, the interface can be expanded to support four single-sided or two double-sided drives. In addition to being fully compatible with the IBM 3740 data formats, the interface supports non-IBM sector lengths of up to 4096 data bytes per sector as well as the mini-sized diskette drive data formats.

The interface is designed expressly for Intel Multibus interface compatibility and can be inserted directly into a standard iSBC 604/614 Modular Cardcage/Backplane as found in the iSBC 80 Series mainframes or into any of the Intellec microcomputer development systems. All circuitry is contained completely on a single printed circuit board which operates from a single +5 volt source. A majority of the logic is LSI (large scale integration) and includes both an Intel 8257 DMA Controller circuit and an Intel 8271 Floppy Disk Controller (FDC) circuit. Additionally, data separation logic is included on the board to eliminate the necessity of separation circuitry within the drive.

The interface's wide range of drive compatibility has been achieved principally by permitting the drive

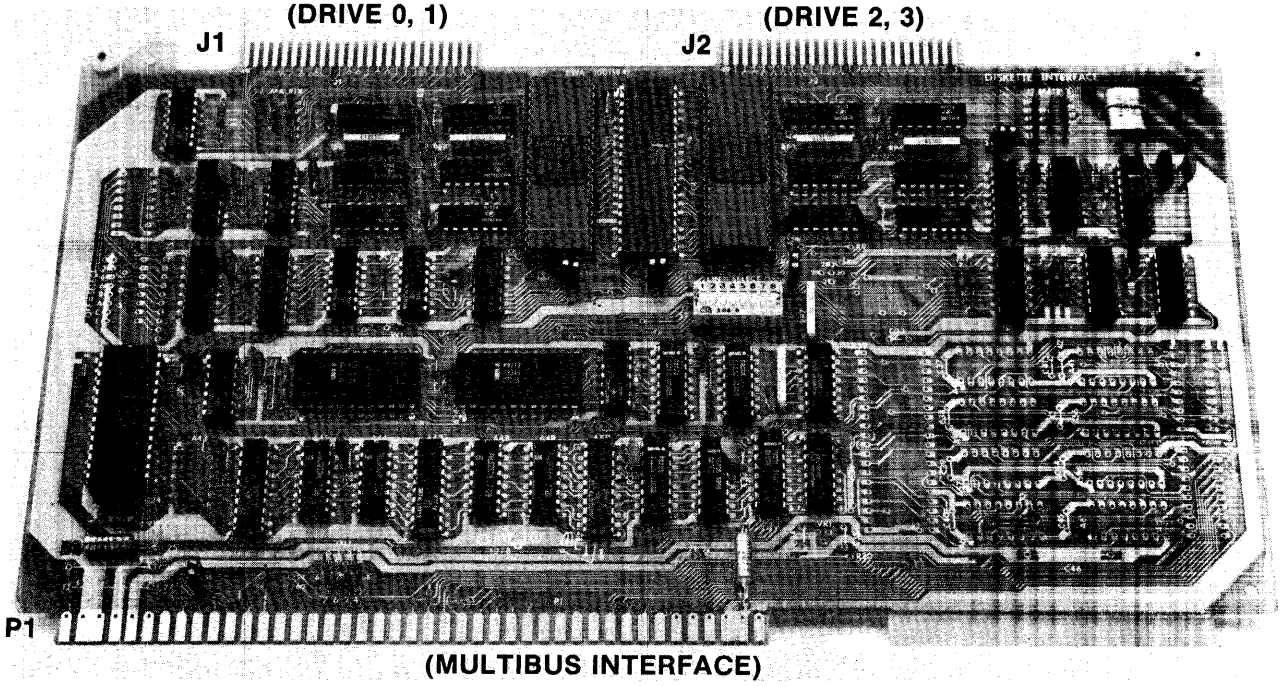


Figure 1-1. iSBC 204 Interface

operating characteristics to be program-specified. The head load, track-to-track access, and head settling times can be individually set to match the characteristics of the selected drive. This feature eliminates the timing compromises that would be expected for compatibility with the slower drives. Additionally, one optional input signal and one optional output signal are included at the drive I/O connector and can be used to support the Disk Change or Two Sided optional drive status indications or the Fault Reset function.

In addition to routine read and write capabilities involving either single or multiple sectors, the interface can read or write deleted data, can verify data previously written without effecting a transfer to memory and, with the special scan commands, can search specified tracks for a unique data pattern. Once all information required for an operation has been supplied to the interface by the executing program, further CPU involvement is unnecessary. The interface performs all required drive positioning and takes control of the bus when needed for data transfer.

1-3. SPECIFICATIONS

Table 1-1 lists the physical and performance characteristics of the iSBC 204 Interface.

Table 1-1. Specifications

COMPATIBILITY							
CPU	Any iSBC System 80 or 86 mainframe, Intellec microprocessor development system, or Multibus interface compatible CPU, capable of operating in a multimaster environment.						
Drive	Single density, standard- and mini-sized diskette drives. Two single-sided or one double-sided drive supported. Optionally expandable to support four single-sided or two double-sided drives.						
Diskette	Unformatted IBM Diskette 1 or equivalent (single-sided). Unformatted IBM Diskette 2 or equivalent (double-sided). Unformatted Shugart SA104 Diskette or equivalent (mini).						
DATA ORGANIZATION AND CAPACITY							
		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 Diskette			77			Up to 255	
Bytes Per Diskette (77 tracks)	256,256 (128-byte sector) 295,680 (256-byte sector) 315,392 (512-byte sector)					315,392	
DRIVE CHARACTERISTICS							
Transfer Rate	250 kilobits per second (standard) 125 kilobits per second (mini)						
Disk Speed	360 rpm (standard) 300 rpm (mini)						

Table 1-1. Specifications (Cont'd)

DRIVE CHARACTERISTICS (CONT'D)	
Track-to-Track Access Time	Programmable from 1 to 255 ms in 1 ms steps (standard) or from 2 to 510 ms in 2 ms steps (mini).
Head Settling Time	Programmable from 0 to 255 ms in 1 ms steps (standard) or from 0 to 510 ms in 2 ms steps (mini).
Head Load Time	Programmable from 0 to 60 ms in 4 ms steps (standard) or from 0 to 120 ms in 8 ms steps (mini).
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%), 2.5 amperes (maximum)
Environmental	
Temperature:	0°C to +55°C (operating) -55°C to +85°C (non-operating)
Humidity:	Up to 90% relative humidity without condensation (operating). All conditions without condensation or frost (non-operating).

CHAPTER 2
PREPARATION FOR USE

2-1. INTRODUCTION

This chapter presents information on the preparation for use and installation of the iSBC 204 Interface. Included within this chapter are instructions describing the unpacking and inspection, installation, board configuration, CPU bus connection, and drive cabling for the interface.

2-2. UNPACKING AND INSPECTION

On receipt of the interface 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 MCSD Technical Support Center (see Section 5-3) 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.

Item 1: iSBC 204 Interface Printed Circuit Assembly, Part Number 1001584.

Item 2: Schematic Diagram, Drawing Number 2003209.

Item 3: Assembly Diagram, Drawing Number 1001584.

Compare the packing 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 interface at a later date.

2-3. INSTALLATION CONSIDERATIONS

The interface is designed expressly for installation into the Intel iSBC 604/614 Modular Cardcage/Backplane as found in the iSBC 80 Series single board computer mainframes or into any odd-numbered slot in an Intellec microcomputer development system or any slot of an Intellec Series II microcomputer development system. The interface additionally can be installed into a user's Multibus interface compatible backplane assembly that meets the interface's mating connector dimensional requirements.

2-4. POWER REQUIREMENTS

The interface operates from a single +5 volt (±5%) source and requires a maximum of 2.5 amperes. When installing the interface in an iSBC 80 Series System Chassis, Intellec microcomputer development system, or custom system, ensure that the system's power supply can meet the additional current requirements of the interface.

2-5. COOLING REQUIREMENTS

The interface dissipates 178 gram-calories per minute (0.72 Btu/minute). The iSBC 80 Series System Chassis and Intellec microcomputer development systems use forced-air cooling which generally is adequate to maintain an internal operating temperature below 55°C. When installing the interface 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. MULTIBUS INTERFACE CONNECTOR

The interface communicates with the CPU (and memory) through the Multibus interface. Tables 2-1 and 2-2 define the Multibus interface pin assignments and corresponding signal definitions.

The interface connects to the Multibus interface through connector P1, and an 86-pin, double-sided printed circuit edge connector with 3.96mm (0.156 inch) contact centers.

Table 2-1. Multibus Pin Assignments

PIN	SIGNAL	FUNCTION	PIN	SIGNAL	FUNCTION
1	GND	Ground	44	ADRF/	Address Bus
2	GND		45	ADRC/	
3	+5 VDC	Power Input	46	ADRD/	
4	+5 VDC		47	ADRA/	
5	+5 VDC		48	ADRB/	
6	+5 VDC		49	ADR8/	
7	—	Reserved	50	ADR9/	
8	—		51	ADR6/	
9	—		52	ADR7/	
10	—		53	ADR4/	
11	GND	Ground	54	ADR5/	
12	GND		55	ADR2/	
13	BCLK/	Bus Clock	56	ADR3/	
14	INIT/	Initialization	57	ADR0/	
15	BPRN/	Bus Priority In	58	ADR1/	
16	BPRO/	Bus Priority Out	59	—	Reserved
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	—	Reserved	67	DAT6/	Data Bus
25	AACK/	Advanced Acknowledge	68	DAT7/	
26	—	Reserved	69	DAT4/	
27	—		70	DAT5/	
28	ADR10/	Address Bus	71	DAT2/	
29			72	DAT3/	
30	ADR11/		73	DAT0/	
31			74	DAT1/	
32	ADR12/		75	GND	Ground
33			76	GND	
34	ADR13/		77	—	Reserved
35	INT6/	Interrupt Request 6	78	—	
36	INT7/	Interrupt Request 7	79	—	
37	INT4/	Interrupt Request 4	80	—	
38	INT5/	Interrupt Request 5	81	+5 VDC	Power Input
39	INT2/	Interrupt Request 2	82	+5 VDC	
40	INT3/	Interrupt Request 3	83	+5 VDC	
41	INT0/	Interrupt Request 0	84	+5 VDC	
42	INT1/	Interrupt Request 1	85	GND	Ground
43	ADRE/	Address Bus	86	GND	

Table 2-2. Multibus Signal Definitions

SIGNAL	FUNCTION
AACK/	Advance Acknowledge. A special acknowledge signal originating from a random access memory (RAM) board and used as an advanced response to allow the interface to proceed with a read or write transfer without waiting for the normal transfer acknowledge (XACK/) signal.
ADR0/-ADRF/ ADR10/-ADR13/	Address. These 20 bidirectional lines specify the address of the memory location or I/O port to be accessed. ADR13/ is the most significant bit.
BCLK/	Bus Clock. This input signal is used to synchronize the interface'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 interface or another higher-priority master board has requested control of the bus.
BREQ/	Bus Request. This output signal is used with parallel priority resoultion schemes and indicates that the interface is requesting control of the bus.
BUSY/	Bus Busy. This bidirectional signal indicates that either the interface or another master board is currently in control of the bus and consequently prevents any other master board from gaining access to the bus.
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 an interface-initiated interrupt request.
IORC/	I/O Read Command. This input signal instructs the interface to place the data associated with the addressed input port onto the data lines.
IOWC/	I/O Write Command. This input signal instructs the interface 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 interface.
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 interface on the data lines is to be written into that location.
XACK/	Transfer Acknowledge. This signal originates from the interface during I/O port transfers and indicates that the interface 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 interface.

2-7. SIGNAL CHARACTERISTICS. The dc characteristics of the iSBC 204 Interface signals are provided in table 2-3. The ac characteristics of the iSBC 204 Interface when operating in the master mode and slave mode are provided in tables 2-4 and 2-5, respectively. Figures 2-1 and 2-2 show the Multibus interface command timing when the interface is operating as a master (Bus Acquisition and Memory Transfer Timing) and as a slave (I/O Transfer Timing).

2-8. BOARD LOCATION CONSIDERATIONS

When installing the interface in a serial priority environment (e.g., within any of the Intel Series 80 mainframes), the interface should occupy the highest priority slot (top physical slot) in the 604/614 Modular Cardcage/Backplane assembly, with any other bus masters and the CPU board located below. This high priority is necessary because the interface must complete a data transfer as fast as every 32 μsec, or data will be lost, when functioning as a bus master during DMA transfers. 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 figure 2-3, a wire-wrap jumper must be installed from terminal post B (BPRN/) to logic ground and 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.

Table 2-3. iSBC 204 Interface DC Characteristics

Signals	Symbol	Parameter Description	Test Conditions	Min.	Max.	Units
All Bus	V_{IL}	Input Low Voltage			0.8	V
Interface Inputs	V_{IH}	Input High Voltage		2.0		V
ADR0/-ADR1/	V_{OL}	Output Low Voltage	I_{OL} = 50 mA		0.6	V
	V_{OH}	Output High Voltage	I_{OH} = -10 mA	2.4		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45V		-0.6	mA
	I_{IH}	Input Current at High V	V_{IN} = 5.25V		0.12	mA
	*C_L	Capacitive load			25	pf
ADR2/-ADR3/	V_{OL}	Output Low Voltage	I_{OL} = 50 mA		0.6	V
	V_{OH}	Output High Voltage	I_{OH} = -10 mA	2.4		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45V		-0.85	mA
	I_{IH}	Input Current at High V	V_{IN} = 5.25V		0.13	mA
	*C_L	Capacitive load			30	pf
ADR4/-ADRB/	V_{OL}	Output Low Voltage	I_{OL} = 24 mA		0.5	V
	V_{OH}	Output High Voltage	I_{OH} = -14 mA	2.4		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V		-1.2	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.7V		80	μA
	*C_L	Capacitive load			15	pf
ADRC/-ADR13/	V_{OL}	Output Low Voltage	I_{OL} = 24 mA		0.5	V
	V_{OH}	Output High Voltage	I_{OH} = -15 mA	2.4		V
	I_{OL}	Off-state Output Current	V_{OL} = 0.4V		-20	μA
	I_{OH}	Off-state Output Current	V_{OH} = 2.7V		20	μA
	*C_L	Capacitive load			15	pf

Table 2-3. iSBC 204 Interface DC Characteristics (Cont'd)

Signals	Symbol	Parameter Description	Test Conditions	Min.	Max.	Units
DAT0/-DAT7/	V_{OL}	Output Low Voltage	I_{OL} = 50 mA		0.6	V
	V_{OH}	Output High Voltage	I_{OH} = -10 mA	2.4		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45V		-0.35	mA
	I_{IH}	Input Current at High V	V_{IN} = 5.25V		110	μA
	*C_L	Capacitive load			25	pf
IORC/, IOWC/	I_{IL}	Input Low Current	V_{IN} = 0.4V		-20	mA
	I_{IH}	Input High Current	V_{IN} = 2.4V		60	μA
	*C_L	Capacitive load			25	pf
MRDC/, MWTC/	I_{OL}	Output Low Current	V_{OL} = 0.45V		-160	μA
	I_{OH}	Output High Current	V_{OH} = 5.25V		100	μA
	V_{OL}	Output Low Voltage	I_{OL} = 32 mA		0.45	V
	V_{OH}	Output High Voltage	I_{OH} = -2 mA	2.4		V
	*C_L	Capacitive load			10	pf
XACK/	V_{OL}	Output Low Voltage	I_{OL} = 16 mA		0.4	V
	V_{OH}	Output High Voltage	I_{OH} = -5.2 mA	2.4		V
	I_{IL}	Input Low Current	V_{IL} = 0.4V		-0.44	mA
	I_{IH}	Input High Current	V_{IH} = 2.4V		60	μA
	*C_L	Capacitive load			25	pf
AACK/	I_{IL}	Input Low Current	V_{IL} = 0.4V		-0.4	mA
	I_{IH}	Input High Current	V_{IH} = 2.4V		20	μA
	*C_L	Capacitive load			15	pf
INT0/-INT7/	V_{OL}	Output Low Voltage	I_{OL} = 16 mA		0.4	V
	V_{OH}	Open Collector with 10K Pullup	I_{OH} = -225 μA	2.4		V
	*C_L	Capacitive load			20	pf
INIT/	I_{IL}	Input Low Current	V_{IL} = 0.4V		-0.9	mA
	I_{IH}	Input High Current	V_{IH} = 2.7V		120	μA
	*C_L	Capacitive load			15	pf
BPR0/	V_{OL}	Output Low Voltage	I_{OL} = 3.2 mA		0.45	V
	V_{OH}	Output High Voltage	I_{OH} = -400 μA	2.4		V
	*C_L	Capacitive load			10	pf
BREQ/	V_{OL}	Output Low Voltage	I_{OL} = 20 mA		0.45	V
	V_{OH}	Output High Voltage	I_{OH} = -400 μA	2.4		V
	*C_L	Capacitive load			10	pf
BPRN/, BCLK/	I_{IL}	Output Low Current	V_{IL} = 0.45V		-0.5	mA
	I_{IH}	Output High Current	V_{IH} = 5.25V		100	μA
	*C_L	Capacitive load			10	pf
BUSY/	V_{OL}	Output Low Voltage	I_{OL} = 20 mA		0.45	V
	V_{OH}	Open Collector				
	I_{IL}	Input Low Current	V_{IL} = 0.45V		-0.6	mA
	I_{IH}	Input High Current	V_{IH} = 5.25V		100	μA
	*C_L	Capacitive load			15	pf

*Capacitive values are approximations.

Table 2-4. AC Characteristics (Master Mode)

Parameter	Minimum	Maximum	Description
t_{BCY}	100 ns		Bus Clock Period
t_{BW}	35 ns		Bus Clock Pulse Width
t_{RQD}		35 ns	BCLK/ to BREQ/ Delay
t_{CPD}		40 ns	BCLK/ to BPRO/ Delay
t_{CBL}		40 ns	BCLK/ to BUSY/ Low Delay
t_{CBH}		55 ns	BCLK/ to BUSY/ High Delay
t_{ASR}	185 ns		Address Set-Up Time (Read)
$T_{AH\ (READ)}$	100 ns		Address Hold Time
$t_{AH\ (WRITE)}$	300 ns		Address Hold Time
t_{DXL}	-385 ns		Read Data Set-Up Time to XACK
t_{XCR}	945 ns	1480 ns	Acknowledge to Command High (Read)
t_{DHR}	-260 ns		Data Hold Time (Read)
t_{ASW}	685 ns		Address Set-Up Time (Write)
t_{DSW}	56 ns		Data Set-Up Time (Write)
t_{XCW}	695 ns	1230 ns	Acknowledge to Command High (Write)
t_{DHW}	150 ns		Data Hold Time (Write)
t_{CXD}	0 ns	100 ns	Command to Acknowledge Delay
t_{INIT}	5 μs		Bus Reset

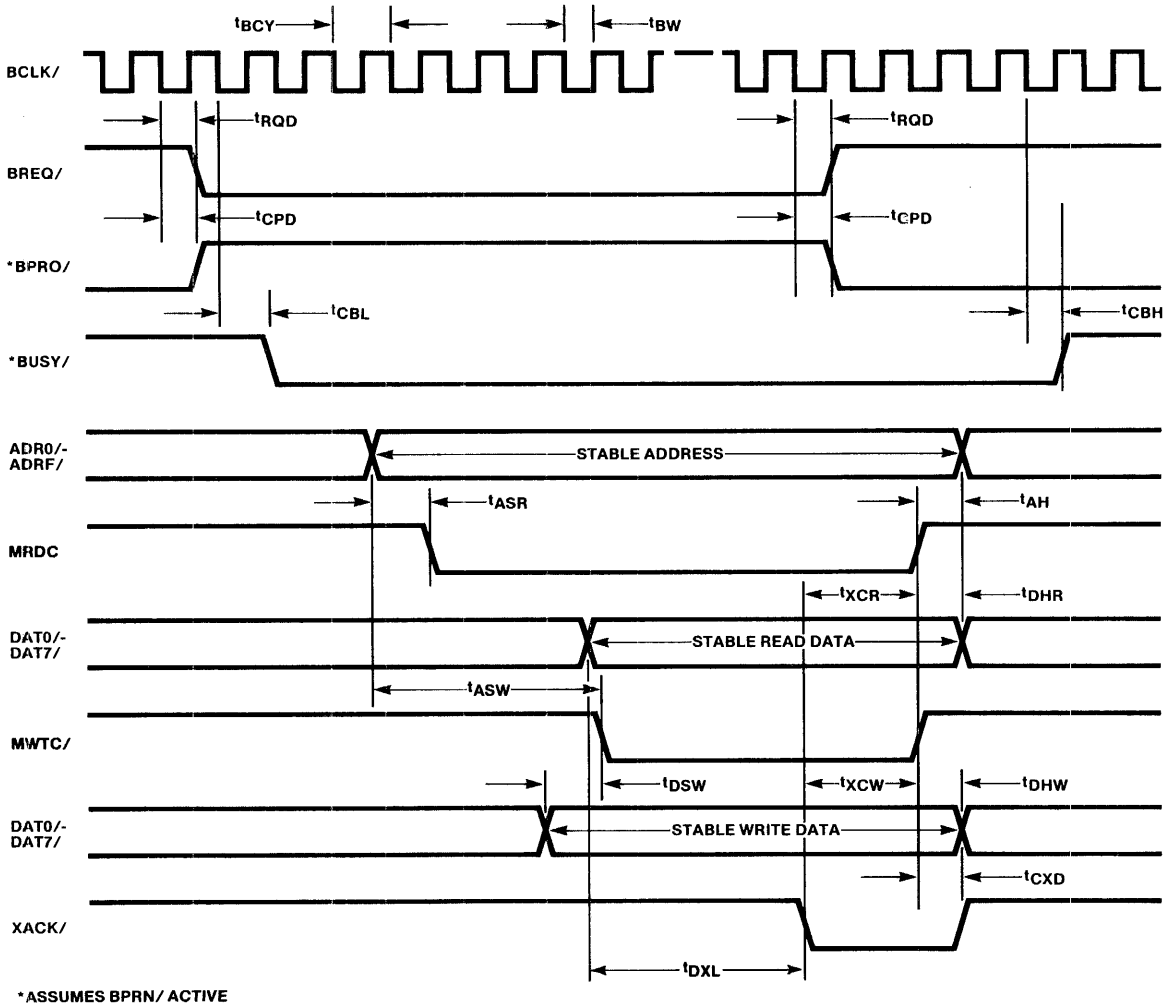


Figure 2-1. Bus Acquisition and Memory Transfer Timing.

Table 2-5. AC Characteristics (Slave Mode)

Parameter	Minimum	Maximum	Description
t_{AS}	20 ns		Address Set-Up Time
t_{AH}	20 ns		Address Hold Time
t_{XACK}	462 ns	584 ns	Command to Acknowledge
t_{DXL}	20 ns		Read Data Set-Up Time
t_{DXT}	35 ns		Read Data Hold Time
t_{CX}		75 ns	Acknowledge Hold Time
t_{DS}	-70 ns		Write Data Set-Up Time
t_{DH}	0		Write Data Hold Time

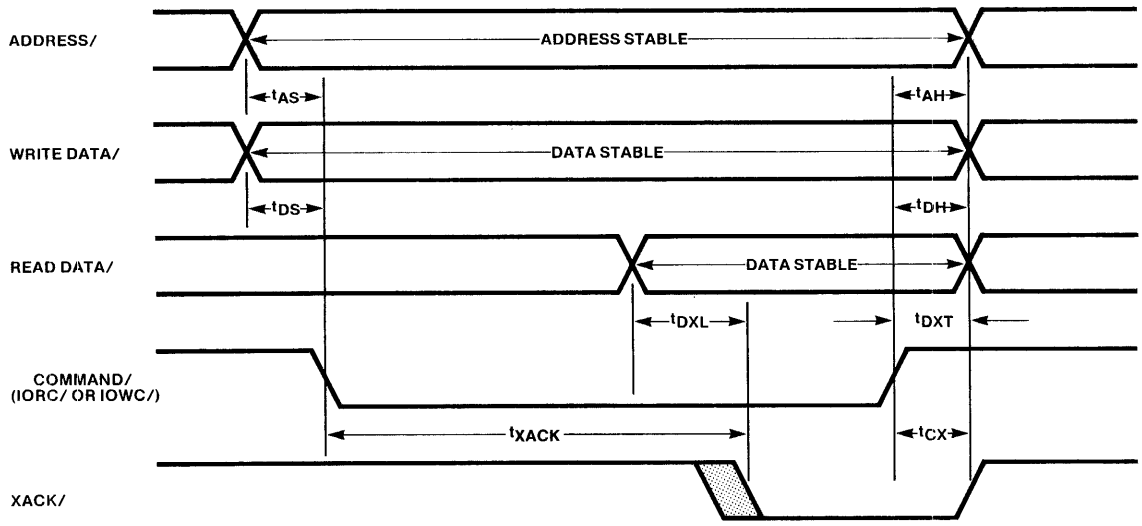


Figure 2-2. I/O Transfer Timing

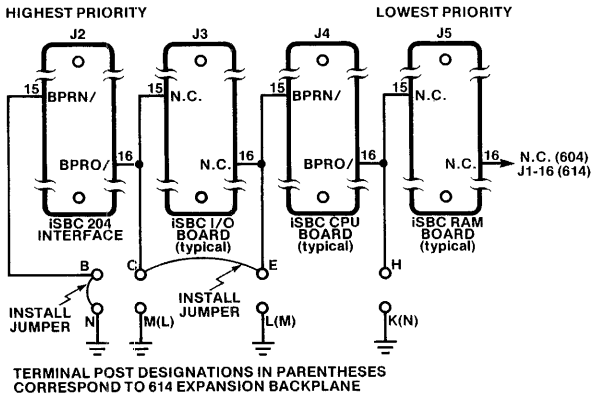


Figure 2-3. Serial Priority Resolution

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 which would provide the required BPRO/-BPRN/ continuity around a "slave" installed in the second slot (J3).

When installing the interface in a parallel priority resolution environment, the interface should be given the highest bus priority. In an Intellec microcomputer development system, the interface 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 microcomputer development system, the interface should be installed in the bottom slot.

2-9. INTERFACE BOARD CONFIGURATION

The interface board includes a number of alterable jumper connections and a set of switches which are used to configure the board to its intended system environment. The following subsections define each jumper connection and its corresponding function.

Note that with the exception of interrupt level priority selection, all jumper connections consist of 30 AWG (solid) Kynar wire connected between individually-numbered, wire-wrap terminal posts. Table 2-6 defines the jumper function, the factory configuration, the corresponding subsection describing the jumper connection, and the location coordinates of the jumper connection on the schematic and assembly drawings.

2-10. DATA RATE

The interface supports two data transfer rates; 125 kilobits per second and 250 kilobits per second. The 125 kilobit rate is required by the mini-sized diskette drives, while the 250 kilobit rate is required by the standard-sized diskette drives. The board is configured at the factory for a 250 kilobit data rate (standard-sized drives). When the interface is to be connected to a mini-sized drive, the following jumper connection modifications must be performed.

1. Remove the jumper between terminal posts 37 and 38.
2. Install a jumper between terminal posts 36 and 37.
3. Remove the jumper between terminal posts 23 and 24.
4. Install a jumper between terminal posts 22 and 23.

2-11. PARALLEL PRIORITY

The interface is configured for installation into a serial priority environment with the factory installation of a jumper between terminal posts 55 and 56. When the interface is to be used in a parallel priority environment (e.g., in one of the Intellec microcomputer development systems), remove the jumper between terminal posts 55 and 56.

2-12. COMMON READY

The interface is arranged at the factory to accept an individual ready indication from each drive in the system. For multi-drive systems in which only a common ready indication is provided, the Ready signal must be coupled to the two Ready inputs of the FDC circuit for proper operation. Two sets of jumpers (one for the standard and one for the optional FDC circuit) are incorporated on the board. To support

Table 2-6. Alterable Jumper Connections

Jumper Function	Factory Configuration	Corresponding Subsection	Schematic Reference*	Assembly Reference*
Data Rate	37 to 38 23 to 24	2-10	2ZD4 3ZC7	C3 C5
Advanced Acknowledge	Removed	2-11	3ZC7	B6
Parallel Priority	55 to 56	2-12	3ZA4	B7
Common Ready	19 to 20 26 to 27	2-13	4ZA6 5ZA6	C5 C4
Ready Enable	Removed	2-14	4ZA7 5ZA7	C6 C4
Alternate Drive I/O Pin Assignments	8 to 1 45 to 47	2-15	4ZA3 5ZA3	D7 D3
Interrupt Selection	63 to 67	2-16	2ZD6	B6
Second FDC	75 to 76 77 to 78	2-17	5ZC3 5ZC4	5A 5C
12-Bit I/O Port Addressing	B to C	~~2-18~~	1ZC3	6C
Reserved Jumper Locations 28-31, 53-54, 59-60, 73-74	Removed	Not used on this configuration.	N/A	4C 3C 6B 4B

*Schematic and assembly drawings are contained in Chapter 5. The first digit of the schematic reference specifies the sheet number, and the last three characters specify the zone (Z) coordinates (e.g., D4).

common ready operation, perform the following jumper modifications.

DRIVE NUMBERS	REMOVE JUMPERS BETWEEN POSTS:	INSTALL JUMPER BETWEEN POSTS:
0 and 1	19 and 20	20 and 21
2 and 3	26 and 27	25 and 26

2-13. READY ENABLE

The mini-sized drives that are supported do not provide a "drive ready" indication to the interface. When this type of drive is used, the corresponding ready input to the FDC circuit either must be permanently enabled or a ready signal generation circuit must be fabricated at the drive (consult the manufacturer's documentation for recommended ready circuit design). Four separate jumper positions are incorporated on the board to permanently enable (ground) the appropriate ready input, one for each of the four possible drives. Depending on the number of drives supported, install wire-wrap jumpers between the following terminal posts.

DRIVE NUMBER	TERMINAL POSTS
0	17 and 18
1	16 and 15
2	34 and 35
3	32 and 33

2-14. ALTERNATE DRIVE I/O PIN ASSIGNMENTS

To increase the interface's drive compatibility, the FAULT and FAULT RESET/OPO signal lines and the LOAD HEAD output signal line are routed initially to a cross-connect matrix for drive I/O connector pin selection. The LOAD HEAD output signal is wired at the factory to appear on pin 17 of drive I/O connectors J1 and J2, while the FAULT and FAULT RESET/OPO signal lines are not connected. Tables 2-7 and 2-8 define the available pin assignments for drive I/O connectors J1 (drives 0 and 1) and J2 (drives 2 and 3).

Table 2-7. Drives 0 and 1 Cross-Connect Matrix

SIGNAL NAME	CROSS-CONNECT MATRIX		CORRESPONDING J1 PIN ASSIGNMENT
	SIGNAL TERMINAL POST	CONNECTOR TERMINAL POST	
		1	17
		2	23
LOAD HEAD	8	3	29
		4	31
		5	47
FAULT RESET/OPO	7	6	49
		10	15
		11	13
FAULT	9	12	9
		13	7
		14	1

Cross-connect matrix factory wired from terminal post 8 (LOAD HEAD) to teminal post 1 (J1, pin 17).

Table 2-8. Drives 2 and 3 Cross-Connect Matrix

SIGNAL NAME	CROSS-CONNECT MATRIX		CORRESPONDING J2 PIN ASSIGNMENT
	SIGNAL TERMINAL POST	CONNECTOR TERMINAL POST	
		47	17
		39	23
LOAD HEAD	45	40	29
		41	31
		42	47
FAULT RESET/OPO	44	43	49
		48	15
		49	13
FAULT	46	50	9
		51	7
		52	1

Cross-connect matrix factory wired from terminal post 45 (LOAD HEAD) to terminal post 47 (J2, pin 17).

2-15. INTERRUPT SELECTION

The interface's internal interrupt request signal can be assigned to any one of eight interrupt priority levels (INT0/-INT7/) on the bus. The interrupt matrix itself is arranged in three rows of four terminal posts each. The terminal posts in the center row (terminal posts 65 through 68) are common, and the eight interrupt priority levels appear on the individual terminal posts in the two outside rows. When shipped from the factory, a shorting connector is positioned over terminal posts 63 and 67 to provide an interrupt request on interrupt priority level 2 (INT2/). To change the interrupt priority level, remove the shorting connector from terminal posts 63 and 67, and insert the connector over the appropriate outside row terminal post and its corresponding center row terminal post. Table 2-9 defines the priority level and corresponding terminal post assignment.

Note that when assigning the interrupt priority level, INT0/ is the highest priority and INT7/ is the lowest priority. Generally, disk interface devices are assigned the highest *available* priority (INT0/ and INT1/ are usually reserved for system devices such as the power-fail monitor and the real time clock). If interface interrupts are to be detected only by polling the controller (the CPU interrogates the interface's interrupt request bit), the shorting connector can be positioned over any two adjacent terminal posts in the center row (terminal posts 65 through 68) or it can be removed from the board.

2-16. SECOND FDC

The interface is configured for a single FDC with the factory installation of jumpers between terminal posts 75 and 76 and 77 and 78. When a second FDC is installed in the interface, remove the jumpers between terminal posts 75 and 76 and 77 and 78.

2-17. 12-BIT I/O PORT ADDRESSING

The interface is configured for an 8-bit I/O port address with the factory installation of a soldered jumper between pads B and C of W1. When 12-bit I/O port addressing is used, the jumper soldered between pads B and C, of W1, must be removed and a jumper must be installed between pads A and B of W1.

2-18. I/O BASE ADDRESS SELECTION

In addition to the configuration jumpers previously described, the location of the interface's I/O port address block must be assigned. This location (base address) is defined by the four most-significant bits (Bits 4-7) of the 8-bit I/O port address, or when 12-bit I/O addressing is used, the location is defined by the eight most-significant bits (Bits 4-B) of the 12-bit I/O port address. When the CPU accesses the interface through the execution of I/O read and write instructions, the base address is used to select the interface, and the four least-significant bits of the I/O port address are used to define the interface command to be executed.

The base address is determined by opening (deselecting) or closing (selecting) the switch positions labeled 4, 5, 6, and 7 (for 8-bit addressing) and 8, 9, A, and B (for 12-bit addressing) of DIP switch S2. These eight switch positions represent the most significant hexadecimal digit of the two-or-four-digit I/O base address and are set at the factory to 0 (all switches off). Table 2-10 defines the available I/O bus addresses (00_{16} to $F0_{16}$) and corresponding switch positions selected for 8-bit addressing. Table 2-11 defines the available I/O bus addresses (000_{16} to $FF0_{16}$) and corresponding switch positions selected for 12-bit addressing.

NOTE

Use switch position numbers printed on the board. Do not use switch position numbers printed on switch.

2-19. DRIVE INTERFACING

The standard interface is capable of supporting either two single-sided drives or one double-sided drive. Optionally, a second 8271 FDC IC can be inserted into the interface board to allow up to four

Table 2-9. Interrupt Priority Levels

INTERRUPT PRIORITY LEVEL	TERMINAL POST CONNECTION	INTERRUPT PRIORITY LEVEL	TERMINAL POST CONNECTION
INT0/	61 to 65	INT4/	72 to 68
INT1/	62 to 66	INT5/	71 to 67
INT2/	63 to 67	INT6/	70 to 66
INT3/	64 to 68	INT7/	69 to 65

Table 2-10. 8-Bit Base Address Switch Settings

I/O Base Address (Hexadecimal)	S2 SWITCH POSITION CALLOUT ON PC BOARD			
	7	6	5	4
00	OFF	OFF	OFF	OFF
10	OFF	OFF	OFF	ON
20	OFF	OFF	ON	OFF
30	OFF	OFF	ON	ON
40	OFF	ON	OFF	OFF
50	OFF	ON	OFF	ON
60	OFF	ON	ON	OFF
70	OFF	ON	ON	ON
80	ON	OFF	OFF	OFF
90	ON	OFF	OFF	ON
A0	ON	OFF	ON	OFF
B0	ON	OFF	ON	ON
C0	ON	ON	OFF	OFF
D0	ON	ON	OFF	ON
E0	ON	ON	ON	OFF
F0	ON	ON	ON	ON

Table 2-11. 12-Bit Base Address Expansion

*Most-Significant Hexadecimal Digit	S2 SWITCH POSITION CALLOUT ON PC BOARD			
	B	A	9	8
0	OFF	OFF	OFF	OFF
1	OFF	OFF	OFF	ON
2	OFF	OFF	ON	OFF
3	OFF	OFF	ON	ON
4	OFF	ON	OFF	OFF
5	OFF	ON	OFF	ON
6	OFF	ON	ON	OFF
7	OFF	ON	ON	ON
8	ON	OFF	OFF	OFF
9	ON	OFF	OFF	ON
A	ON	OFF	ON	OFF
B	ON	OFF	ON	ON
C	ON	ON	OFF	OFF
D	ON	ON	OFF	ON
E	ON	ON	ON	OFF
F	ON	ON	ON	ON

*Precedes I/O base address value listed in table 2-10 (8-Bit Base Address Switch Settings).

single-sided or two double-sided drives to be supported. Each FDC circuit supports two drives through its associated I/O connector (J1 or J2). Since each circuit essentially is independent from the other, drives associated with one circuit (and I/O connector) can have dissimilar operating characteristics (e.g., head load time, track-to-track access time, etc.) from the drives associated with the other FDC circuit. Note however that since the standard- and mini-sized drives have unique data rates, the interface cannot support both standard- and mini-sized drives simultaneously. Figure 2-4 depicts a typical four drive system.

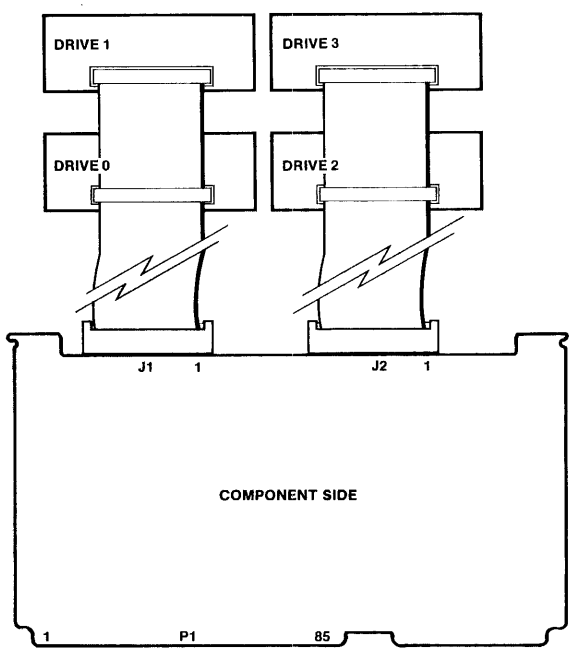


Figure 2-4. Multiple Drive Connection

2-20. INTERFACE-DRIVE I/O SIGNALS

The Drive I/O signal definitions are listed in table 2-12 and the Drive Interface dc characteristics are listed in table 2-13. Figures 2-5 through 2-7 describe the timing and ac characteristics of the I/O signals between the interface and the drive.

2-21. INTERFACE MATING CONNECTOR

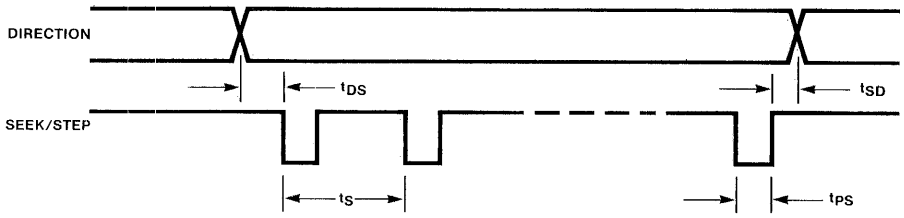
The interface pc board connectors (J1 and J2) follow Intel numbering conventions in that odd-numbered pins appear on the top (component) side of the board and even-numbered pins appear on the bottom (circuit side). Consequently, pin 1 of the interface mating connector aligns with pin 2 of the pc board connector. Note that when using the AMP 1-583717-1 single termination connector to mate with the interface board pc connector, pin A of the mating connector aligns with pin 2 of the interface connector.

Table 2-12. Drive I/O Signal Definitions

SIGNAL NAME	FUNCTION
SELECT 0	A low-state active control level that enables the multiplexed I/O lines for drive 0 (connector J1) or drive 2 (connector J2). When active, this signal allows the corresponding drive to accept the remaining drive input signals from the interface and to gate its output signals to the interface.
SELECT 1	A low-state active control level that enables the multiplexed I/O lines for drive 1 (connector J1) or drive 3 (connector J2). When active, this signal allows the corresponding drive to accept the remaining drive input signals from the interface and to gate its output signals to the interface.
FAULT RESET/OPO	This optional output control line is user program-controlled and generally is used to clear a write fault indication within the drive or, with the mini-sized drives, to control the drive's motor (a logic low level enables the drive motor).
WRITE ENABLE	A low-state active control level 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 (logic high) state, the write electronics are disabled and the drive reads data from the diskette.
SEEK/STEP	An active low-going pulse which 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 (DIREC) output line. The pulse width is 10 μs, and the pulse repetition rate (track-to-track access time) is program-selectable in 1 ms steps ranging from 1 to 255 ms (standard-sized drives) or in 2 ms steps ranging from 2 to 510 ms (mini-sized drives). When the optional count input (COUNT/OPI) is used, the SEEK/STEP signal is held at an active low level until the appropriate number of user-supplied step pulses have been received on the count input line.
DIREC	A control level indicating the direction that the head is stepped. When this line is at a logic low level, the drive's read/write head is stepped towards the spindle (step in), and when this line is at a logic high level, the head is stepped towards track 0 (step out).
LOAD HEAD	A low-state active control level that causes the drive's read/write head to be loaded against the diskette. Note that when the head initially is loaded, there is a programmed delay (head load time) prior to any read or write operation. Provision also is made to unload (retract) the head following an operation within a programmed number of diskette revolutions.
WRITE DATA	The serial data/clock composite write signal. A high- to low-going transition on this line indicates a bit to be written on the diskette.
READY 0	An active low level on this input indicates that drive 0 (or drive 2) is ready. The qualifications for this signal are drive dependent and usually include diskette in place, door closed, and two index marks detected. The interface continually monitors this input during an operation and, if a not ready condition occurs, immediately terminates the operation.
READY 1	This input signal is functionally indentical to the READY 0 signal and, when active, indicates that drive 1 (or drive 3) is ready.
FAULT	An active low level on this input indicates that a write fault condition has been detected by the drive. Drives that include write fault detection logic generally latch this condition.

Table 2-12. Drive I/O Signal Definitions (Cont'd)

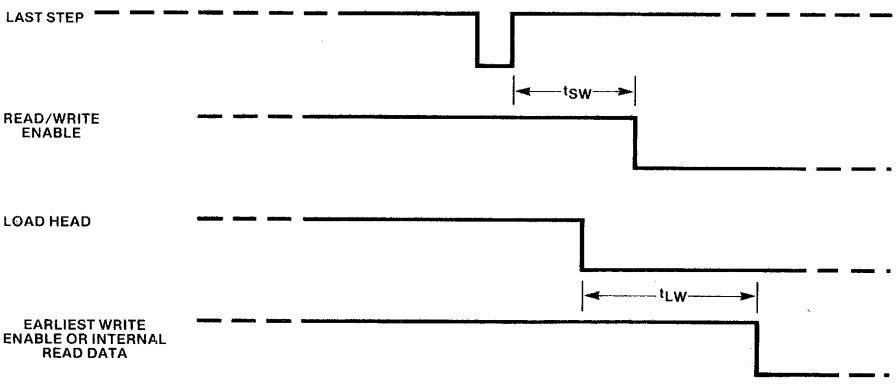
SIGNAL NAME	FUNCTION
COUNT/OPI	A user defined input line that accepts an external clock signal for drive stepping (SEEK/STEP output held logically low) or any special or optional drive status signal (e.g., two sided, disk change).
WRITE PROTECT	An active low level on this input indicates that the drive is write protected.
TRK0	An active low level on this input indicates that the drive's read/write head currently is positioned over track 0.
INDEX	A low-going pulse coincident with the detection of the index hole in the diskette. The width of the pulse must be 0.5 μs or greater.
UNSEP DATA	The composite (unseparated) data and clock signal generated during a diskette read operation. A high- to low-going transition indicates a clock bit or data "one" bit



FUNCTION	MINIMUM	MAXIMUM
t_{DS}	10 μs	
t_{SD}	10 μs	
t_{PS}	10 μs	
t_S	1 ms	255 ms
	*2 ms	*510 ms

*Mini-Sized Drives

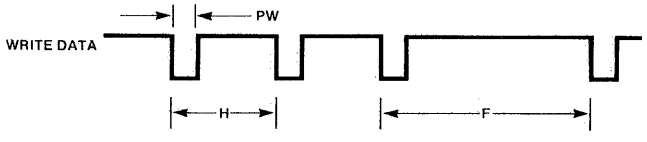
Figure 2-5. Seek Timing 568-6



FUNCTION	MINIMUM	MAXIMUM
t_{SW}	0 ms	255 ms
	*0 ms	*510 ms
t_{LW}	0 ms	60 ms
	*0 ms	*120 ms

*Mini-Sized Drives

Figure 2-6. Head Settling/Head Load Timing 568-7



PULSE WIDTH (PW) = t_{CY} ± 30 ns
HALF BIT CELL (H) = 8 t_{CY}
FULL BIT CELL (F) = 16 t_{CY}

FUNCTION	MINIMUM	MAXIMUM
t_{CY}	249 ns	251 ns
	*498 ns	*502 ns

*Mini-Sized Drives

Figure 2-7. Write Data Timing 568-8

Table 2-13. Drive Interface DC Characteristics

Signals	Symbol	Parameter Description	Test Conditions	Min.	Max.	Units
WR DATA, WR ENABLE, SEEK/STEP, DIREC, LOW CURRENT, SELECT, 0, SELECT 1, FAULT-RST/OPO, LOAD HEAD	V_{OL}	Output Low Voltage	I_{OL} = 40mA		0.7	V
	I_{OH}	Output High Leakage	Open Collector Off V_{OH} = 5.25V		250	μA
	C_L[1]	Capacitive Load			15	pf
UNSEP DATA, COUNT/ OPI, TRACK 0, WR PROTECT, INDEX, READY 0, READY 1, FAULT	V_{IL}	Input Low Voltage		2.0	0.6	V
	V_{IH}	Input High Voltage				V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V		-35	mA[2]
	I_{IH}	Input Current at High V	V_{IN} = 5.25V		3.3	mA[2]
	C_L[1]	Capacitive Load			15	pf

Notes:
1. Capacitance values are approximations.
2. Includes current from 150 Ω pull-up.

Table 2-14 defines the pin assignments of the pc board interface connectors J1 and J2 and the corresponding mating connector pin assignments.

2-22. CABLE FABRICATION

The I/O cable between the interface and drive(s) consists of the desired length of either flat ribbon or twisted pair cable. Each cable end is fitted with the appropriate mating connector and, in the case of daisy-chain cables, a second drive mating connector is inserted into the cable. A majority of the drives described in Section 2-29 are directly pin-to-pin compatible and can use the flat ribbon cable and mass-termination type connectors. Drives that are

Table 2-14. Connector Pin Correlation

INTERFACE CONNECTOR (J1 or J2)	MATING CONNECTOR		INTERFACE CONNECTOR (J1 or J2)	MATING CONNECTOR	
	3M 3415-0001	AMP 1-583717-1		3M 3415-0001	AMP 1-583717-1
1	2	1	26	25	P
2	1	A	27	28	14
3	4	2	28	27	R
4	3	B	29	30	15
5	6	3	30	29	S
6	5	C	31	32	16
7	8	4	32	31	T
8	7	D	33	34	17
9	10	5	34	33	U
10	9	E	35	36	18
11	12	6	36	35	V
12	11	F	37	38	19
13	14	7	38	37	W
14	13	H	39	40	20
15	16	8	40	39	X
16	15	J	41	42	21
17	18	9	42	41	Y
18	17	K	43	44	22
19	20	10	44	43	Z
20	19	L	45	46	23
21	22	11	46	45	a
22	21	M	47	48	24
23	24	12	48	47	b
24	23	N	49	50	25
25	26	13	50	49	c

not directly pin-to-pin compatible require the twisted pair cable and a single-termination type connector.

2-23. FLAT RIBBON CABLE. The flat ribbon cable used is 50 conductor, 28 AWG (stranded) and is available from several sources including the 3M Company (P/N 3365/50) and T&B/Ansley Corporation (P/N 171-50). The interface mating connector is a 3M 3415-0001, T&B/Ansley 609-5015 or equivalent. The drive mating connector is specified in table 2-15, Drive Parameters. When two drives are to be connected to the interface on a single cable (daisy-chained), a second drive interface connector is inserted directly into the cable. Figure 2-8 illustrates a typical daisy-chain flat ribbon cable. Note that to insert the mass-termination type connectors into the cable, a special press is required.

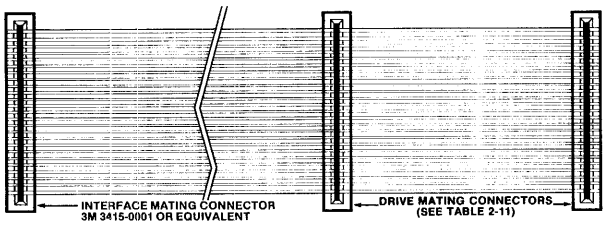


Figure 2-8. Flat Ribbon Cable

2-24. TWISTED PAIR CABLE. Twisted pair cable is used when the interface and drive are not directly pin-to-pin compatible or when the improved electrical characteristics of twisted pair cable are required. A combination cable, which alternates between twisted pair and flat ribbon sections (e.g., Spectra-Strip "Twist 'N' Flat" cable), can be used and permits the use of mass-termination type connectors. When fabricating a daisy-chain, twisted-pair cable, the combination cable is recommended since the mass-termination type connector can be readily inserted into the flat ribbon sections. In this case, the two drive mating connectors would be the mass-termination type specified in table 2-15 and the individual contacts of the interface mating connector would be wired according to compatibility requirements. Figure 2-9 illustrates a typical daisy-chain combination cable.

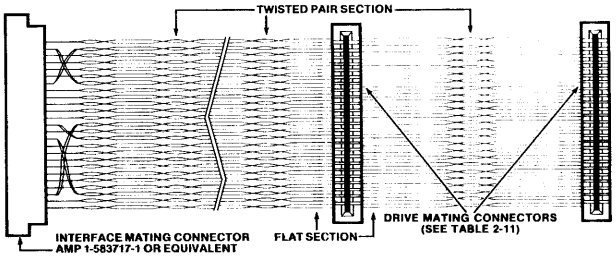


Figure 2-9. Combination Cable

In figure 2-9 the interface mating connector is an AMP P/N 1-583717-1 with either 583854-5 (solder) or 583616-5 (crimp) type contacts. The combination cable is Spectra-Strip "Twist 'N' Flat" 28 AWG (stranded). For the 5¼ inch mini-sized drives, 34 conductor (17 pair) cable, P/N 455-248-34, is used while with the standard-sized drives, 50 conductor (25 pair) cable, P/N 455-248-50, is used.

2-25. DRIVE MODIFICATIONS

The ensuing subsections define the general modifications which must be performed to all drives. Unique modifications associated with individual drive models are described in the drive cabling tables in Section 2-29.

2-26. DRIVE TERMINATION

When two drives are to be connected (daisy-chained) to one of the interface's drive I/O connectors (J1 or J2), the termination resistance on the following common (multiplexed) output signal lines must be removed from the first physical drive on the cable (drive nearest the interface).

Direction
Step
Write Data
Write Gate
Head Load
Low Current (if used)
Fault Reset/OPO (if used)

Refer to the corresponding drive technical documentation for the locations of the termination resistors (or networks) and the method of removal.

Table 2-15. Drive Parameters

DRIVE MANUFACTURER AND MODEL NUMBER	PARAMETER							
	FLAT CABLE DRIVE MATING CONNECTOR	TWISTED PAIR DRIVE MATING CONNECTOR	REQUIRED DRIVE OPTIONS	SUPPORTED DRIVE OPTIONS	HEAD LOAD TIME	HEAD SETTLING TIME	STEP RATE (TRACK-TO-TRACK ACCESS TIME)	REQUIRED DRIVE MODIFICATIONS
SHUGART SA400	3M 3463-0001 T&B/Ansley 609-3415 AMP 88104-1	AMP 583717-5 With Contact: 583854-5 (Solder) 583616-5 (Crimp)	None	N/A	75ms	10ms	40ms	None
SHUGART SA800	3M 3415-0001 T&B/Ansley 609-5015 AMP 88083-1	AMP 1-583717-1 With Contact: 583854-5 (Solder) 583616-5 (Crimp) Viking 3VH25/1JN5	None	Write Protect	35ms	8ms	8ms	See Table 2-18
SHUGART SA850	3M3415-0001 T&B/Ansley 609-5015 AMP 88083-1	AMP 1-583717-1 With Contact: 583854-5 (Solder) 583616-5 (Crimp) Viking 3VH25/1JN5	None	N/A	35ms	15ms	3ms	See Table 2-19
MEMOREX 550	3M 3415-0001 T&B/Ansley 609-5015 AMP 88083-1	AMP 1-583717-1 With Contact: 583854-5 (Solder) 583616-5 (Crimp) Viking 3VH25/1JN5	None	Write Protect	35ms	10ms	3ms	See Table 2-20
MEMOREX 552	3M 3415-0001 T&B/Ansley 609-5015 AMP 88083-1	AMP 1-583717-1 With Contact: 583854-5 (Solder) 583616-5 (Crimp) Viking 3VH25/1JN5	None	Write Protect	35ms	10ms	3ms	See Table 2-21
CDC 9404	3M 3425-3000 T&B/Ansley 609-5001M	AMP 4-87456-0 With Contact: 86015-5	None	Write Protect	60ms	10ms	10ms	None
PERTEC FD200	3M 3463-0001 T&B/Ansley 609-3415 AMP 88104-1	AMP 583717-5 With Contact: 583854-5 (Solder) 583616-5 (Crimp)	None	N/A	35ms	10ms	25ms	None
GSI-110	3M 3415-0001 T&B/Ansley 609-5015 AMP 88083-1	AMP 1-583717-1 With Contact: 583854-5 (Solder) 583616-5 (Crimp) Viking 3VH25/1JN5	Auto-Erase	N/A	35ms	14ms	6ms	See Table 2-24
WANGCO MOD 82	3M 3463-0001 T&B/Ansley 609-3415 AMP 88104-1	AMP 583717-5 With Contact: 583854-5 (Solder) 583616-5 (Crimp)	None	Head Load	60ms	20ms	30ms	See Table 2-25
WANGCO MOD 76S	3M 3415-0001 T&B/Ansley 609-5015 AMP 88083-1	AMP 1-583717-1 With Contact: 583854-5 (Solder) 583616-5 (Crimp) Viking 3VH25/1JN5	None	Write Protect	30ms	14ms	6ms	See Table 2-26

2-27. DRIVE NUMBERING ASSIGNMENTS

Table 2-16 defines the drive numbering assignments for up to four single-sided or two dual-sided drives.

Internal drive selection may be determined by wire jumper, shorting plug, or individual switches. Refer to drive technical documentation and drive cabling tables (table 2-17 through table 2-26) for drive configuration. Generally, drives are shipped by their manufacturers configured as drive 0 (or 1).

2-28. RADIAL READY

A majority of the compatible drives only provide a "ready" indication when the drive is selected. The interface requires that the corresponding READY input from a drive be enabled irrespective of drive selection. Refer to the appropriate drive documentation and perform the drive modifications necessary to provide an ungated READY signal (generally referred to as the "Radial Ready Option"). Note that the 5¼ inch mini-drives do not provide a ready indication and that the corresponding READY input on the interface board must be permanently enabled or circuitry must be added to the drive to provide the required ready indication (see Section 2-13).

2-29. DRIVE PARAMETERS

Table 2-15 defines the drive parameters for a number of drives that are compatible with the interface. Note that the drive characteristics specified in the table are intended to supplement the drive manufacturer's documentation. *In all cases, the manufacturer's manuals should be consulted to ensure that the parameters specified have not been changed.*

Also included within this section are individual drive cabling tables for compatible drives. Note that this list is not intended to be conclusive. When it is desired to support a drive that is not described in the cabling tables, the drive parameters, signal definitions and cabling requirements should be examined for interface compatibility.

Table 2-16. Drive Numbering Assignments

INTERFACE CONNECTOR	INTERFACE SIGNAL NAME	PHYSICAL DRIVE NUMBER		INTERNAL DRIVE NUMBER*	
		SINGLE	DUAL	SINGLE	DUAL**
J1-Pin 25	SELECT 0	Drive 1	Drive 1, Surface 1	0 or 1	0 or 1
J1-Pin 27	SELECT 1	Drive 2	Drive 1, Surface 2	1 or 2	0 or 1
J2-Pin 25	SELECT 0	Drive 3	Drive 2, Surface 1	0 or 1	1 or 2
J2-Pin 27	SELECT 1	Drive 4	Drive 2, Surface 2	1 or 2	1 or 2

*Internal drive number assignments vary among manufacturers and may begin with 0 (0-3) or 1 (1-4).

**SELECT 1 signal used as HEAD SELECT signal on dual-sided drives.

Table 2-17. Shugart SA400 Drive Cabling

ISBC 204 INTERFACE		SHUGART SA400 DRIVE	
SIGNAL NAME	J1/J2 PIN NUMBER	J1 PIN NUMBER	SIGNAL NAME
READY 1	3[1]	No Connect	
LOW CURRENT	5	No Connect	
COUNT/OPI	11	No Connect	
INDEX	19	8[3]	INDEX/SECTOR
READY 0	21[1]	No Connect	
SELECT 0	25	10	DRIVE SELECT 1
SELECT 1	27	12	DRIVE SELECT 2
DIREC	33	18	DIRECTION SELECT
SEEK/STEP	35	20	STEP
WR DATA	37	22	WRITE DATA
WR ENABLE	39	24	WRITE GATE
TRK0	41	26	TRACK 00
WR PROTECT	43	28	WRITE PROTECT
UNSEP DATA	45	30	READ DATA
LOAD HEAD		No Connect	
FAULT		No Connect	
FAULT RESET/OPO	n[2]	16	MOTOR ON
	No Connect	14	DRIVE SELECT 3

Notes:

1. Pin must be grounded on interface board (see Section 2-13).
2. Interface connector pin number determined by cross-connect matrices (see Section 2-14).
3. For flat cable compatibility, jumper pins 4 and 8 of J1 on SA400.

Table 2-18. Shugart SA800 Drive Cabling

ISBC 204 INTERFACE		SHUGART SA800 DRIVE	
SIGNAL NAME	**J1/J2 PIN NUMBER**	**J1 PIN NUMBER**	**SIGNAL NAME**
READY 1	3	4[1]	READY
LOW CURRENT	5	No Connect	
COUNT/OPI	11	12[2]	DISK CHANGE (OPTIONAL)
INDEX	19	20	INDEX
READY 0	21	22	READY
SELECT 0	25	26	DRIVE SELECT 1
SELECT 1	27	28	DRIVE SELECT 2
DIREC	33	34	DIRECTION SELECT
SEEK/STEP	35	36	STEP
WR DATA	37	38	WRITE DATA
WR ENABLE	39	40	WRITE GATE
TRK0	41	42	TRACK 00
WR PROTECT	43	44	WRITE PROTECT (OPTIONAL)
UNSEP DATA	45	46	READ DATA
LOAD HEAD	17[3]	18[4]	HEAD LOAD
FAULT		No Connect	
FAULT RESET/OPO		No Connect	
	No Connect	16	IN USE
	No Connect	30	DRIVE SELECT 3
	No Connect	32	DRIVE SELECT 4
	No Connect	48	SEP DATA
	No Connect	50	SEP CLOCK

Notes:

1. Second drive must be modified internally to provide READY output on pin 4. See Section 7.5 (Radial Ready) in SA800 OEM manual.
2. Disk Change output must be enabled within the drive. See Section 7.10 (Disk Change) in SA800 OEM manual.
3. Cross-connect matrices must be wired to provide LOAD HEAD signal on pin 17. See table 2-7 and table 2-8 for wiring information.
4. Optional internal drive modification. See Section 7.3 (Select Drive and Enable Stepper Motor Without Loading Head) in SA800 OEM manual.

Table 2-19. Shugart SA850 Drive Cabling

iSBC 204 INTERFACE		SHUGART SA850 DRIVE	
SIGNAL NAME	J1/J2 PIN NUMBER	J1 PIN NUMBER	SIGNAL NAME
READY 1	3	No Connect	
LOW CURRENT	5	No Connect	
COUNT/OPI	11	12[1]	DISK CHANGE
INDEX	19	20	INDEX
READY 0	21	22	READY
SELECT 0	25	26[2]	DRIVE SELECT 1
SELECT 1	27	28[2]	DRIVE SELECT 2
DIREC	33	34	DIRECTION SELECT
SEEK/STEP	35	36	STEP
WR DATA	37	38	WRITE DATA
WR ENABLE	39	40	WRITE GATE
TRK0	41	42	TRACK 00
WR PROTECT	43	44	WRITE PROTECT
UNSEP DATA	45	46	READ DATA
LOAD HEAD	17[3]	18[4]	HEAD LOAD
FAULT		No Connect	
FAULT RESET/OPO		No Connect	
	No Connect	10[5]	TWO SIDED
	No Connect	14	SIDE SELECT
	No Connect	16	IN USE
	No Connect	30	DRIVE SELECT 3
	No Connect	32	DRIVE SELECT 4

Notes:

1. Disk Change output must be enabled within the drive. See Section 7.10 (Disk Change) in SA850 OEM manual.
2. Drive Select signals used to select side. See Section 7.12 (Side Selection Using Drive Select) in SA850 OEM manual.
3. Cross-connect matrices must be wired to provide LOAD HEAD signal on pin 17. See table 2-7 and table 2-8 for wiring information.
4. Optional internal drive modification. See Section 7.3 (Select Drive and Enable Stepper Motor Without Loading Heads) in SA850 OEM manual.
5. TWO SIDED signal can be used in place of DISK CHANGE by connecting input of 2S to output of DC on drive pc board.

Table 2-20. Memorex 550 Drive Cabling

iSBC 204 INTERFACE		MEMOREX 550 DRIVE	
SIGNAL NAME	J1/J2 PIN NUMBER	J1 PIN NUMBER	SIGNAL NAME
READY 1	3	4[1]	-READY
LOW CURRENT	5	No Connect	
COUNT/OPI	11	12	-DISC CHANGE
INDEX	19	20	-INDEX
READY 0	21	22	-READY
SELECT 0	25	26	-DRIVE SELECT 1
SELECT 1	27	28	-DRIVE SELECT 2
DIREC	33	34	DIRECTION
SEEK/STEP	35	36	-STEP
WR DATA	37	38	-WRITE DATA
WR ENABLE	39	40	-WRITE GATE
TRK0	41	42	-TRACK 00
WR PROTECT	43	44	-WRITE PROTECT (OPTIONAL)
UNSEP DATA	45	46	-READ DATA COMPOSITE
LOAD HEAD	17[2]	18[3]	-HEAD LOAD
FAULT		No Connect	
FAULT RESET/OPO		No Connect	
	No Connect	16	-IN USE
	No Connect	24	-SECTOR
	No Connect	30	-DRIVE SELECT 3
	No Connect	32	-DRIVE SELECT 4
	No Connect	48	-SEPARATED DATA
	No Connect	50	-SEPARATED CLOCK

Notes:

1. Second drive must be modified internally to provide -READY output on pin 4. See "Radial Ready Line Option" description in drive documentation.
2. Cross-connect matrices must be wired to provide LOAD HEAD signal on pin 17. See table 2-7 and table 2-8 for wiring information.
3. Optional internal drive modification. See "Select Drive Without Loading Head Option" description in drive documentation.

Table 2-21. Memorex 552 Drive Cabling

iSBC 204 INTERFACE		MEMOREX 552 DRIVE	
SIGNAL NAME	J1/J2 PIN NUMBER	J1 PIN NUMBER	SIGNAL NAME
READY 1	3	4	-READY
LOW CURRENT	5	No Connect	
COUNT/OPI	11	12	-DISC CHANGE
INDEX	19	20	-INDEX
READY 0	21	22	-READY
SELECT 0	25	26	-DRIVE SELECT 1
SELECT 1	27	28	-DRIVE SELECT 2
DIREC	33	34	DIRECTION
SEEK/STEP	35	36	-STEP
WR DATA	37	38	-WRITE DATA
WR ENABLE	39	40	-WRITE GATE
TRK0	41	42	-TRACK 00
WR PROTECT	43	44	-WRITE PROTECT (OPTIONAL)
UNSEP DATA	45	46	-READ DATA COMPOSITE
LOAD HEAD		No Connect	
FAULT		No Connect	
FAULT RESET/OPO		No Connect	

Table 2-22. CDC 9404 Drive Cabling

iSBC 204 INTERFACE		CDC 9404 DRIVE	
SIGNAL NAME	J1/J2 PIN NUMBER	J1 PIN NUMBER	SIGNAL NAME
READY 1	3	30	UNIT 2 READY
LOW CURRENT	5	10	LOW WRITE CURRENT
COUNT/OPI	11	No Connect	
INDEX	19	8	INDEX
READY 0	21	28	UNIT READY 1
SELECT 0	25	20	UNIT SELECT 1
SELECT 1	27	22	UNIT SELECT 2
DIREC	33	14	DIRECTION
SEEK/STEP	35	12	STEP
WR DATA	37	18	WRITE DATA
WR ENABLE	39	16	WRITE ENABLE
TRK0	41	6	TRACK 00
WR PROTECT	43	36	WRITE PROTECT
UNSEP DATA	45	2	READ DATA/CLOCK COMPOSITE
LOAD HEAD	n[1]	4	HEAD LOAD
FAULT	n[1]	44	WRITE FAULT
FAULT RESET/OPO	n[1]	46	WRITE FAULT RESET
	No Connect	24	UNIT SELECT 3
	No Connect	26	UNIT SELECT 4
	No Connect	32	UNIT READY 3
	No Connect	34	UNIT READY 4
	No Connect	38	READ DATA SEPARATED
	No Connect	42	READ CLOCK SEPARATED

Notes:

1. Interface pin number selection determined by cross-connect matrices (see Section 2-14).
2. Not directly compatible with flat cable.

Table 2-23. Pertec FD200 Drive Cabling

iSBC 204 INTERFACE		PERTEC FD200 DRIVE	
SIGNAL NAME	J1/J2 PIN NUMBER	J1 PIN NUMBER	SIGNAL NAME
READY 1	3[1]	No Connect	
LOW CURRENT	5	No Connect	
COUNT/OPI	11	No Connect	
INDEX	19	8	INDEX
READY 0	21[1]	No Connect	
SELECT 0	25	10	SELECT 0
SELECT 1	27	12	SELECT 2
DIREC	33	18	DIRECTION
SEEK/STEP	35	20	STEP
WR DATA	37	22	WRITE DATA
WR ENABLE	39	24	WRITE ENABLE
TRK0	41	26	TRACK 00
WR PROTECT	43	28	WRITE PROTECT
UNSEP DATA	45	30	READ DATA
LOAD HEAD		No Connect	
FAULT		No Connect	
FAULT RESET/OPO	n[2]	16	DRIVE-MOTOR ENABLE
	No Connect	14	SELECT 1

Notes:

1. Pin must be grounded on interface board (see Section 2-18).
2. Interface connector pin number determined by cross-connect matrices (see Section 2-14).
3. For flat cable compatibility, jumper pins 4 and 8 of J1 on FD200.

Table 2-24. GSI-110 Drive Cabling

iSBC 204 INTERFACE		GSI-110 DRIVE	
SIGNAL NAME	J1/J2 PIN NUMBER	J3 PIN NUMBER	SIGNAL NAME
READY 1	3	4[1]	READY/
LOW CURRENT	5	6[2]	LO I/
COUNT/OPI	11	No Connect	
INDEX	19	20	INDEX/
READY 0	21	22	READY/
SELECT 0	25	26	SELECT 0/
SELECT 1	27	28	SELECT 1/
DIREC	33	34	STEP IN/
SEEK/STEP	35	36	STEP/
WR DATA	37	38	WRT DATA/
WR ENABLE	39	40	WRITE/
TRK0	41	42	TRACK 00/
WR PROTECT	43	44	WRT PROTECT/
UNSEP DATA	45	46	RAW DATA/
LOAD HEAD	17[3]	18	HDLD/
FAULT		No Connect	
FAULT RESET/OPO		No Connect	
	No Connect	2	IN USE/
	No Connect	14[4]	ERASE/
	No Connect	24	SECTOR/
	No Connect	30	SELECT 2/
	No Connect	32	SELECT 3/
	No Connect	48	SEP DATA/
	No Connect	50	SEP CLK/

Notes:

1. Second drive must be modified internally to provide READY/ output on pin 4. See "Daisy-Chain Radial" description in Volume 2 Technical Manual.
2. For flat cable pin-to-pin compatibility, LO I/ output must be moved on all drives to pin 6 by cutting the etched circuit jumper between pads 16 and installing a jumper from pad 6 to the input side of pad 16.
3. Cross-connect matrices must be wired to provide LOAD HEAD signal to pin 17. See table 2-7 and table 2-8 for wiring information.
4. Auto Erase option must be installed in all drives.

Table 2-25. Wangco MOD 82 Drive Cabling

iSBC 204 INTERFACE		WANGCO MOD 82 DRIVE	
SIGNAL NAME	**J1/J2 PIN NUMBER**	**J1 PIN NUMBER**	**SIGNAL NAME**
READY 1	3[1]	No Connect	
LOW CURRENT	5	No Connect	
COUNT/OPI	11	No Connect	
INDEX	19	8	INDEX/SECTOR
READY 0	21[1]	No Connect	
SELECT 0	25	10	DRIVE SELECT 1
SELECT 1	27	12	DRIVE SELECT 2
DIREC	33	18	DIRECTION SELECT
SEEK/STEP	35	20	STEP
WR DATA	37	22	WRITE DATA
WR ENABLE	39	24	WRITE GATE
TRK0	41	26	TRACK 00
WR PROTECT	43	28	WRITE PROTECT
UNSEP DATA	45	30	READ DATA
LOAD HEAD	n[2]	4[3]	HEAD LOAD
FAULT		No Connect	
FAULT RESET/OPO	n[2]	16	MOTOR ON
	No Connect	14	DRIVE SELECT 3

Notes:

1. Pin must be grounded on interface board (see Section 2-18).
2. Interface connector pin number determined by cross-connect matrices (see Section 2-14).
3. Optional internal drive modification. See "Head Load" in drive documentation. (Note that jumper is added from F to H to direct input to pin 4.)
4. For flat cable compatibility, jumper pins 4 and 8 of J1 on MOD 82.

Table 2-26. Wangco MOD 76S Drive Cabling

ISBC 204 INTERFACE		WANGCO MOD 76S DRIVE	
SIGNAL NAME	J1/J2 PIN NUMBER	J1 PIN NUMBER	SIGNAL NAME
READY 1	3	4[1]	READY
LOW CURRENT	5	No Connect	
COUNT/OPI	11		
INDEX	19	20	INDEX
READY 0	21	22	READY
SELECT 0	25	26	DRIVE SELECT 1
SELECT 1	27	28	DRIVE SELECT 2
DIREC	33	34	DIRECTION SELECT
SEEK/STEP	35	36	STEP
WR DATA	37	38	WRITE DATA
WR ENABLE	39	40	WRITE GATE
TRK0	41	42	TRACK 00
WR PROTECT	43	44	WRITE PROTECT (OPTIONAL)
UNSEP DATA	45	46	READ DATA
LOAD HEAD	17[2]	18[3]	HEAD LOAD
FAULT		No Connect	
FAULT RESET/OPO		No Connect	
	No Connect	30	DRIVE SELECT 3
	No Connect	32	DRIVE SELECT 4
	No Connect	48	SEP DATA
	No Connect	50	SEP CLOCK

Notes:

1. Second drive must be modified internally to provide READY output on pin 4. See "Radial Ready" description in drive interface specification.
2. Cross-connect matrices must be wired to provide LOAD HEAD signal on pin 17. See table 2-7 and table 2-8 for wiring information.
3. Optional internal drive modification. See "Select Drive Without Loading Head" description in drive interface specification.

CHAPTER 3
PROGRAMMING CONVENTION

3-1. INTRODUCTION

This chapter describes the diskette data track formats and the I/O port commands which are executed by the CPU to select and convey information to and from the interface's flexible disk controller (FDC) and DMA controller (DMAC) circuits. In addition, the individual FDC commands which control all diskette operations in the transfer of the data to and from the drive are covered. Individual flow charts and programming examples depicting the various diskette operations are also shown.

All diskette operations are defined and initiated by the CPU through the execution of a series of I/O port commands. The I/O port commands send the starting memory address of the buffer and the type of transfer (read or write) to the DMAC. All other information, about the disk operation, is sent via I/O port commands to the FDC. Once all information has been received, the interface completes the specified operation without further intervention from the CPU. The CPU can either interrogate or "poll" the interface to determine when an operation is complete or, when an operation is completed, the interface can notify the CPU through its interrupt facility. When an operation has been completed, the CPU must interrogate the interface through the I/O port commands to determine the outcome of the operation.

Figure 3-1a shows the relationship of the FDC and DMAC circuits during a DMA data transfer (master mode). Figure 3-1b shows the relationship of the FDC and DMAC circuits during I/O operations (slave mode) e.g. loading FDC commands.

To initiate a disc operation, several I/O port commands are prepared and executed by the CPU. These commands define the operation to be performed, provide all supplementary information (parameters) required to perform the operation and, if a transfer of data to or from a diskette is required, define the direction of the data transfer and the starting memory location of the first data byte to be transferred.

3-2. DISKETTE ORGANIZATION

The interface is compatible with two physical sizes of diskettes; a single- or double-sided, standard-sized diskette which consists of 77 tracks, and a single-sided, mini-sized diskette which typically consists of 35 tracks. Two of the available tracks, are generally reserved as alternates. The tracks are numbered sequentially (beginning at the outermost track) from 00-76 (standard) or from 00-34 (mini). 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-selectable (soft sectored) and are established when the track is formatted.

All tracks on the diskette are referenced to a physical index mark (a small hole in the diskette). Each time the hole passes the photo detector cell (one revolution of the diskette), an index pulse is generated to indicate the logical beginning of the track. This index pulse is also used in track formatting operations to initiate the operation.

The iSBC 204 interface supports the IBM 3740 format of 26, 15, or 8 sectors per track with, respectively, 128, 256, or 512 data bytes per sector (standard-sized diskette) and additionally offers the sector formats for both the standard-sized and mini-sized diskettes listed in table 3-1.

Data and clock bits are written on the diskette using a double frequency recording technique. This allows a data bit to be inserted between two adjacent clock bits. (The presence of a data bit represents a binary one while the absence of a data bit represents a binary zero.) The two adjacent clock bits are referred to as a bit cell and except for unique field identifiers, all clock bits written on the diskette are binary ones. As shown in figure 3-2, eight consecutive bit cells comprise a byte of data with bit cell 0 representing the most significant bit (MSB).

A sector is further divided into two fields; an ID field and a Data field. In addition to the specified number of data bytes, each sector contains address, control, and error checking information. Figures 3-3 and 3-4 depict the IBM compatible standard-sized and mini-sized diskette track formats respectively.

Unique address marks (8-bit data patterns with clock bits intentionally omitted) are used to identify the beginning of each ID and data field and, on the standard diskette, the location of the index mark.

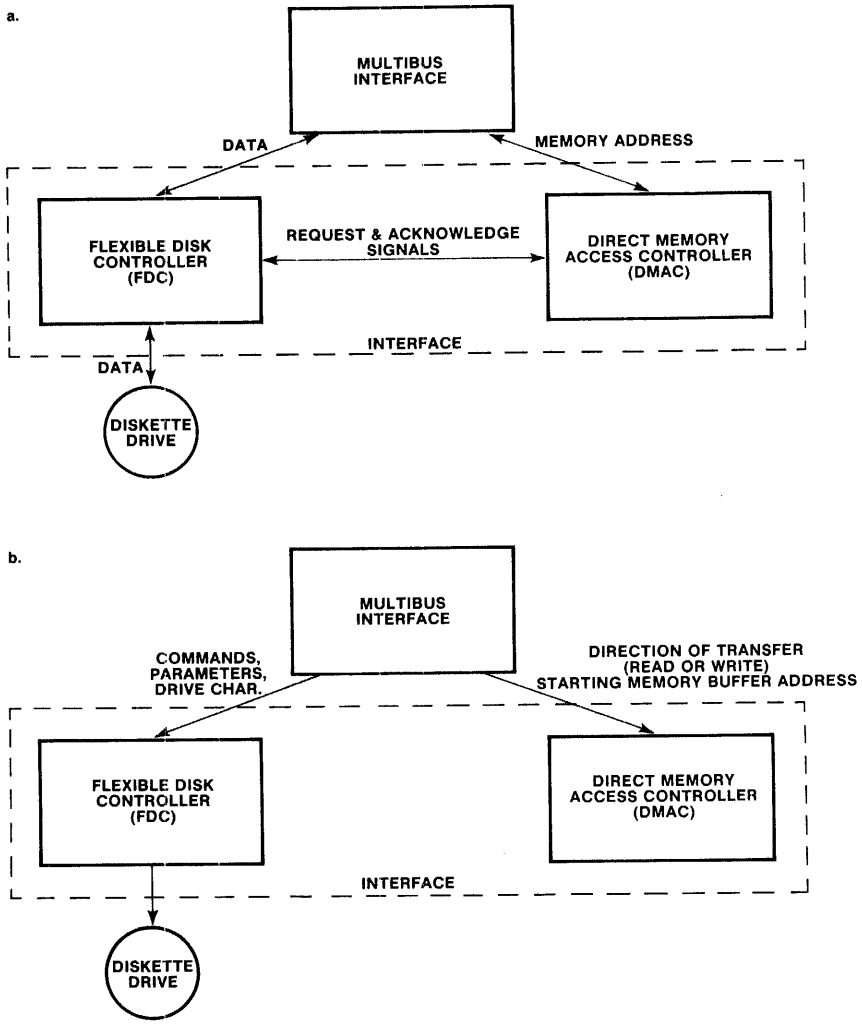


Figure 3-1. Relationship of FDC and DMAC Circuits

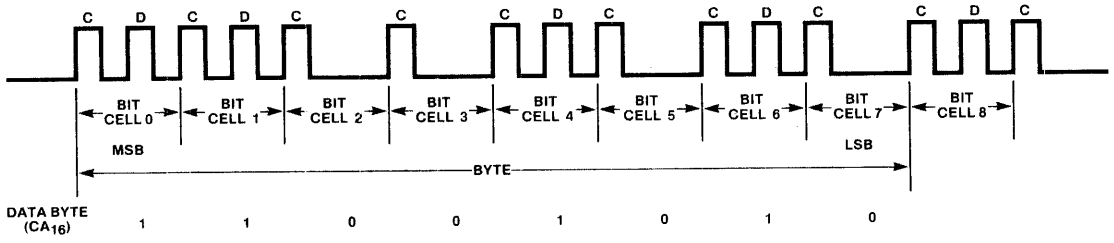
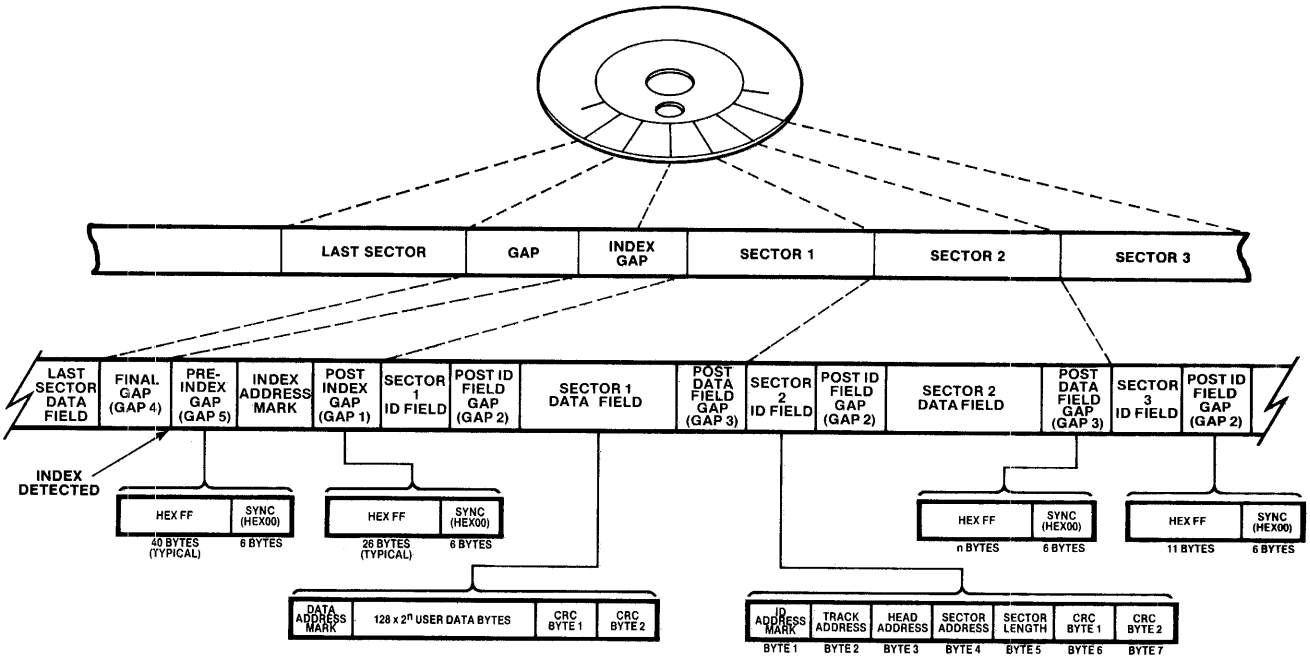


Figure 3-2. Data Byte

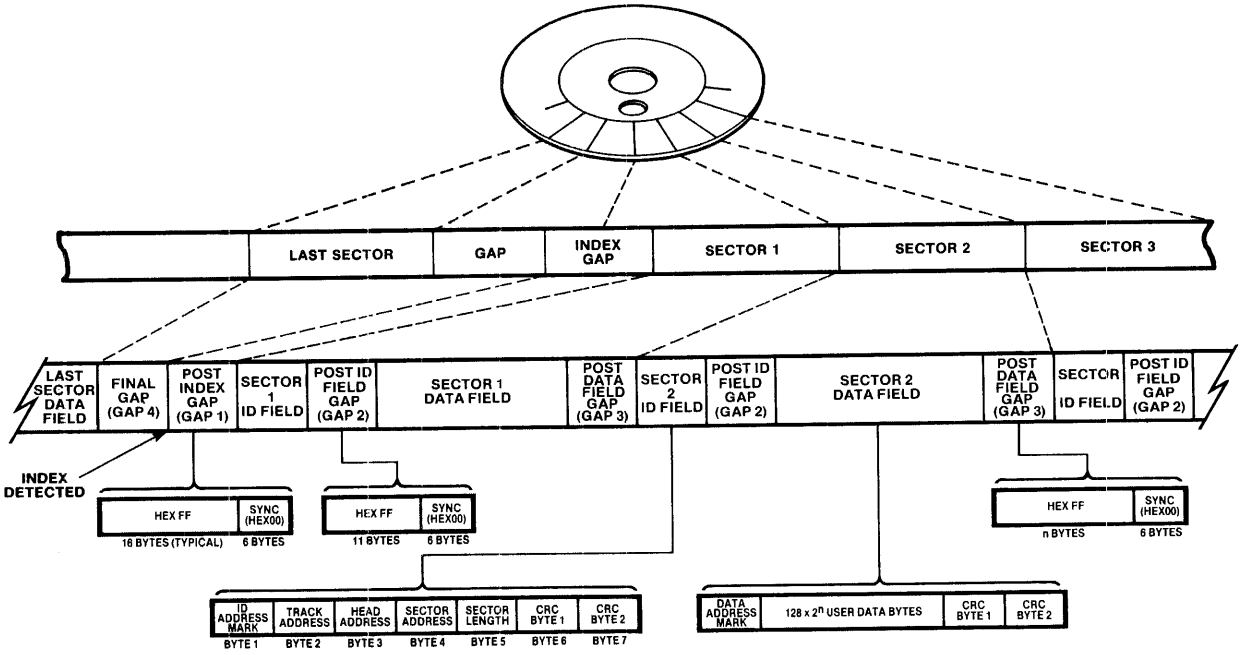


NUMBER OF SECTORS	NUMBER OF BYTES										
	GAP 1		ID FIELD	GAP 2		DATA FIELD	GAP 3		GAP 4	GAP 5	
	*ONES	SYNC		ONES	SYNC		*ONES	SYNC		*ONES	SYNC
26	26	6	7	11	6	131	27	6	275	40	6
15	26	6	7	11	6	259	48	6	129	40	6
8	26	6	7	11	6	515	90	6	146	40	6
4	26	6	7	11	6	1027	224	6	236	40	6
2	26	6	7	11	6	2051	255	6	719	40	6
1	26	6	7	11	6	4099	0	0	1007	40	6

* Program Specified

5208 Bytes Per Track

Figure 3-3. Standard Diskette Track Format



NUMBER OF SECTORS	NUMBER OF BYTES								
	GAP 1		ID FIELD	GAP 2		DATA FIELD	GAP 3		GAP 4
	*ONES	SYNC		ONES	SYNC		*ONES	SYNC	
18	16	6	7	11	6	131	11	6	24
10	16	6	7	11	6	259	21	6	30
5	16	6	7	11	6	515	74	6	88
2	16	6	7	11	6	1027	255	6	740
1	16	6	7	11	6	2051	0	0	1028

* Program Specified

3125 Bytes Per Track

Figure 3-4. Mini Diskette Track Format

Table 3-1. Sector Formats

DATA BYTES PER SECTOR	MAXIMUM NUMBER OF SECTORS PER TRACK	
	STANDARD DISKETTE	MINI DISKETTE
128	26	18
256	15	10
512	8	5
1024	4	2
2048	2	1
4096	1	(Not Available)

- Pre-Index Gap. The pre-index gap (gap 5) is used only with standard diskettes and consists of a program-selectable number of bytes of ones followed by six bytes of zeros. The number of all-ones bytes is generally set for 40 (IBM compatibility). The pre-index gap is written only when the track is formatted.

- Index Address Mark. The index address mark consists of a clock byte of $D7_{16}$ and a data byte of FC_{16}. On a standard diskette, one index address mark is written on each track when the track is formatted (the mini diskette does not use an index address mark).

- Post Index Gap. The post index gap (gap 1) consists of a program-selectable number of bytes of ones followed by six bytes of zeros. With the standard diskette, 26 bytes of ones are generally specified (IBM compatibility) while with the mini diskette, 16 bytes of ones are typical. The six zero bytes are used during subsequent read or write operations to synchronize the data separation

logic with the data to be read from the ID field. The post index gap is written only when the track is formatted.

- Sector n ID Field. The sector n ID field consists of seven bytes and is written only when the track is formatted. The first byte is the ID address mark consisting of a clock byte of $C7_{16}$ and a data byte FE_{16}. The second, third, and fourth bytes are the track, head, and sector address, respectively, and the 5th byte is the sector length. The last two bytes are the 16-bit CRC (cyclic redundancy check) character for the sector n ID field. The ID address mark is supplied by the FDC. The track, head, sector address, and the sector length are supplied to the FDC by the program. The CRC character is derived, by the FDC, from the data in the first 5 bytes.

- Post ID Field Gap. The post ID field gap (gap 2) is identical for both the standard and mini diskette formats and consists of 17 bytes (fixed). The first 11 bytes are all-one bytes and the last bytes are all -zero bytes. Gap 2 is written initially when the track is formatted. During subsequent write operations, the drives write circuitry is enabled at byte 12 and the six bytes of zero are written each time the sector is updated (write operation). During subsequent read operations, the last six bytes are used to synchronize the data separation logic with the incoming data field.

- Sector n Data Field. The length (number of data bytes) of the data field is determined by the program when the track is formatted. The first byte of the data field is the data address mark which consists of a clock byte of $C7_{16}$ and a data byte of FB_{16}. When a sector is to be deleted (i.e., bad spot on the diskette), a deleted sector address mark (clock byte $C7_{16}$, data byte $F8_{16}$) is written in place of the data address mark. The last two bytes of the data field comprise the CRC character.

- Post Data Field Gap. The post data field gap (gap 3) is written when the track is formatted and separates the preceding data field from the next physical ID field on the track. Note that a post data field gap is not written following the last physical sector on a track. The gap itself consists of a program-selectable number of all-one bytes followed by six all-zero bytes. Following a sector update (write), operation, the drives write logic is disabled during the first-all ones byte.

- Final Gap. The final gap (gap 4) is written when the track is formatted and consists of all-one bytes extending from the last physical data field on the track to the physical index mark. The length of the gap is dependent on both the number of bytes per sector specified and the lengths of the program-selectable gaps specified.

3-3. I/O PORT COMMANDS

CPU communications with the interface is accomplished through a 4-bit I/O port address block as defined by the four least significant bits of the I/O address (ADR0/-ADR3/). The location of this block is determined by the interfaces I/O base address, which is selectable by the user, through a switch on the interface board. The switch value selected corresponds to the eight most significant I/O address bits (ADR4/-ADRB/).

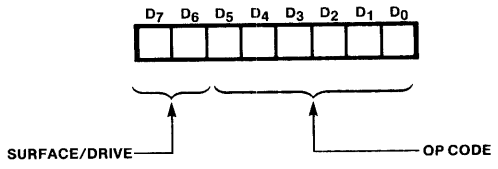
The resident program executes an I/O port read or write instruction at one of the locations within the I/O port address block to transfer information either to or from the interface. Table 3-2 defines the interface commands associated with each I/O port address within the block. Note that each command execution transfers one byte of data and that the *commands required* column refers to the number of times the command must be executed to transfer all associated command data. The application and format of each of the interface's I/O port commands are described in the following paragraphs.

3-4. WRITE FDC COMMAND REGISTER

The write FDC command register command loads (writes) a one byte command into the FDC's command register. (See Sections 3-15 to 3-36 for individual command descriptions.)

System address bus:
I/O BASE address + 0 (I/O Write)

System data bus:



3-5. READ FDC STATUS REGISTER

The Read FDC Status Register command reads the contents of the FDC's Status Register.

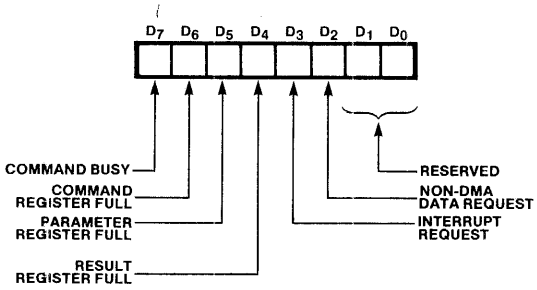
System address bus:
I/O BASE address + 0 (I/O Read)

Table 3-2. I/O Port Interface Commands

PORT ADDRESS (ADR0/-ADR3/)	COMMAND FUNCTION	COMMANDS REQUIRED
0	Write FDC command register or read FDC status register	1
1	Write FDC parameter register or read FDC result register	*
2	Reset FDC (write)	2
3	Not Used	—
4	Read or write DMAC starting memory address register	2
5	Read or write DMAC control register	2
6	Not Used	—
7	Not Used	—
8	Write DMAC mode set register	1
9	Select FDC (write)	1
A	Write DMAC I/O base memory address register	2
B	Not Used	—
C	Not Used	—
D	Not Used	—
E	Not Used	—
F	Reset interface (write)	1

*The read FDC result register command requires one data byte. The number of data bytes required with the write FDC parameter register command is dependent on the FDC command specified.

System data bus:



Bit 7: Command Busy. Set (logical 1) when a command byte is written to the command register and remains set until command processing is completed. Note that the FDC processes one command at a time.

Bit 6: Command Register Full. Set when a command byte is written to the command register, and cleared (logical zero) when the FDC begins command processing.

Bit 5: Parameter Register Full. Set when a parameter byte is written to the parameter register, and cleared when the parameter is accepted.

Bit 4: Result Register Full. Set when a result byte is present in the result register and cleared when the result byte is read by the CPU. Note that the data contained in the result register is valid only after command processing has been completed (when the command busy bit is clear). Reading the result register while a command is being processed, provides invalid data.

Bit 3: Interrupt Request. Reflects the state of the FDC INT Pin. It is set when the FDC requests attention as a result of the completion of an operation or the failure to complete an intended operation. This bit is cleared by a read result register I/O port command.

Bit 2: Non-DMA Data Request. This mode is not supported by the iSBC 204 interface.

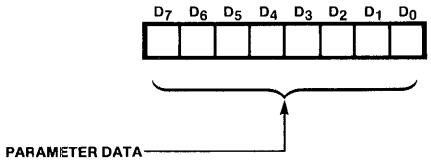
Bits 1, 0: Not used.

3-6. WRITE FDC PARAMETER REGISTER

The write FDC parameter register command writes a one byte parameter to the FDC's parameter register. The number of parameter bytes required is dependent upon the individual command and varies in number from 0 to 5 (refer to the individual FDC command descriptions for the number of parameters required in the parameter byte formats).

System address bus:
I/O BASE address + 1 (I/O Write)

System address bus

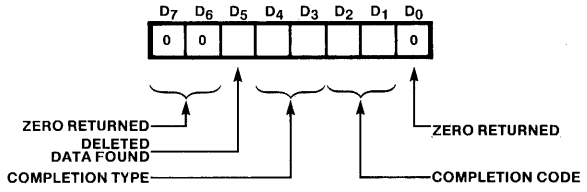


3-7. READ FDC RESULT REGISTER

Following an interrupt, the result register contains the result byte described following this paragraph. When the read FDC result register command is issued following a read special register or a read drive status FDC command, the result register contains the result byte described in Sections 3-28 and 3-29.

System address bus:
I/O BASE address + 1 (I/O Read)

System data bus:



Bits 7, 6: Not used. (Zero returned.)

Bit 5: Deleted Data Found. Set when a deleted data address mark is encountered during a diskette read operation.

Bit 4, 3: Completion Type. This field provides general information regarding the outcome of an operation.

BIT 4	BIT 3	INTERPRETATION
0	0	Normal Completion
0	1	System Error
1	0	Operator Intervention
1	1	Command/Drive Error

Bits 2, 1: Completion Code. This field provides more detailed information regarding the completion type.

COMPLETION TYPE		COMPLETION CODE		DEFINITION
BIT 4	BIT 3	BIT 2	BIT 1	
0	0	0	0	Successful Completion or Scan Not Found
0	0	0	1	Scan Comparison Found
0	0	1	0	Scan Range Found
0	0	1	1	Not Used
0	1	0	0	Clock Error
0	1	0	1	Late DMA
0	1	1	0	ID Field CRC Error
0	1	1	1	Data Field CRC Error
1	0	0	0	Drive Not Ready
1	0	0	1	Write Protect
1	0	1	0	Track 00 Not Found
1	0	1	1	Write Fault
1	1	0	0	Sector Not Found
1	1	0	1	Not Used
1	1	1	0	Not Used
1	1	1	1	Not Used

Bit 0: Not used. (Zero returned.)

Table 3-3 explains each individual completion code.

Three points should be noted regarding the structure of the Result byte.

1. An all zero result byte indicates that the specified operation was completed without error. This fact simplifies user programming by allowing a branch on zero result.

2. The encoding of the completion type field allows software to readily distinguish between possible recoverable errors and fatal (non-recoverable) errors. For example, the individual completion code errors defined under completion type 01 are all considered recoverable. Whenever a completion type 01 result is encountered, the program can retry the operation, without having to know the specific nature (completion code) of the error.

3. The bit positions allocated to the completion type and completion code, by not using bit position zero, allow two-byte addressing for a users error address table in memory. The result byte then can be used as a table offset to point to the first byte of the address of the corresponding error message or error handling routine in the table.

Table 3-3. Completion Code Interpretation

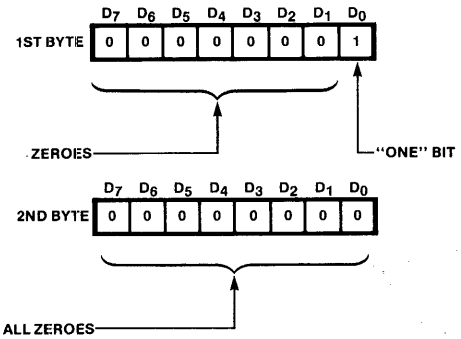
DEFINITION	INTERPRETATION
Successful Completion or Scan Not Found	The diskette operation specified was completed without error or the data pattern specified with the scan command was not found on the track addressed.
Scan Comparison Found	The data pattern specified with the scan command was found on the track addressed.
Scan Range Found	Depending on the scan type specified, a data pattern either equal to or greater than the data pattern specified or equal to or less than the data pattern specified was found on the track addressed. (Note that with these scan types, if the data pattern specified equals a pattern on the track, the scan comparison found completion code is returned.)
Clock Error	During a diskette read operation, a clock bit was missing (dropped). Note that this function is disabled when reading any of the ID address marks (which contain missing clock pulses). If this error occurs, the operation is terminated immediately and an interrupt is generated.
Late DMA	During either a diskette read or write operation, the data channel did not respond within the allotted time interval to prevent data from being overwritten. This error immediately terminates the operation and generates an interrupt.
ID Field CRC Error	The CRC word (two bytes) derived from the data read in an ID field did not match the CRC word written in the ID field when the track was formatted. If this error occurs, the associated diskette operation is prevented and no data is transferred.
Data Field CRC Error	During a diskette read operation, the CRC word derived from the data field read did not match the data field CRC word previously written. If this error occurs, the data read from the sector should be considered invalid.
Drive Not Ready	The drive addressed was not ready. This indication is caused by any of the following conditions: 1. Drive not powered up 2. Diskette not loaded 3. Non-existent drive addressed 4. Drive went not ready during an operation Note that this bit is cleared only through an FDC read drive status command (see Section 3-28).
Write Protect	A diskette write operation was specified on a write protected diskette. The intended write operation is prevented and no data is written on the diskette.
Track 00 Not Found	During a seek to track 00 operation, the drive failed to provide a track 00 indication after being stepped 255 times.
Write Fault	This error is dependent on the type of drive supported and indicates that the fault input to the FDC has been activated by the drive.
Sector Not Found	Either the sector addressed could not be found within one complete revolution of the diskette (two index marks encountered) or the track address specified did not match the track address contained in the ID field. Note that when the track address specified and the track address read do not match, the FDC automatically increments its track address register (stepping the drive to the next track) and again compares the track addresses. If the track addresses still do not match, the track address register is incremented a second time and another comparison is made before the sector not found bit is set.

3-8. WRITE FDC RESET REGISTER

The write FDC reset register command writes a data byte to the FDC's reset register. Two bytes are required to initiate an FDC reset. The first byte must contain a one in the lower order bit position (all other bits must be zero), and the second byte must be all zeros.

System address bus:
I/O BASE address + 2 (I/O Write).

System data bus:



The reset FDC command performs the following functions:

1. Any command in progress is terminated immediately, and the command and parameter registers are cleared.
2. All drive control signals are inactivated causing the drive to unload the head and idle at its current track location.
3. The status and result registers are cleared.
4. The mode register is set to $C0_{16}$.

NOTE

This places the FDC in the DMA Mode with a double actuator drive.

5. The FDC enters an idle state awaiting subsequent command entry.

The reset FDC command does not affect drive characteristics established with the FDC specify command or the contents of the FDC's bad track and current track registers.

Note that the interface can include (optional) a second FDC circuit. The two circuits are uniquely selectable through the select FDC I/O port command (see Section 3-13). A reset FDC command only affects the FDC circuit currently selected.

THIS OPERATION MUST BE ACTIVE FOR 10 OR MORE 8271 CHIP CLOCK CYCLES.

3-9. READ OR WRITE DMAC STARTING MEMORY ADDRESS REGISTER (16-BIT MEMORY ADDRESS)

The write DMAC starting memory address register command is used prior to initiating a diskette read or write operation to specify the starting memory address of the first data byte to be transferred to or from the diskette. It is also used with the scan operation to specify the starting memory address of the scan data pattern.

The read DMAC starting memory address register command is used following a diskette read or write operation to determine the last memory address accessed.

Two commands are required to completely read or write the DMAC's starting memory address register. The first command accesses the register's low-order byte and the second command accesses the register's high-order byte.

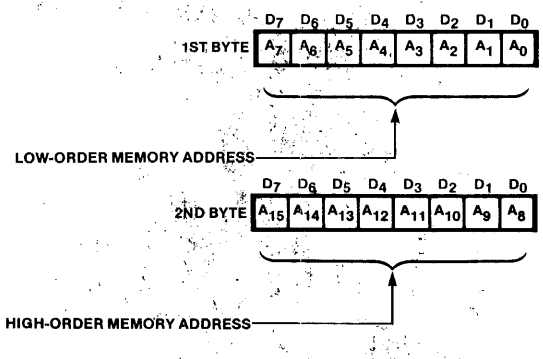
The DMA circuit contains a first/last flip-flop which toggles at the completion of each register write or read operation. The first/last flip-flop determines whether the upper or lower byte of the register is to be accessed. The first/last flip-flop is reset by the RESET input and whenever the mode set register is loaded. To maintain proper synchronization when accessing the registers, all register command operations should occur in pairs, with the lower byte of the register always being accessed first.

NOTE

In systems utilizing an interrupt structure, interrupts should be disabled prior to any paired programming operations to prevent an interrupt from splitting them. The result of such a split would leave the first/last flip-flop in the wrong state.

System address bus:
I/O BASE address + 4 (I/O Read or I/O Write).

System data bus:



3-10. READ OR WRITE DMAC STARTING MEMORY ADDRESS REGISTER (20-BIT MEMORY ADDRESS)

The iSBC 204 interface is capable of supporting Intel's 20-bit (1 megabyte) memory addressing architecture. This expanded address capability is realized by specifying a 16-bit base or segment address value which is then added to each normal 16-bit memory or offset address value. The segment and offset address values are displaced from one another by four bits and are added together as shown in figure 3-5. Four commands are required to completely read or write the DMAC starting memory address register when 20-bit addressing is used. The first command loads the low-order offset address byte into the register, and the second command loads the high-order offset address byte. The third command loads the low-order segment address byte into the register, and the fourth command loads the high-order segment address byte.

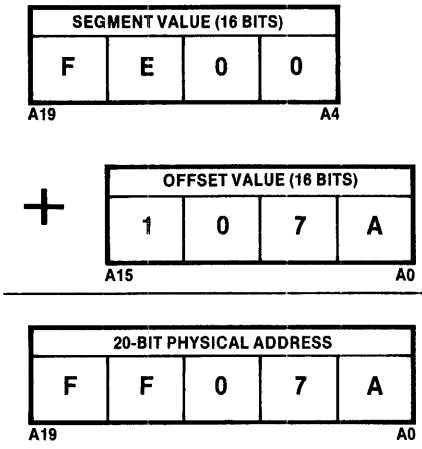
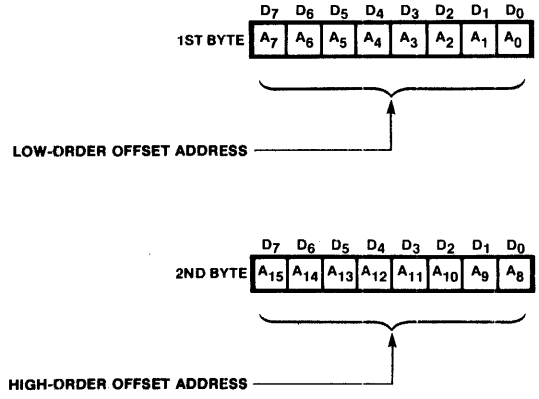


Figure 3-5. Segment-Offset Addition

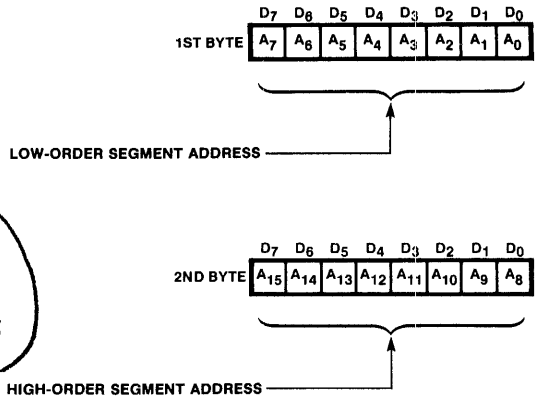
System address bus:
I/O BASE address + 4 (I/O Read or I/O Write)

System data bus:



System address bus:
I/O BASE address + A (I/O Write)

System data bus:



The displacement and addition of the segment and offset values are handled automatically by the interface.

3-11. READ OR WRITE DMAC CONTROL REGISTER

The write DMAC control register command is used to specify the DMA transfer mode (Read, Write, or Verify) for the intended diskette operation. This command is also used to load the scan pattern byte length (key length) into the registers terminal count field when using the DMAC's auto-load mode in conjunction with the FDC's scan command.

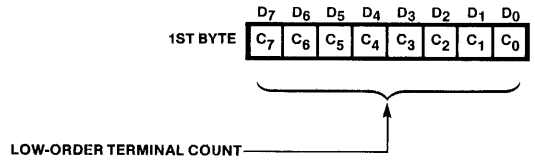
The read DMAC control register command is used to access the DMAC's terminal count register. Generally, this command is used to verify that the DMA transfer mode has been loaded correctly into the high-order byte of the register and, when using the auto-load mode, to verify the proper key length has been loaded into the terminal count field.

For routine (non-scan) data transfers between the diskette and memory, the terminal count field is not required (the number of data bytes to be transferred is programmed-selectable through the FDC). Two commands, however, must be issued to properly condition internal logic within the DMAC so that the second data byte (specifying the DMA transfer mode) is loaded into (or read from) the control register's high-order byte.

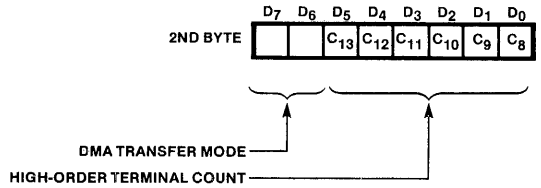
Note that the terminal count field can be used with routine data transfers and acts as a safeguard to prevent possible over-writing of dedicated memory in the event the FDC malfunctions. When using the terminal count field with routine transfers, the binary value loaded must be one less than the sector byte length.

System address bus:
I/O BASE address + 5 (I/O Read or I/O Write)

System data bus:



Bits 7-0: Low-Order Terminal Count. Used with the auto-load mode to specify the low order eight bits of the terminal count field.



Bits 7, 6: DMA Transfer Mode. Specifies the DMA transfer mode as follows:

BIT 7	BIT 6	DMA TRANSFER MODE	ASSOCIATED FDC COMMANDS
0	0	Verify DMA Cycle	Verify Data and Deleted Data
0	1	Write DMA Cycle	Read Data, Read Data and Deleted Data, Read ID
1	0	Read DMA Cycle	Write Data, Write Deleted Data, Format Track, Scan Data, Scan Data and Deleted Data
1	1	(illegal)	N/A

Verify DMA Cycle. Acknowledges FDC-initiated DMA request for each DMA cycle without actually transferring data to memory. This mode allows the FDC to read a sector from the diskette (to access the CRC character) without having to transfer the sector into memory.

Write DMA Cycle. Allows transfer of data from the diskette (diskette read operation).

Read DMA Cycle. Allows transfer of data from memory to the diskette (diskette write operation).

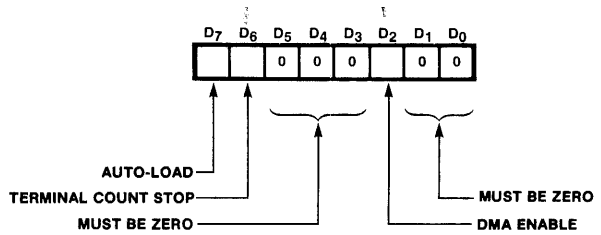
Bits 5-0: High order terminal count. Used with the auto-load mode to specify the high order six bits of the terminal count field. When used, the combined 14-bit terminal count value specified (bits 7-0 of the first byte and bits 5-0 of the second byte) must be one less than the byte length of the data pattern (key) to be compared.

3-12. WRITE DMAC MODE SET REGISTER

Writes a single byte to DMAC's mode set register.

System address bus:
I/O BASE address + 8 (I/O Write)

System data bus:



Bit 7: Auto-Load. This bit is used exclusively with scan operations and, when set, causes the contents of the DMAC scan pattern address and DMAC scan control registers to be loaded (non-destructively) into the DMAC starting memory address and DMAC control registers, respectively, whenever the last byte of a scan data pattern is transferred from memory. Since register updating only occurs each time a complete scan pattern is transferred, the pattern is compared repetitively with the data being read from the diskette.

Bit 6: Terminal Count Stop. This bit is only used when a terminal count value is specified for a routine data transfer and, when set, terminates the current DMA transfer and resets the DMA enable bit when the terminal count is reached (when the specified number of bytes have been transferred). The enable bit must be reprogrammed to continue or begin another DMA transfer.

Bits 5-3: Must be zero.

Bit 2: DMA Enable. This bit, when set, enables the DMAC to perform the operation specified by the DMA transfer mode field and the DMAC control register.

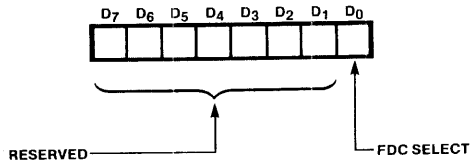
Bits 1, 0: Must be zero.

3-13. SELECT FDC

This command is used only when the interface includes two FDC circuits and it is desired to select (or deselect) the second FDC. (The second FDC is used to access the third and fourth drives in systems with more than two single-sided drives or to access the second drive when two double-sided drives are connected.) Bit 0 of the associated data byte determines which FDC will be selected.

System address bus:
I/O BASE address + 9 (I/O Read)

System data bus:



Bits 7-1: Reserved for future use.

Bit 0: FDC Select. If set, selects the second FDC or, if clear, selects the first FDC. Note that FDC selection remains in effect until the interface is reset (selecting the first FDC) or a subsequent select FDC command is issued to select or deselect the second FDC.

3-14. RESET INTERFACE

The reset interface command resets the interface to an initialized state. Note that this command is effective only through an I/O Write operation and that the associated data byte is not used.

System address bus:
I/O BASE address + F (I/O ~~Read~~ Write)

System data bus: Irrelevant

The extent of the reset interface command is as follows:

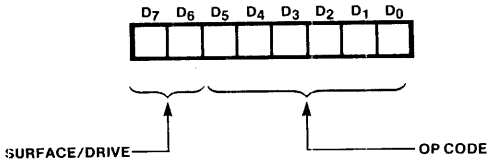
1. Immediately terminates any diskette operation or data transfer operation in progress.
2. Initializes the FDC select logic to select the FDC associated with connector J1.
3. Clears the DMAC's status and mode set registers (including the update flag).
4. Resets the first/last flip-flop to the first position.

Note that the FDC circuit(s) is not affected by the reset interface command and must be reset through a reset FDC I/O port command.

3-15. FDC COMMAND SET

The FDC command set is logically grouped into four basic command types: special data processing commands, special drive commands, routine read/write commands, and the scan commands.

All FDC commands are loaded into the selected FDC by a write FDC command register port command and exhibit the following byte format.



Bits 7, 6: Surface/Drive. These two bits select either one of the two single-sided drives associated with the selected FDC or select one of the two surfaces of the double sided drive associated with the selected FDC. (FDC selection is determined by the I/O port command, Select FDC.) Bit 6 corresponds to the FDC's Select 0 output and, when set, selects the first drive (drive 0 or drive 2) or top surface. Bit 7 corresponds to the select 1 output and, when set, selects the second drive (drive 1 or drive 3) or bottom surface.

Bits 5-0: Op Code. This field specifies the operation to be performed.

3-16. FDC COMMAND PROCESSING

The FDC command processing consists of three phases which are entered in the following sequence.

1. Command phase
2. Execution phase
3. Result phase.

3-17. COMMAND PHASE

During the command phase, the executing program in the CPU initiates command processing by writing a one byte command to the FDC's command register. Depending on the command specified (command op code), up to five bytes of support parameter data may be required from the CPU before the command execution phase can be entered. The parameter data is sent to the FDC parameter register, one byte at a time. The FDC reads the parameter resister, stores the data after each byte is written, and clears the parameter full bit. With commands that do not access the diskette, command processing is limited to only the command phase.

Figure 3-6 is a flow chart of the command phase and table 3-4 provides a typical subroutine for the command phase. Note that the flow chart shows that a command may not be issued if the FDC status register indicates the device is busy. Issuing a command while another command is in progress is illegal. The flow chart also shows a parameter buffer full check. The FDC status indicates the state of the parameter buffer. If a parameter is issued while the parameter buffer is full, the previous parameter is overwritten and lost.

3-18. EXECUTION PHASE

During the execution phase, the operation specified during the command phase is performed. During

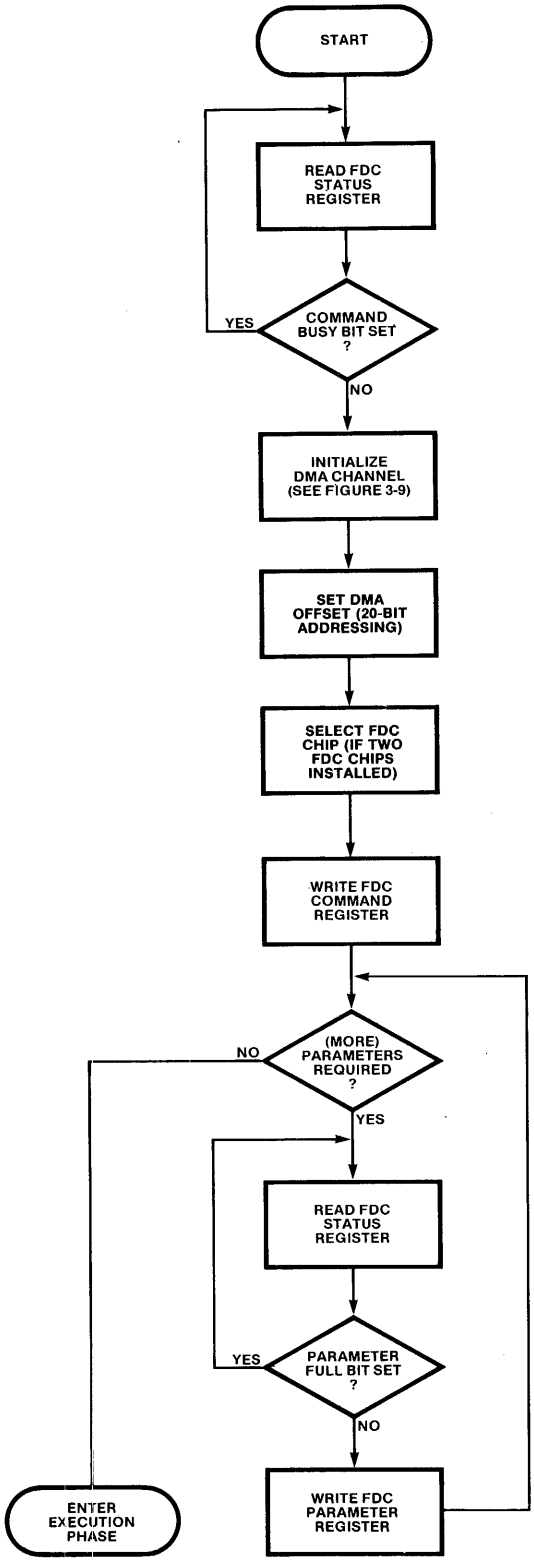


Figure 3-6. Command Phase Flow Chart

this phase, there is no CPU involvement since the interface utilizes DMA for the data transfers to and from the diskette. When execution is complete (or cannot be completed due to an error condition), the result phase is entered. Table 3-5 summarizes many of the basic execution phase characteristics associated with the individual FDC commands.

Figure 3-7 is a flow chart of the execution phase and table 3-6 provides a typical subroutine for the command phase.

- **Load Head.** Commands noted yes in the load head column of table 3-5, load the drive's read/write head against the diskette. Commands noted no in this column neither load nor unload (retract) the read/write head. After a head is loaded, it remains loaded until a specified number of diskette revolutions occur. The following delay times are associated with the

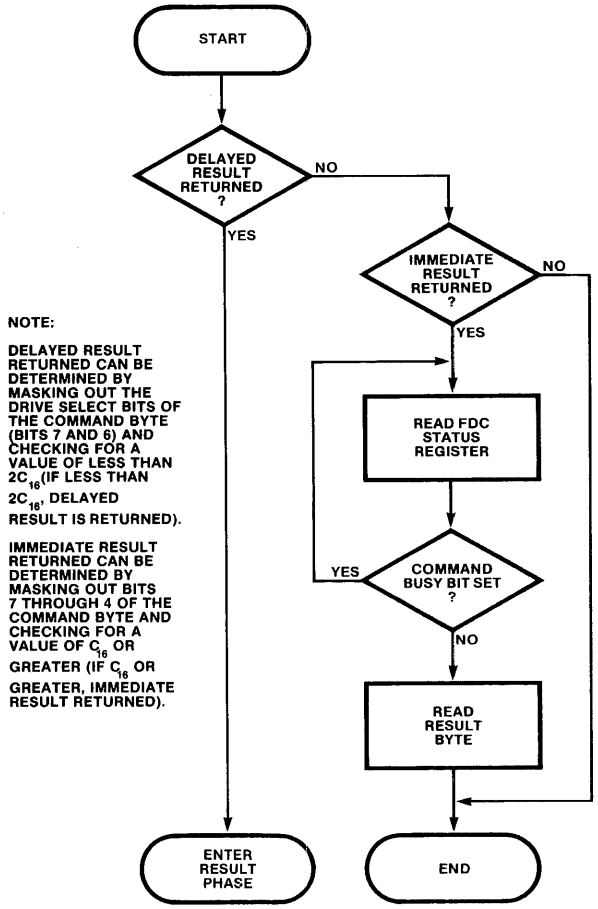


Figure 3-7. Execution Phase Flow Chart

Table 3-4. Typical Command Phase

```
;RDAT WRITES A COMMAND TO THE COMMAND REGISTER.
;EXAMPLE USES A ROUTINE READ COMMAND.
;BASE = I/O BASE ADDRESS SWITCH SETTING.
;READS A 128-BYTE SECTOR FROM DRIVE 1, TRACK 3, SECTOR 5.
;EXAMPLE ASSUMES 16-BIT ADDRESSING.
;EXAMPLE ASSUMES ONLY ONE FDC CHIP.

BASE    EQU     00H
        PUBLIC  RDAT
        EXTRN   EXCP
        EXTRN   DMAI

        CSEG
RDAT:   IN      BASE + 0    ;READ STATUS REGISTER
        ANI     80H         ;CHECK COMMAND BUSY BIT SET (D7)
        JNZ     RDAT        ;WAIT FOR NOT BUSY
; ********************************************************************
;                          INITIALIZE DMA CHANNEL
; ********************************************************************
        CALL    DMAI        ;INITIALIZE DMA CHANNEL
; ********************************************************************
;                          LOAD COMMAND REGISTER
; ********************************************************************
        MVI     A,92H       ;LOAD A REGISTER WITH READ COMMAND OP CODE
        OUT     BASE + 0    ;WRITE OP CODE TO FDC COMMAND REGISTER
RDAT1:  IN      BASE + 0    ;READ STATUS REGISTER
        ANI     20H         ;CHECK PARAMETER FULL BIT SET (D5)
        JNZ     RDAT1       ;WAIT FOR NOT FULL
        MVI     A,03H       ;LOAD A REGISTER WITH TRACK ADDRESS
        OUT     BASE + 1    ;WRITE TRACK ADDRESS TO FDC PARAMETER REGISTER
RDAT2:  IN      BASE + 0    ;READ STATUS REGISTER
        ANI     20H         ;CHECK PARAMETER FULL BIT SET (D5)
        JNZ     RDAT2       ;WAIT FOR NOT FULL
        MVI     A,05H       ;LOAD A REGISTER WITH SECTOR ADDRESS
        OUT     BASE + 1    ;WRITE SECTOR ADDRESS TO FDC PARAMETER REGISTER
RDAT3   IN      BASE + 0    ;READ STATUS REGISTER
        ANI     20H         ;CHECK PARAMETER FULL BIT SET
        JNZ     RDAT3       ;WAIT FOR NOT FULL
; ********************************************************************
;                          ENTER EXECUTION PHASE
; ********************************************************************
        JMP     EXCP        ;JUMP TO EXECUTION PHASE
        END
```

Table 3-5. Execution Phase Characteristics

COMMAND	CHARACTERISTIC						
	LOAD HEAD	CHECK READY	CHECK WRITE PROTECT	IMPLICIT SEEK	VERIFY SEEK	RESULT RETURNED	COMPLETION INTERRUPT
Scan Data	Yes	Yes	No	Yes	Yes	Yes	Yes
Scan Data and Deleted Data	Yes	Yes	No	Yes	Yes	Yes	Yes
Write Data	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Write Deleted Data	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Read Data	Yes	Yes	No	Yes	Yes	Yes	Yes
Read Data and Deleted Data	Yes	Yes	No	Yes	Yes	Yes	Yes
Read ID	Yes	Yes	No	Yes	No	Yes	Yes
Verify Data and Deleted Data	Yes	Yes	No	Yes	Yes	Yes	Yes
Format Track	Yes	Yes	Yes	Yes	No	Yes	Yes
Seek	Yes	Yes [1]	No	Yes	No	Yes	Yes
Read Drive Status	No	No	No	No	No	Note 2	No
Specify	No	No	No	No	No	No	No
Reset	No	No	No	No	No	No	No
Read SP Registers	No	No	No	No	No	Note 3	No
Write SP Registers	No	No	No	No	No	No	No

NOTES:
1. Checks READY at end of seek.
2. See Read Drive Status command.
3. See Read Special Register commands.

Table 3-6. Typical Execution Phase

```
;EXCP CHECKS THE COMMAND TO DETERMINE IF AN IMMEDIATE RESULT IS RETURNED
;OR A DELAYED RESULT IS RETURNED.
;IF DELAYED RESULT, EXAMPLE PROGRAM ENTERS RESULT PHASE.
;IF IMMEDIATE RESULT, EXAMPLE PROGRAM READS RESULT.

BASE    EQU     00H
        PUBLIC  EXCP
        EXTRN   MAIN
        EXTRN   RSLT

        CSEG
EXCP:   MVI     A,92H       ;LOAD A REGISTER WITH COMMAND OP CODE
; ********************************************************************
;                       CHECK FOR TYPE OF RESULT RETURNED
; ********************************************************************
        ANI     3FH         ;MASK OFF SELECT BITS
        SUI     2CH         ;SUBTRACT CONSTANT FROM COMMAND OP CODE

        JC      RSLT        ;IF CARRY SET COMMAND LESS THAN 2C
; ********************************************************************
;                   JUMP TO RESULT PHASE IF COMMAND LESS THAN 2C.
;                    INDICATES A DELAYED RESULT WILL BE RETURNED.
; ********************************************************************

; ********************************************************************
;                       CHECK FOR IMMEDIATE RESULT RETURNED.
; ********************************************************************
        MVI     A,92H       ;LOAD A REGISTER WITH COMMAND OP CODE
        ANI     OFH         ;MASK OFF BITS 7-4
        SUI     OCH         ;SUBTRACT CONSTANT FROM COMMAND OP CODE
        JC      MAIN        ;IF CARRY, NO RESULT RETURNED
; ********************************************************************
;                   JUMP BACK TO MAIN PROGRAM IF NO RESULT WILL BE RETURNED.
; ********************************************************************

; ********************************************************************
;                       PROCESS IMMEDIATE RESULT IF NO JUMP
; ********************************************************************
EXCP1:  IN      BASE + 0    ;READ STATUS REGISTER
        ANI     80H         ;CHECK COMMAND BUSY BIT SET
        JNZ     EXCP1       ;WAIT FOR NOT BUSY
        IN      BASE + 1    ;READ RESULT
        HLT                 ;RETURN TO MAINLINE PROGRAM WITH RESULT IN A
                            ;REGISTER
        END
```

state of the head (loaded or retracted) and the current track location.

HEAD STATE	SEEK REQUIRED	DELAY
Retracted	No	*Head Load Time
Retracted	Yes	Seek Time plus *Head Load Time
Loaded	No	None
Loaded	Yes	Head Settling Time
*Head Load Time includes Head Settling Time		

- **Check Ready.** Commands noted yes in the check ready column, check the state of the drive's ready line and are aborted immediately if the drive is not ready (or if the drive goes not ready during command execution). The seek command is the only command involving an actual drive operation that does not check the drive ready line before the operation is performed. The seek command checks the drive ready line after the seek is complete.

- **Check Write Protect.** All commands that write data on the diskette check the state of the drive's write protect line. These commands are marked with a yes in the right protect column on table 3-5. An intended write operation is aborted immediately (no data is written) if an attempt is made to write on a write protected diskette.

- **Implicit Seek.** All commands that transfer data, either to or from the diskette, have an implicit seek; that is, as part of command execution, the drive's read/write head is moved (if necessary) to the track specified prior to performing the intended operation. Those commands that have an implicit seek as part of command execution are marked with a yes in the implicit seek column of table 3-5.

- **Verify Seek.** Commands that verify the seek operation first read the sector ID field to determine if the drive has been stepped to the specified track and if the specified sector has been located. Those commands, that verify the seek, are listed with a yes in the verify seek column. When reading the ID field, the FDC first checks the CRC character and then compares the track and sector address bytes within the ID field with the track and sector addresses specified with the command.

- **Result Returned.** Commands noted yes in the result column of table 3-5 provide information regarding the outcome of an operation for examination by the CPU (read result register I/O port command). Note that while a seek command does have a result phase, a subsequent read or write operation must be performed to verify that the correct track was located.

- **Completion Interrupt.** Commands noted yes in the completion interrupt column of table 3-5, cause the FDC to generate an interrupt when command execution is complete (or cannot be completed).

3-19. RESULT PHASE

The result phase is entered following the execution phase in order to determine the results of the operation. Following execution of most commands, the interface generates an interrupt to inform the CPU of the completion of the execution phase. The CPU, when interrupted, must examine the interface's result register to determine the outcome of the operation.

When a Read Drive Status command or a Read Special Register command is issued the data is immediately placed into the result register and no interrupt is generated.

Figure 3-8 is a flow chart of the result phase and table 3-7 provides a typical subroutine for the result phase. Note that the flow chart shown in figure 3-7 reads the result byte immediately, without entering the result phase, when an immediate result is returned.

3-20. SPECIAL DATA PROCESSING COMMANDS

There are three commands that are catagorized as special data processing commands; Specify, Format Track, and Read ID.

3-21. SPECIFY COMMAND

The specify command must be used prior to performing any diskette operation (including the formatting of a new diskette) to define the drive's inherent operating characteristics and also is used following a formatting operation or installation of another diskette to define the locations of defective (bad) tracks. Since the specify command only loads internal registers within the FDC and does not involve an actual diskette operation. Command processing is limited to only the command phase. Note that once the operating characteristics and bad tracks have been specified for a given drive and diskette, it is necessary to redefine these values only if a diskette with different bad track locations is to be used or if the system is powered down (the reset FDC and reset interface commands do not affect the initial values defined).

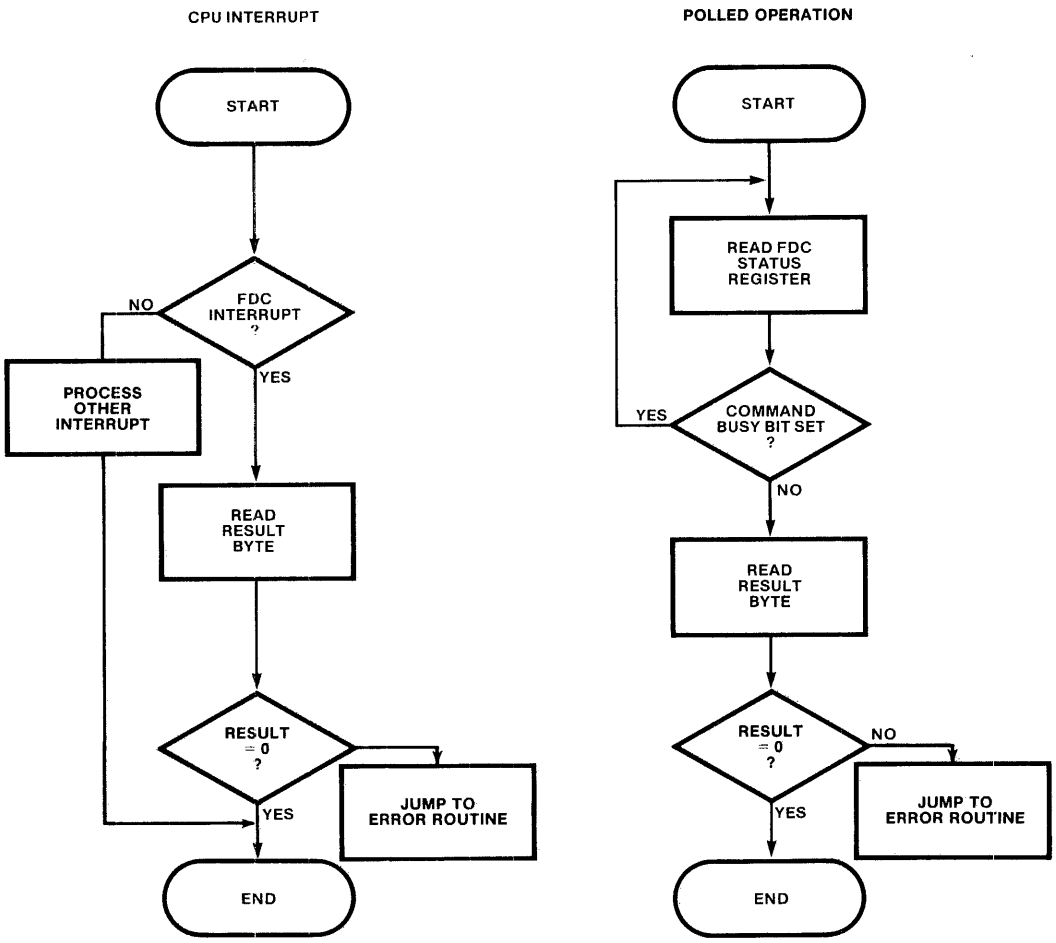


Figure 3-8. Result Phase Flow Chart

Table 3-7. Typical Result Phase

```
;RSLT CHECKS FDC STATUS REGISTER FOR A COMMAND BUSY CLEAR.
;READS THE RESULT BYTE AFTER COMMAND BUSY BIT IS CLEAR.

BASE    EQU     00H
        PUBLIC  RSLP
        EXTRN   ERROR

        CSEG
RSLT:   IN      BASE + 1        ;READ STATUS REGISTER
; ********************************************************************
;                               CHECK COMMAND BUSY BIT
; ********************************************************************
        ANI     80H             ;CHECK COMMAND BUSY BIT SET (D7)
        JNZ     RSLT            ;WAIT FOR NOT BUSY
; ********************************************************************
;                               READ AND CHECK RESULT BYTE
; ********************************************************************
        IN      BASE + 1        ;READ RESULT BYTE
        ORA     A               ;SET FLAGS
        JNZ     ERROR           ;IF NON-ZERO FLAG SET = ERROR
        JMP     MAIN            ;JUMP BACK TO MAIN PROGRAM
        END
```

The FDC command code is 35_{16}, and four parameter bytes are required to complete command processing. Note that the surface/drive bits (bits 7 and 6) of the command byte are both zero (operating characteristics are common for both drives or surfaces associated with the selected FDC, and bad tracks are uniquely defined for each drive or surface). The first parameter (parameter 0) defines the type of specify command as shown in the following table.

PARAMETER BYTE VALUE	SPECIFY COMMAND TYPE
$0D_{16}$	Define Drive Characteristics
10_{16}	Define Bad Tracks Drive/Surface 0
18_{16}	Define Bad Tracks Drive/Surface 1

3-22. DRIVE CHARACTERISTIC PARAMETERS

Parameter 0: $0D_{16}$ - Define drive characteristics.

Parameter 1: Step Rate. Defines the interval between step pulses issued by the FDC (track-to-track access time). The step rate either is specified in 1 ms steps ranging from 0 to 255 ms (standard-sized drives) or in 2 ms steps ranging from 0 to 510 ms (mini-sized drives). When zero is entered, the count mode is enabaled, and the step rate is controlled by an external clock signal. Consult the drive manufacturer's specifications for the track-to-track access time.

Parameter 2: Head Settling Time. Defines the interval that the FDC waits after stepping to the addressed track (with the head loaded) before initiating a diskette read or write operation. The head settling time either is specified in increments of 1 ms ranging from 0 to 255 ms (standard-sized drives) or in 2 ms increments ranging from 0 to 510 ms (mini-sized drives).

Parameter 3: Index Count/Head Load Time. The index count field (bits 7-4) specifies the number of diskette revolutions to occur, following an operation, before the FDC unloads (retracts) the drive's read/write head and deselects the drive. The number of revolutions ranges from 1 to 14. If zero is specified, the head is unloaded immediately following the operation while if 15 is specified, the head remains loaded until the FDC is reset.

The head load time field (bits 3-0) specifies the time interval that the FDC waits after loading the head before initiating a read or write operation. Note that head load time includes provision for head settling time. Head load time either is specified in increments of 4 ms ranging from 0 to 60 ms (standard-sized drives) or in increments of 8 ms ranging from 0 to 120 ms (mini-sized drives).

3-23. BAD TRACK PARAMETERS

Parameter 0: 10_{16} - location for bad tracks for drive/surface 0 or 2.
18_{16} - location of bad tracks for drive/surface 1 or 3.

Parameter 1: First Bad Track. Specifies the track address (track number) of the first defective track. If there are no bad tracks on the specified drive/surface, this parameter must be set to FF_{16}.

Parameter 2: Second Bad Track. Specifies the track address of the second defective track. Note that if two bad tracks are associated with the specified drive/surface, the track address specified in parameter 1 must be a lower number than the track address specified in parameter 2. If there is no second bad track, this parameter must be set to FF_{16}.

Parameter 3: Current track address (if known), otherwise FF_{16}.

The flow chart shown in figure 3-9 and table 3-8 provide a typical subroutine for locating the bad tracks on a new diskette.

3-24. FORMAT TRACK COMMAND

The format track command formats or "initializes" a track by writing the ID field, gaps, and address marks for each sector. The track to be formatted is defined by the command's associated track address parameter (parameter 0). Prior to command execution, the user must prepare a table in memory containing the ID field values (track, head, sector addresses, and sector length) for each sector on the track. During command execution, the interface accesses the table and, using the values supplied, writes each sector on the track. The ID field address mark originates from the FDC and is written automatically as the first byte of each sector's ID field. The track, head and sector address bytes and the sector length byte are taken, in order, from the table. The ID field CRC character is derived from the data written in the first five bytes and is written as the last two bytes of the ID field. Gaps are written automatically by the FDC with the lengths of the variable gaps determined by command parameters. The data field address mark is written automatically as the first byte of the data field. Data pattern $E5_{16}$ is written into each data byte of each sector's data field, and a CRC character is derived from the address mark and data written in the field and is appended to the last data byte.

As previously mentioned, the order of sector number assignment is taken directly from the formatting table in memory. The memory location of the first entry in the table is specified by the DMAC starting memory address I/O port command. Four entries are required for each sector: a track address, a head

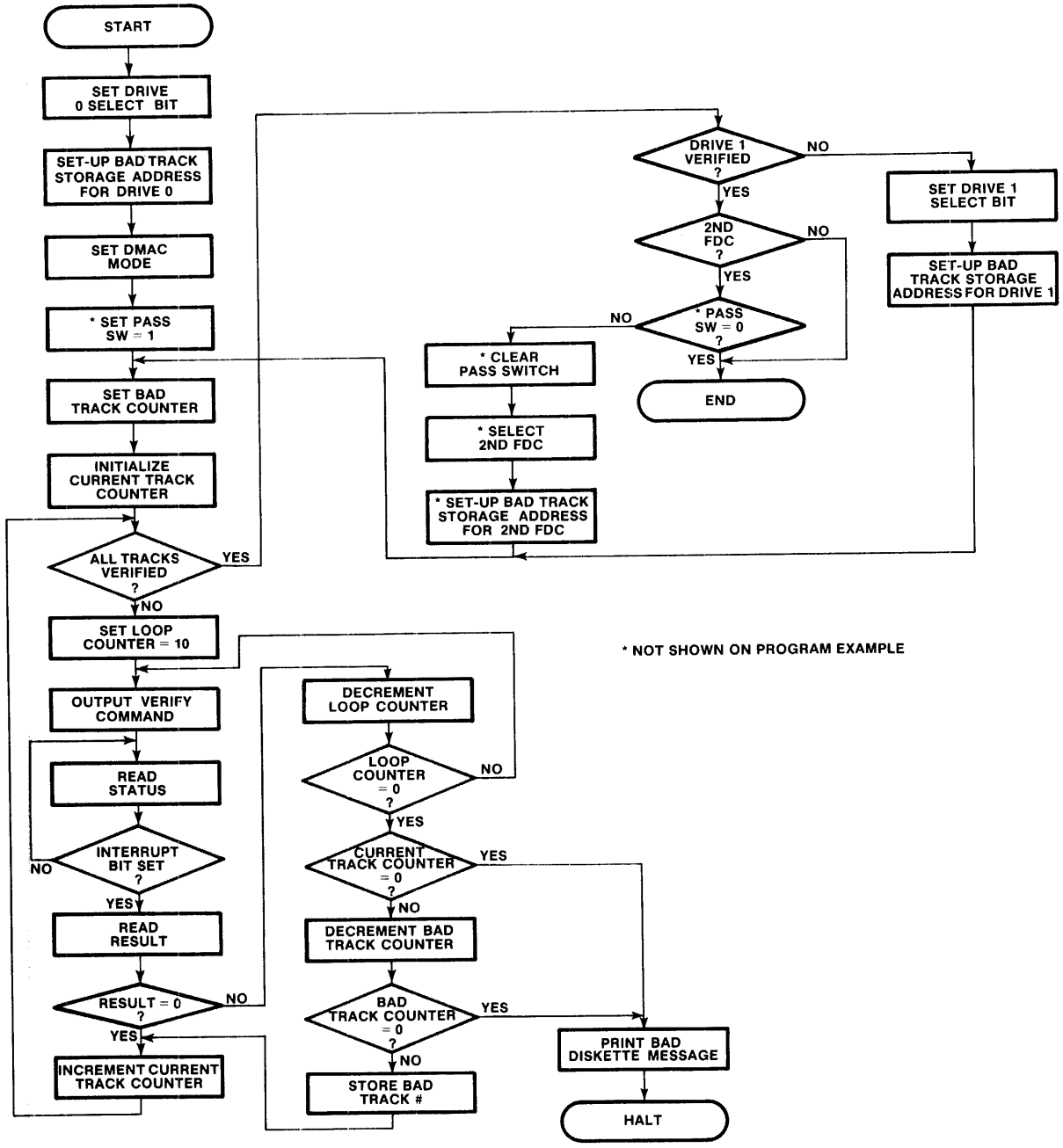


Figure 3-9. Locating Bad Tracks Flow Chart

Table 3-8. Typical Bad Track Location Routine

```
;EXAMPLE ASSUMES IBM COMPATIBLE FORMAT.
;EXAMPLE USES ONE FDC AND TWO SINGLE-SIDED DRIVES.
;EXAMPLE USES 128-BYTE SECTOR, 26 SECTORS PER TRACK, 77 TRACKS.
;MAXIMUM TWO BAD SECTORS PER DISKETTE.
;TEN TRYS TO VERIFY BEFORE RECORDING BAD TRACK.
;
;REGISTER USAGE:
  ;B = TRACK ADDRESS.
  ;C = BAD TRACK COUNTER.
  ;D = LOOP COUNTER.
  ;E = SELECT BIT.
  ;H,L = BAD TRACK STORAGE ADDRESS
;
;BAD TRACK STORAGE:
  ;BADTKS = DRIVE 0 BAD TRACK.
  ;BADTKS + 1 = DRIVE 0 BAD TRACK.
  ;BADTKS + 2 = DRIVE 1 BAD TRACK.
  ;BADTKS + 3 = DRIVE 1 BAD TRACK.
;
;
BASE      EQU       00H
          PUBLIC    BADT
          EXTRN     LDBTK
          EXTRN     PRINT0

          EXTRN     PRINTBD
;
;
;*****************************************************************************
;                          RESERVE MEMORY SPACE FOR BAD TRACK STORAGE
;*****************************************************************************
;
BADTKS:   DS        04H           ;BAD TRACK STORAGE
;
;*****************************************************************************
;                                  SET DRIVE 0 SELECT BIT
;*****************************************************************************
;
BADT:     MVI       E,40H         ;LOAD E WITH DRIVE 0 SELECT
;
;*****************************************************************************
;                              SET UP BAD TRACK STORAGE ADDRESS
;*****************************************************************************
          LXI       H,BADTKS      ;LOAD H AND L REGISTERS WITH BAD TRACK STORAGE ADDRESS
          XRA       A             ;CLEAR A REGISTER
;*****************************************************************************
;                                      SET DMAC MODE
;*****************************************************************************
          OUT       BASE + 5      ;WRITE VERIFY MODE TO DMAC
          OUT       BASE + 5      ;SECOND PARAMETER
BADT1:    MVI       C,03          ;SET BAD TRACK COUNTER
          XRA       A             ;CLEAR A REGISTER
          MOV       B,A           ;INITIALIZE CURRENT TRACT COUNTER
BADT2:    MOV       A,B           ;MOVE CURRENT TRACK COUNTER INTO A REGISTER
          CPI       4DH           ;COMPARE CURRENT TRACK TO LAST
          JZ        BADT10        ;IF ZERO, ALL TRACKS CHECKED-DO NEXT DRIVE
          MVI       D,0AH         INITIALIZE LOOP COUNTER TO 10
```

Table 3-8. Typical Bad Track Location Routine (Cont'd)

```
BADT3:  IN      BASE + 0    ;READ STATUS REGISTER
        ANI     80H         ;CHECK COMMAND BUSY BIT SET (D7)
        JNZ     BADT3       ;WAIT FOR NOT BUSY
;*************************************************************************************
;                                   VERIFY TRACKS
;*************************************************************************************
        MVI     A,1FH       ;LOAD VERIFY DATA OP CODE IN A REGISTER
        ORA     E           ;APPEND SELECT BIT TO OP CODE
        OUT     BASE + 0    ;WRITE VERIFY DATA OP CODE TO COMMAND REGISTER
BADT4   IN      BASE + 0    ;READ STATUS REGISTER
        ANI     20H         ;CHECK PARAMETER FULL BIT SET (D5)
        JNZ     BADT4       ;WAIT FOR NOT FULL
        MOV     A,B         ;LOAD CURRENT TRACK INTO A REGISTER
        OUT     BASE + 1    ;WRITE TRACK ADDRESS TO COMMAND REGISTER
BADT5:  IN      BASE + 0    ;READ STATUS REGISTER
        ANI     20H         ;CHECK PARAMETER FULL BIT SET
        JNZ     BADT5       ;WAIT FOR NOT FULL
        MVI     A,01H       ;LOAD STARTING SECTOR INTO A REGISTER
        OUT     BASE + 1    ;WRITE STARTING SECTOR TO COMMAND REGISTER
BADT6:  IN      BASE + 0    ;READ STATUS REGISTER
        ANI     20H         ;CHECK PARAMETER FULL BIT SET (D5)
        JNZ     BADT6       ;WAIT FOR NOT FULL
        MVI     A,1AH       ;LOAD SECTOR LENGTH AND NUMBER OF SECTORS INTO A REGISTER
        OUT     BASE + 1    ;WRITE SECTOR LENGTH AND NUMBER OF SECTORS TO COMMAND REGISTER
BADT7:  IN      BASE + 0    ;READ STATUS REGISTER
        ANI     80H         ;CHECK COMMAND BUSY BIT SET (D7)
        JZ      BADT7       ;WAIT FOR NOT BUSY
;*************************************************************************************
;                                  CHECK FOR BAD TRACK
;*************************************************************************************
        IN      BASE + 1    ;READ RESULT REGISTER
        ORA     A           ;SET FLAGS
        JNZ     BADT9       ;IF NON-ZERO FLAG SET = BAD TRACK
;*************************************************************************************
;                                SETUP TO VERIFY NEXT TRACK
;*************************************************************************************
BADT8:  INR     B           ;INCREMENT CURRENT TRACK COUNTER
        JMP     BADT2       ;VERIFY NEXT TRACK
;*************************************************************************************
;                                   BAD TRACK ROUTINE
;*************************************************************************************
BADT9:  DCR     D           ;DECREMENT LOOP COUNTER
        JNZ     BADT3       ;TRY AGAIN-UP TO 10 RETRIES
        MOV     A,B         ;LOAD CURRENT TRACK COUNTER INTO A REGISTER
        ORA     A           ;SET FLAGS
        JZ      PRINT0      ;PRINT TRACK 0 BAD IF ZERO FLAG SET
        DCR     C           ;DECREMENT BAD TRACK COUNTER
        JZ      PRINTBD     ;IF ZERO FLAG SET, MORE THAN TWO BAD TRACKS PRINT BAD DISKETTE MESSAGE
        MOV     A,B         ;LOAD CURRENT TRACK INTO A REGISTER
        MOV     M,A         ;STORE BAD TRACK NUMBER
        INX     H           ;INCREMENT BAD TRACK STORAGE ADDRESS
        JMP     BADT8       ;VERIFY NEXT TRACK
BADT10: MOV     A,E         ;MOVE SELECT BIT INTO A REGISTER
        ANI     40H         ;CHECK SELECT BIT FOR DRIVE 0
;*************************************************************************************
;                                      FINISHED?
;*************************************************************************************
```

Table 3-8. Typical Bad Track Location Routine (Cont'd)

```
;
        JZ          LDBTK         ;IF ZERO-FINISHED
;
;*************************************************************************************
;                                  NOT FINISHED. VERIFY DRIVE ONE
;*************************************************************************************
        MVI         E,80H         ;LOAD E REGISTER WITH DRIVE 1 SELECT BIT
        LXI         H,BADTKS + 2  ;INITIALIZE BAD TRACK STORAGE FOR DRIVE 1
        JMP         BADT1         ;REPEAT FOR NEXT TRACK
        END
```

address, a sector address, and a sector length. Note that the order of sector entries in the table is the sequence in which sector numbers appear on the track when it is formatted and that the number of entry sets (track, head, sector address, and sector length) in the table must equal the number of sectors allocated to a track. Caution must be exercised when creating the formatting table since entries are not verified by the interface and it is possible to format a track with an illegal, redundant, or missing record number.

NOTE

The head address and sector length are written on the diskette but are not used by the FDC.

Since the sector number (sector address) 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 multi-sector transfers by permitting the number of sectors specified (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 (standard size diskette) 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 3-10), 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).

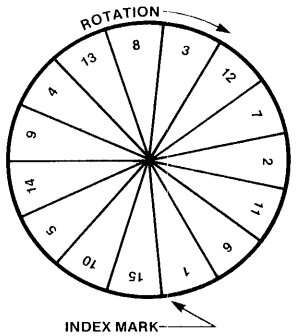


Figure 3-10. Sector Interleaving

Table 3-9 describes the organization of the formatting table that would be used to format the diskette shown in figure 3-10.

Note that the head address and sector length entries are not used by the FDC. These entries have been included for IBM format compatibility and may be user-defined.

The command code for the format track command is either 63_{16} (drive/surface 0 or 2) or $A3_{16}$ (drive/surface 1 or 3). Five parameters are required for command processing.

Table 3-9. Formatting Table

BYTE	FUNCTION	DATA CONTENTS
1	Track Address	XX_{16}
2	Head Address	XX_{16}
3	Sector Address 1	01_{16}
4	Sector Length	XX_{16}
5	Track Address	XX_{16}
6	Head Address	XX_{16}
7	Sector Address 6	06_{16}
8	Sector Length	XX_{16}
9	Track Address	XX_{16}
10	Head Address	XX_{16}
11	Sector Address 11	$0B_{16}$
12	Sector Length	XX_{16}
.		
.		
.		
53	Track Address	XX_{16}
54	Head Address	XX_{16}
55	Sector Address 10	$0A_{16}$
56	Sector Length	XX_{16}
57	Track Address	XX_{16}
58	Head Address	XX_{16}
59	Sector Address 15	$0F_{16}$
60	Sector Length	XX_{16}

Parameter 0: Track Address. Specifies the physical track number to be formatted. Legal values range from 00_{16} to $4C_{16}$ (0-76) for a standard diskette and from 00_{16} to 22_{16} (0-34) for a mini diskette. Note that this parameter must be identical to the track address byte entry in the formatting table.

Parameter 1: Gap 3 Length. Specifies the number of bytes of ones to be written in the post data field gap. For IBM 128 data byte sector format compatibility, 27 ($1B_{16}$) bytes are specified. (See figures 3-3 and 3-4 for gap lengths of non-128 data byte sectors.)

Parameter 2: Sector Length/Sectors Per Track. The sector length field (bits 7-5) specifies the number of data bytes to be written in each sector while the sectors per track field (bits 4-0) specifies the number of sectors on the track. The following table defines the correlation between the number of bytes per sector and the number of sectors per track.

SECTOR LENGTH FIELD			DATA BYTES PER SECTOR	SECTORS PER TRACK	
BIT 7	BIT 6	BIT 5		STANDARD	MINI
0	0	0	128	26	18
0	0	1	256	15	10
0	1	0	512	8	5
0	1	1	1024	4	2
1	0	0	2048	2	1
1	0	1	4096	1	N/A
1	1	0	8192	N/A	N/A
1	1	1	16,384	N/A	N/A

Parameter 3: Gap 5 Length. Specifies the number of bytes of ones to be written in the pre-index gap. For IBM 128 data byte sector format compatibility, 40 (28_{16}) bytes are specified. (See figure 3-3 for gap lengths of non-128 data byte sectors.) Note that with the mini-sized diskettes, there is no pre-index gap (zero is specified for this parameter).

Parameter 4: Gap 1 Length. Specifies the number of bytes of ones to be written in the post index gap. For IBM 128 data byte sector format compatibility, 26 ($1A_{16}$) bytes are specified. (See figures 3-3 and 3-4 for gap lengths of non-128 data byte sectors.)

Figure 3-11 is a flow chart of the format track routine and table 3-10 provides a typical subroutine for formatting tracks on the diskette.

3-25. READ SECTOR ID COMMAND

The read sector ID command transfers (reads) a specified number of sector ID fields from the diskette into memory. Only the track, head, sector address, and the sector length bytes of each ID field are transferred (the ID field address mark and the CRC bytes are not transferred although the CRC bytes are verified). As a prerequisite to command execution, the program must specify a DMAC starting memory address for the first sector ID field byte to be transferred. During command execution, the ID fields are read from the addressed track in sequential order beginning with the first sector ID field following the index mark. Note that the read sector ID command accesses the logical track address rather than the physical track location (the contents of the bad track registers are used in the calculation of the track address).

The command code for the read ID command is either $5B_{16}$ (drive/surface 0 or 2) or $9B_{16}$ (drive/surface 1 or 3). Three parameters are required for command processing.

Parameter 0: Track Address.

Parameter 1: Must be zero.

Parameter 2: Field Count. Specifies the number of sector ID fields to be transferred. The number specified must be in the range of 1 to the number of sectors on the track (0 is illegal).

3-26. SPECIAL DRIVE COMMANDS

Four commands are categorized as special drive commands; the seek command, the read drive status command, and the read and write special register commands.

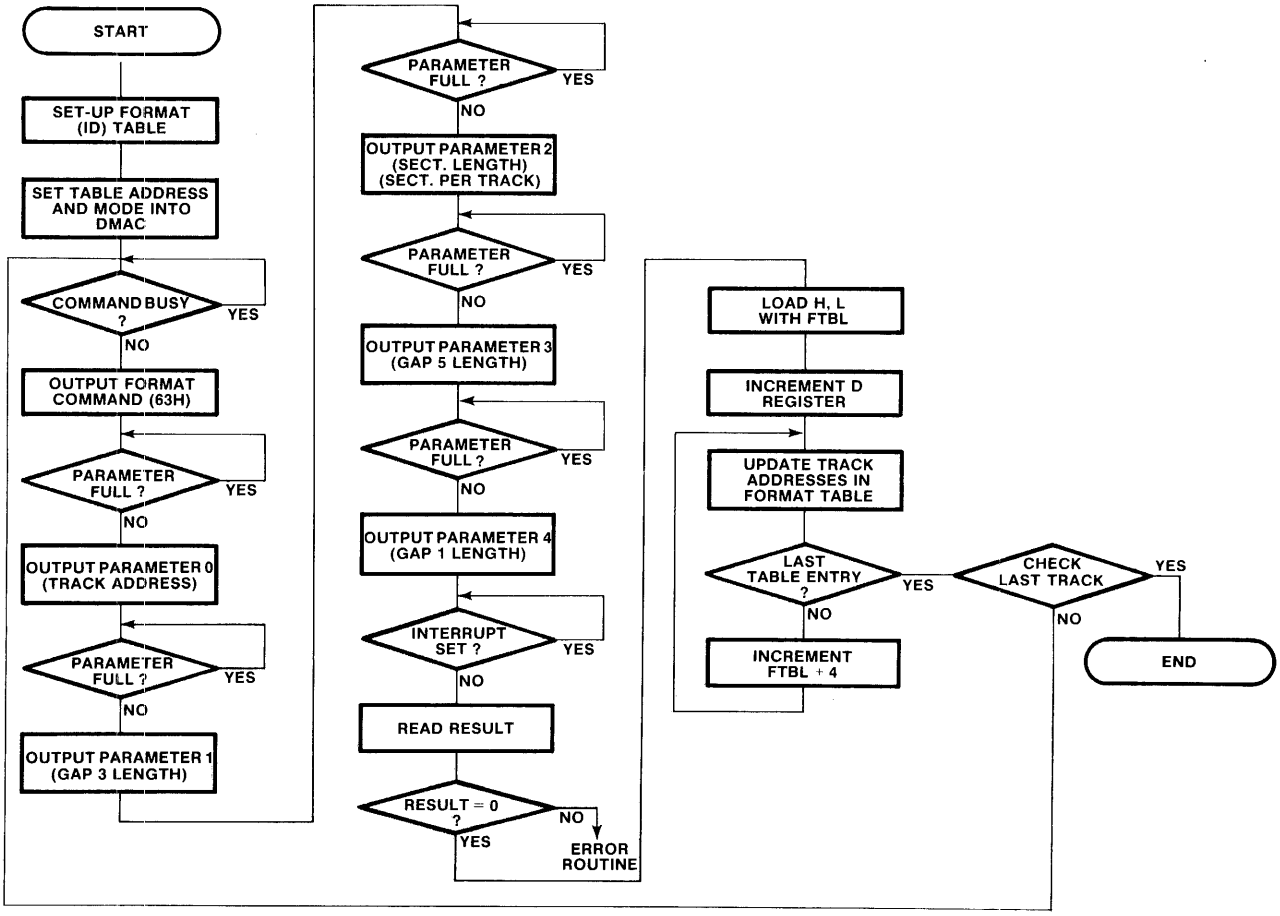


Figure 3-11. Formatting One Track on Drive 0

3-27. SEEK COMMAND

The seek command causes the drive's read/write head to be positioned over the track specified. The FDC determines the difference between the current track location and the track location specified, and steps the drive the corresponding number of tracks. When performing a seek operation, the FDC utilizes its internal bad tracks registers to determine the physical number of steps required to reach the specified track. (If a bad track exists between the current and specified track location, an additional step is inserted.) Note that with the seek command, no track address verification is performed (the sector ID field is not read).

NOTE

The head is loaded after the operation.

When the current track location is unknown (e.g., following an initial power-up sequence), a seek to track 0 command is used to initialize the FDC's internal current track register. When processing a seek to track 0 command, the FDC steps the drive's read/write head out (towards track 0) until the track 0 indication is received from the drive. If the track 0 indication is not received after 255 steps, the "track 0 not found" status bit is set in the result register.

The command code for the seek command is either 69_{16} (track/surface 0 or 2) or $A9_{16}$ (track/surface 1 or 3). Only one parameter is associated with the seek command.

Parameter 0: Track Address. Specifies the physical track location for the seek operation. Legal track addresses range from 00 to 76 (00_{16} to $4C_{16}$) for the

Table 3-10. Typical Format Track Routine

```
;FORM FORMATS ALL TRACKS ON A DISKETTE
;FTBL = FORMAT TABLE POINTER
;BASE = I/O BASE ADDRESS SWITCH SETTING
;FORMAT (FTBL) TABLE IS SETUP AT THE BEGINNING OF THIS ROUTINE.

BASE    EQU     00H
FTBL    EQU     4000H
        PUBLIC  FORM
        EXTRN   ERROR
        EXTRN   FORT
        EXTRN   FDAT
CONS    EQU     FTBL + 100

        CSEG
; ********************************************************************
;                       SETUP FORMAT TABLE IN MEMORY
; ********************************************************************
FORM:   CALL    FORT        ;SETUP FORMAT TABLE
        MVI     D,00H       ;CLEAR D REGISTER
; ********************************************************************
;                       SET TABLE ADDRESS AND MODE INTO DMAC
; ********************************************************************
FORMA:  MVI     A,LOW FTBL  ;LOW-BYTE OF TABLE ADDRESS
        OUT     BASE + 4    ;OUTPUT LOW-ORDER STARTING ADDRESS BYTE
        MVI     A,HIGH FTBL ;HIGH-BYTE OF TABLE ADDRESS
        OUT     BASE + 4    ;OUTPUT HIGH-ORDER STARTING ADDRESS BYTE
        XRA     A           ;CLEAR A REGISTER
        OUT     BASE + 5    ;LOAD LOW-ORDER DMAC CONTROL REGISTER BYTE
        MVI     A,80H       ;SET READ MODE VALUE
        OUT     BASE + 5    ;LOAD HIGH-ORDER DMAC CONTROL REGISTER BYTE
FORM1:  IN      BASE + 0    ;READ STATUS
        ANI     80H         ;CHECK COMMAND BUSY BIT SET (D7)
        JNZ     FORM1       ;WAIT FOR NOT BUSY
; ********************************************************************
;                   SEND FORMAT COMMAND AND PARAMETERS TO FDC
; ********************************************************************
        MVI     A,63H       ;LOAD FORMAT COMMAND INTO A REGISTER
        OUT     BASE + 0    ;WRITE FORMAT COMMAND TO FDC COMMAND REGISTER
FORM2:  IN      BASE + 0    ;READ STATUS
        ANI     20H         ;CHECK PARAMETER FULL BIT SET (D5)
        JNZ     FORM2       ;WAIT FOR PARAMETER NOT FULL
        MOV     A,D         ;LOAD TRACK ADDRESS INTO A REGISTER
        OUT     BASE + 1    ;OUTPUT TRACK ADDRESS
FORM3:  IN      BASE + 0    ;READ STATUS
        ANI     20H         ;CHECK PARAMETER FULL BIT SET (D5)
        JNZ     FORM3       ;WAIT FOR PARAMETER NOT FULL
        MVI     A,27H       ;LOAD GAP 3 LENGTH (27 BYTES) INTO A REGISTER
        OUT     BASE + 1    ;OUTPUT GAP 3 LENGTH
FORM4:  IN      BASE + 0    ;READ STATUS
        ANI     20H         ;CHECK PARAMETER FULL BIT SET (D5)
        JNZ     FORM4       ;WAIT FOR PARAMETER NOT FULL
        MVI     A,1AH       ;LOAD SECTOR LENGTH/SECTORS PER TRACK (128/26) TO A REGISTER
        OUT     BASE + 1    ;OUTPUT SECTOR LENGTH/SECTORS PER TRACK
FORM5:  IN      BASE + 0    ;READ STATUS
        ANI     20H         ;CHECK PARAMETER FULL BIT SET (D5)
        JNZ     FORM5       ;WAIT FOR PARAMETER NOT FULL
        MVI     A,40H       ;LOAD GAP 5 LENGTH (40 BYTES) INTO A REGISTER
        OUT     BASE + 1    ;OUTPUT GAP 5 LENGTH
FORM6:  IN      BASE + 0    ;READ STATUS
        ANI     20H         ;CHECK PARAMETER FULL BIT SET (D5)
        JNZ     FORM6       ;WAIT FOR PARAMETER NOT FULL
        MVI     A,26H       ;LOAD GAP 1 LENGTH (26 BYTES) INTO A REGISTER
        OUT     BASE + 1    ;OUTPUT GAP 1 LENGTH
```

Table 3-10. Typical Format Track Routine (Cont'd)

```
; ********************************************************************************
;                                WAIT FOR FORMAT TO COMPLETE
; ********************************************************************************
FORM7:  IN      BASE + 0        ;READ STATUS
        ANI     08H             ;CHECK INTERRUPT REQUEST BIT SET
        JZ      FORM7           ;WAIT FOR INTERRUPT
        IN      BASE + 1        ;READ RESULT REGISTER
        ORA     A               ;SET FLAG
        JNZ     ERROR           ;CHECK TYPE OF ERROR
; ********************************************************************************
;                        THE FOLLOWING ROUTINE UPDATES THE FORMAT TABLE
; ********************************************************************************
        LXI     H,FTBL          ;LOAD TABLE ADDRESS INTO H AND L REGISTERS
        INR     D               ;INCREMENT TRACK ADDRESS
LOOP:   MOV     M,D             ;WRITE TRACK ADDRESS INTO FORMAT TABLE
        LXI     B,CONS          ;LOAD CONSTANT INTO B AND C REGISTERS
        MVI     A,LOW CONS      ;LOAD LOW BYTE OF CONSTANT IN A REGISTER
        CMP     L               ;COMPARE LOW BYTE OF CONSTANT TO LOW BYTE OF FTBL POINTER
        JNZ     STEP            ;IF NON-ZERO, JUMP TO INCREMENT POINTER
        MVI     A,HIGH CONS     ;LOAD HIGH BYTE OF CONSTANT IN A REGISTER
        CMP     H               ;COMPARE HIGH BYTE OF CONSTANT TO HIGH BYTE OF FTBL POINTER
        JNZ     STEP            ;IF NON-ZERO, JUMP TO INCREMENT POINTER
        MVI     A,78            ;LOAD A REGISTER WITH NUMBER OF TRACKS + 1
        CMP     D               ;COMPARE D WITH NUMBER OF TRACKS + 1
        JNZ     FORMA           ;IF NON-ZERO, MORE TRACKS TO FORMAT
        JMP     RDAT            ;END OF FORMAT ROUTINE
STEP:
        INX     H               ;INCREMENT FTBL POINTER BY 4
        INX     H
        INX     H
        JMP     LOOP            ;JUMP TO LOOP TO CONTINUE UPDATING FORMAT TABLE
        END



;CREATE FORMAT TABLE
;H AND L = FTBL POINTER
;D = SECTOR ADDRESS

FTBL    EQU     4000H
        PUBLIC  FORT

        CSEG
FORT:   MVI     A,01H           ;LOAD SECTOR INTO A
        MOV     D,A             ;MOVE SECTOR INTO D
        LXI     H,FTBL          ;LOAD FTBL
FORT1:  MVI     M,00H           ;WRITE TRACK TO MEM
        INX     H               ;STEP FTBL
        MVI     M,00H           ;WRITE HEAD TO MEM
        INX     H
        MOV     M,D             ;WRITE SECTOR TO MEM
        INX     H
        MVI     M,00H           ;WRITE SECTOR LENGTH TO MEM
        INX     H
        INR     D
        MOV     A,D
        CPI     27
        JNZ     FORT1
        RET
        END
```

standard-sized diskette and from 00 to 34 (00_{16} to 22_{16}) for the mini-sized diskette.

CAUTION

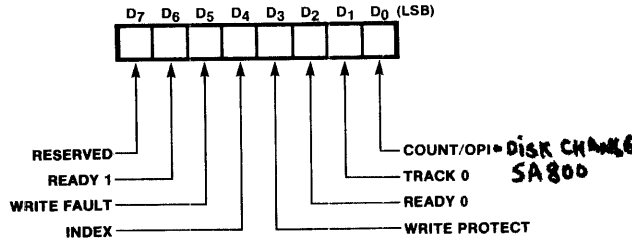
If a track address greater than 76 or 34 is specified, the FDC will continue to step the drive in an attempt to reach the nonexistent track. This could result in drive damage. If this occurs, the FDC must be reset and a seek to track 0 operation must be performed to re-initialize the FDC's current track register to a known location.

3-28. READ DRIVE STATUS COMMAND

The read drive status command examines the drive's incoming status lines and is used expressly to clear a "drive not ready" condition from within the FDC or to read the optional count input pin. When a drive is not ready (or when a drive goes not ready), this condition is retained in an internal latch within the FDC and is noted in the result register when a read result register I/O port command is executed. The read drive status command, in addition to presenting a data byte reflecting the states of the incoming drive status lines, enables the internal latch to again monitor the drive's ready line. After the drive is placed in a "ready" state, two subsequent read drive status commands must be issued to update the latch to reflect a "drive ready" indication in the result register.

The first read result register will indicate that the drive went not ready and will update the interval latch to the ready state. The second read will indicate the updated latch value.

The command code for the read drive status command is $6C_{16}$ (drive/surface 0 or 2) or AC_{16} (drive/surface 1 or 3). No parameters are required for command execution. The individual drive status bits returned are interpreted as follows:



Bit 7: Reserved for future use ("1" returned).

Bit 6: Ready 1. Indicates the current state of the internal "ready" latch associated with drive/surface 1. If set (logical "1"), indicates that the drive is in a "ready" state. If clear (logical "0"), indicates that the drive has gone "not ready" since the latch was last updated. Note that this bit and the Ready 0 bit (bit 2) are both returned irrespective of the drive/surface specified in the command code bits 7 and 8.

Bit 5: Write Fault. When in a logical "1" state, indicates that a drive-defined fault condition is present in the specified drive. Note that drives providing a write fault indication generally latch this line within the drive.

Bit 4: Index. When in a logic "1" state, indicates that an index pulse is present on the specified drive's index line.

Bit 3: Write Protect. When in a logical "1" state, indicates that a write protected diskette is installed in the specified drive.

Bit 2: Ready 0: Indicates the current state of the internal "ready" latch associated with drive/surface 0. If set, indicates that the drive is in a "ready" state and if reset, indicates that the drive has gone "not ready" since the latch was last updated.

Bit 1: Track 0. When in a logical "1" state, indicates that the specified drive's read/write head currently is positioned over track 0.

Bit 0: Count/OPI. Indicates the current state of the optional count input pin. USED ON SA800 TO DETECT DISK CHANGE SEE 2.19

Note that only the two ready bits are latched and that all other bits reflect the current states of their associated status lines when the read drive status command is processed. The read drive status command is not normally used to check the status. It is normally used to clear a "drive not ready" condition. The read result register I/O port command usually provides sufficient information on drive status.

3-29. READ AND WRITE SPECIAL REGISTER COMMANDS

The read and write special register commands are used to access special registers within the FDC. (The registers accessed are described in table 3-11.) The command code to read a special register is $7D_{16}$ (drive/surface 0 or 2) or BD_{16} (drive/surface 1 or 3), and the command code to write a special register is $7A_{16}$ (drive/surface 0 or 2) or BA_{16} (drive/surface 1 or 3).

The write special register command requires two parameters. The first parameter (parameter 0) specifies the address of the special register (see table 3-11), and the second parameter is the data to be written into the register.

The read special register command requires only a register address parameter. Following command execution, the contents of the special register read are

Table 3-11. Read/Write Special Registers

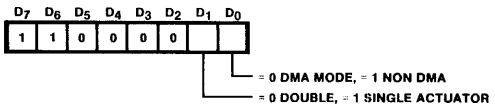
REGISTER NAME	REGISTER ADDRESS (IN HEXADECIMAL)	REGISTER CONTENTS
Scan Sector Number	06	The number of the sector in which the specified scan data pattern was located.
Scan Block Count	14	The number of 128-byte blocks remaining to be compared during a scan operation when the scan data pattern was located. This register initially contains a value equivalent to the number of 128-byte blocks, minus 1, allocated to a sector and is decremented with every 128 bytes read. Note that when scanning 128-byte sectors, the scan block count is zero. This register is used when scanning non-128 byte sectors to determine the number of bytes in a sector remaining to be scanned when the scan data pattern was located (see Appendix A).
Scan Byte Count	13	The number of bytes remaining to be compared in the current block of 128 bytes when the scan data pattern was located. This register is initialized to 128 at the beginning of a block and is decremented with each byte compared.
Drive/Surface 0 or 2 Current Track Address	12	The current track location of the read/write head on drive/surface 0 or 2. Note that the contents of this register should be considered valid only after a seek, read, or write operation.
Drive/Surface 1 or 3 Current Track Address	1A	The current track location of the read/write head on drive/surface 1 or 3. Note that the contents of this register should be considered valid only after a seek, read or write operation.
Mode Register SET TO $CØ BY WRITING FDC RESET REGISTER.	17	Writes the parameters to the mode register.  **Bits 6 and 7** Must be one **Bits 2-5** Must be set to zero (not used) **Bit 1** Double/Single Actuator: Selects single or double actuator mode. If the single actuator mode is selected, the FDC assumes that the physical track location of both disks is always the same. This mode facilitates control of a drive which has a single actuator mechanism to move two heads. **Bit 0** Data Transfer Mode: This bit selects the data transfer mode. If this bit is a zero, the FDC operates in the DMA mode (DMA Request/ACK). If this bit is a one, the FDC operates in non-DMA mode. When the FDC is operating in DMA mode, interrupts are generated at the completion of commands. If

Table 3-11. Read/Write Special Registers (Cont'd.)

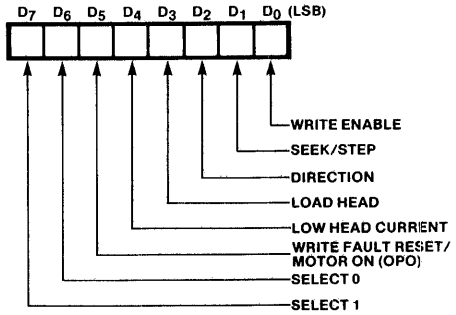
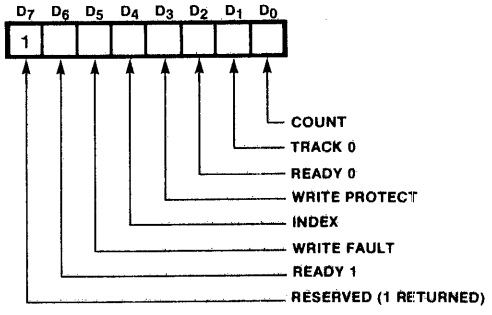
REGISTER NAME	REGISTER ADDRESS (IN HEXADECIMAL)	REGISTER CONTENTS
		the non-DMA mode is selected, the FDC generates an interrupt for every data byte transferred. NOTE Bits 0 and 1 are initialized to zero. The non-DMA mode is not supported by the iSBC 204 interface.
Output Port	23	Reads or writes the FDC's drive output port. Each bit of the register corresponds to the FDC output pin of the same name.  Other than for diagnostic purposes, this register is written to modify the state of the user-specified Write Fault Reset/Motor On bit (bit 5). On standard-sized drives with write fault detection logic, this bit is set to generate the write fault reset signal. This signal is used to clear a write fault indication within the drive. On mini-sized drives, this bit is set to turn on the drive motor prior to initiating a drive operation or is cleared to turn off the drive motor following an operation. If the register contents are read and stored prior to writing the register, the remaining bits can be restored to their previous state when the register is subsequently written to modify bit 5. Note that any time the register is read or written with the command select bits set to a different value than is currently in the register, the register will be cleared (drive deselected). Also note that when writing to the output port register to turn off the drive motor, the drive must first be deselected (to ensure that the head is unloaded) by a read command with the select bits set to zero.
Input Port	22	Reading this port will return the data that the FDC sees at the corresponding pins. Reading this port will update the drive not ready status, but will not clear the status. 

Table 3-11. Read/Write Special Registers (Cont'd.)

REGISTER NAME	REGISTER ADDRESS (IN HEXADECIMAL)	REGISTER CONTENTS
Drive/Surface 0 Bad Track Register 1	10	The track address of the first bad track associated with drive/surface 0. Note that if FF_{16} is returned (all "one" bits), no bad track was specified.
Drives/Surface 0 Bad Track Register 2	11	The track address of the second bad track associated with drive/surface 0.
Drive/Surface 1 Bad Track Register 1	18	The track address of the first bad track associated with drive/surface 1.
Drive/Surface 1 Bad Track Register 2	19	The track address of the second bad track associated with drive/surface 1.

present in the FDC's result register and are available to the program through a read result register I/O port command. Note that the special registers should not be written while an operation is in progress. Also, when reading special registers, the data will not be valid until the current operation is complete.

3-30. ROUTINE READ AND WRITE COMMANDS

The following commands are categorized as routine read and write commands.

- Read Data
- Read Data and Deleted Data
- Write Data
- Write Deleted Data
- Verify Data and Deleted Data

All of the above commands are available in two command formats; the standard command format in which a single, 128-byte sector is read or written and the special command format in which either multiple sectors (of any sector length) or single sectors of more than 128 bytes are read or written. The distinction between the two command formats is noted in the command op code (and also in the number of parameters required). Table 3-12 lists the command codes for the standard and special command formats. The standard command format requires two parameters and the special command format requires three parameters.

Parameter 0: Track Address. Specifies the track number in which the sector(s) to be read or written is located. Legal values range from 00_{16} to $4C_{16}$ (0 to 76) for a standard-sized diskette and from 00_{16} to 22_{16} (0 to 34) for a mini-sized diskette.

Parameter 1: Sector Address. Specifies the sector number to be read or written. In multi-sector transfers, the sector number specified is the first sector to be read or written, and the specified number of sectors (parameter 2) are processed consecutively.

Parameter 2: Sector Length/Number of Sectors. The sector length field (bits 7-5) is only used with the special command format and specifies the number of data bytes allocated to a sector as follows:

BIT			SECTOR LENGTH (IN BYTES)
7	6	5	
0	0	0	128
0	0	1	256
0	1	0	512
0	1	1	1024
1	0	0	2048
1	0	1	4096
1	1	0	8192
1	1	1	16,384

The number of sectors field (bits 4-0) is used only with multi-sector transfers and specifies the number of consecutive sectors to be read or written during the transfer (if zero is entered, a single sector is transferred). The maximum number of sectors that can be transferred is limited to the physical number of sectors on the track.

All of the routine read and write commands perform the functions noted in table 3-5 (see Section 3-18) in

Table 3-12. Standard and Special Command Format Codes

COMMAND	COMMAND CODE (IN HEXADECIMAL)			
	STANDARD COMMAND FORMAT		SPECIAL COMMAND FORMAT	
	DRIVE/ SURFACE 0	DRIVE/ SURFACE 1	DRIVE/ SURFACE 0	DRIVE/ SURFACE 1
Read Data	52	92	53	93
Read Data and Deleted Data	56	96	57	97
Write data	4A	8A	4B	8B
Write Deleted Data	4E	8E	4F	8F
Verify Data and Deleted Data	5E	9E	5F	9F

that the specific drive status lines are examined prior to beginning command execution, an implicit seek to the track addressed is performed, and the drive's read/write head is loaded. Regardless of the type of command specified (read, write, or verify), the FDC first reads the ID field(s) to verify that the correct track has been located and to locate the addressed sector. When a transfer is complete (or cannot be completed), the FDC sets the interrupt request bit in the status register and provides an indication of the outcome of the operation in the result register.

3-31. READ DATA COMMAND

The read data command transfers the data contents from the specified sector(s) on the diskette into memory. The read data command will not transfer a deleted sector. During a multi-sector transfer, if a deleted sector is encountered (deleted sector address mark read at the beginning of the data field), the sector is included in the total count of sectors read, but its contents are not transferred to memory. When a deleted sector is encountered, the deleted data found bit is set in the result register.

3-32. READ DATA AND DELETED DATA COMMAND

The read data and deleted data command is identical to the read data command with the exception that when a deleted sector is encountered, its contents are transferred to memory. As with the read data command, if a deleted sector is encountered, the deleted data found bit is set in the result register.

3-33. WRITE DATA COMMAND

The write data command permits a sector (or sectors) on the diskette to be updated (written) from memory.

3-34. WRITE DELETED DATA COMMAND

The write deleted data command is identical to the write data command with the exception that the FDC replaces the normal data address mark with a deleted sector address mark when the sector is updated.

3-35. VERIFY DATA AND DELETED DATA COMMAND

The verify data and deleted data command reads the specified number of normal and deleted sectors into the FDC (to verify the data field CRC bytes), but does not transfer the data to memory. Note that if a CRC error is detected during a multi-sector transfer, processing is terminated with the sector in error. The address of the failing sector can be determined by examining the scan sector number register using the read special register command (see Section 3-29).

3-36. SCAN COMMANDS

There are two scan commands: scan data and scan data and deleted data. Both commands are used to search a sector or multiple sectors for a specific data pattern or "key" from memory. The FDC's operation during a scan operation is unique in that data is read from memory and from the diskette simultaneously.

During the scan operation, the key is compared repetitively with the data read from the diskette (e.g., an eight byte key would be compared with the first eight bytes (1-8) read from the diskette, the second eight bytes (9-16), the third eight bytes (17-24). The scan operation is concluded when the key is located or when the specified number of sectors have been searched without locating the key. When concluded, the FDC requests an interrupt. The program must

then read the result register to determine if the scan was successful (if the key was located). If successful, several of the FDC's special registers can be examined (read special register command) to determine more specific information relating to the scan (i.e., the sector number in which the key was located, the number of bytes within the sector or block that were not compared and the number of sectors or blocks remaining to be compared when the key was located). Note that the special registers specifying the number of bytes and the number of blocks remaining to be scanned are based on a 128-byte sector. To interpret the contents of these registers with non-128 bytes sectors, see Appendix A.

The interface does not do a sliding scan, it does a fixed block linear search. This means the key in memory is compared to an equal length block in a sector; when these blocks meet the scan conditions the scan will stop. Otherwise, the scan continues until all the sectors specified have been searched.

The following factors regarding key length must be considered when establishing a key in memory.

1. When searching multiple sectors, the length of the key must be evenly divisible into the sector length to prevent the key from being split at subsequent sector boundaries. Since the character FF_{16} is not compared, the key in memory can be padded to the required length using this character. For example, if the actual pattern compared on the diskette is twelve characters in length, four bytes of FF_{16} would be appended to the key. Consequently, the last block of sixteen bytes compared within the first sector would end at the sector boundary, and the first byte of the next sector would be compared with the first byte of the key. Splitting data over sector boundaries will not work properly since the interface expects the start of a key at each sector boundary.

2. Since the first byte of the key is compared with the first byte of the sector, when the pattern does not begin with the first byte of the sector, the key must be offset using the character FF_{16}. For example, if the first byte of a nine byte pattern begins on the fifth byte of the sector, four bytes of FF_{16} are prefixed to the key (and three bytes of FF_{16} are appended to the key to meet the length requirement) so that the first actual comparison begins on the fifth byte.

The command codes for the two scan commands are as follows:

COMMAND	COMMAND CODE (IN HEXADECIMAL)	
	Drive/Surface 0 or 2	Drive/Surface 1 or 3
Scan Data	40_{16}	80_{16}
Scan Data and Deleted Data	44_{16}	84_{16}

Both scan commands require five parameters:

Parameter 0: Track Address. Specifies the track number containing the sectors to be scanned. Legal values range from 00_{16} to $4C_{16}$ (0 to 76) for a standard diskette and from 00_{16} to 22_{16} (0 to 34) for a mini-sized diskette.

Parameter 1: Sector Address. Specifies the first sector to be scanned. The number of sectors scanned is specified in parameter 2, and the order in which sectors are scanned is specified in parameter 3.

Parameter 2: Sector Length/Number of Sectors. The sector length field (bits 7-5) specifies the number of data bytes allocated to each sector (see parameter 2, routine read and write commands for field interpretation). The number of sectors field (bits 4-0) specifies the number of sectors to be scanned. The number specified ranges from one sector to the physical number of sectors on the track.

Parameter 3: Scan Type/Step Size. The scan type field (bits 7 and 6) specifies the comparison performed.

BIT		COMPARISON PERFORMED
7	6	
0	0	Equal. Compares each byte within the key for a corresponding equal byte within the sector. The scan stops after the first equal condition is met.
0	1	Greater Than. Compares the key for a corresponding pattern on the sector in which the binary value of the data on the diskette is either greater than or equal to the binary value of the key. The scan stops after the first greater than or equal condition is met.
1	0	Less Than. Compares the key for a corresponding pattern on the sector in which the binary value of the data on the diskette is either less than or equal to the binary value of the key. The scan stops after the first less than or equal condition is met.
1	1	Illegal.

The step size field (bits 5-0) specifies the sector offset (the number of sectors between the current sector being scanned and the next sector to be scanned). The address of the next sector to be scanned is derived by adding the step size to the current sector address. If a step size of one is specified, the next sequential sector is scanned.

Parameter 4: Field Length. Specifies the number of bytes to be compared (length of the key). While the range of legal values is from 0 to 255, the field length specified should be evenly divisible into the sector length to prevent the key from being split at sector boundaries. If a zero is used for the field length, 256 bytes are compared.

Figure 3-12 is an example of several scan commands.

3-37. INSTRUCTION SEQUENCES

When the drives are powered up, the initialization sequence, shown in the flow chart of figure 3-13, should be run to initialize the heads to track zero. Table 3-13 shows a typical initialization routine. The DMA initialization sequence shown in figure 3-14 should be run next. Table 3-14 lists a typical DMA initialization routine. If the diskettes that were installed are unformatted (new), the format sequence shown in figure 3-11 should be run. The drive and diskette(s) are now ready for operation.

Assume a step size of 01, a sector size of 256, and four sectors on track 1 with the following data:

Sector 01:	01	02	03	04	05	06	07	08	09	10	11	12	13	00	00
Sector 02:	A1	A2	A3	A4	A5	A6	A7	A8	A9	B1	B2	B3	B4	00	00
Sector 03:	1A	2A	3A	4A	5A	6A	7A	8A	9A	1B	2B	3B	4B	00	00
Sector 04:	F1	F2	F3	F4	F5	F6	F7	F8	F9	1F	2F	3F	4F	00	00

Parameter 3 Command [1]	Parameter 4 Field Length [2]	Parameter 1 Sector Address	Parameter 2 Number of Sectors	Key [3]
00	4	1	1	05,06,07,08
00	4	2	2	4A,5A,6A,7A
00	16	1	4	FF,FF,FF,FF,F5,F6,F7,F8,F9,FF,FF,FF,FF,FF,FF,FF
00	2	1	2	7A,8A
01	2	1	4	F9,1F
01	4	2	3	F2,F1,F7,FA
01	4	3	2	5A,5A,6A,7A
10	4	4	1	F7,F8,F9,FF
10	2	1	2	05,04
10	4	2	1	A5,A5,6A,7A

Completion Code [4]	Special Registers			Comments
	R06	R14	R13	
01	01	01	79H	Scan met on second field
00	X	X	X	Scan not met.
01	04	01	71H	Scan met on sector 4, first field.
00	X	X	X	Scan did not check sector 3, therefore scan was not met.
01	04	01	77H	Scan met in sector 4.
10	04	01	79H	Scan not met. Key field greater than disk field.
10	03	01	79H	Scan met. Key field less than disk field
10	04	01	7DH	Scan met equal.
10	01	01	7FH	Scan met. Key field greater than disk field.
10	02	01	7DH	Scan met. Key field less than disk field.

1. Command —
   00 = Equal
   01 = Greater than or equal
   10 = Less than or equal

2. Field Length — Each record is partitioned into a number of fields equal to the record size divided by the field length. Note that the record size should be evenly divisable by the field length to insure proper operation of multi record scan. Also, maximum field length = 256 bytes.

3. Key — The key is a string of bytes located in the user system memory. The key length should equal the field length. By programming the 8257 DMA Controller into the auto load mode, the key will be recursively read in by the chip (once per field).

4. Completion Code — Shows how Scan command was met or not met.
   00 — SCAN Not Met
   01 — SCAN Met Equal
   10 — SCAN Met Not Equal

5. Special Registers
   R06 — This register contains the record number where the scan was met.
   R14 — This register contains the MSB count and is decremented every 128 characters.

Length (ℓ) (D7-D5 of PAR 2)	Record Size	R14 = $2\ell - 1$ (Initialize at Beginning of Record
000	128 Bytes	0
001	256 Bytes	1
010	512 Bytes	3
011	1024 Bytes	7
•	•	•
•	•	•
•	•	•

R — 13 This register contains a modulo 128 LSB count which is initialized to 128 at beginning of each record. This count is decremented after each character is compared except for the last character in a pattern match situation.

Pointer = sector length — ((R14 contents) x 128 + (R13 contents))

Figure 3-12. Scan Command Example

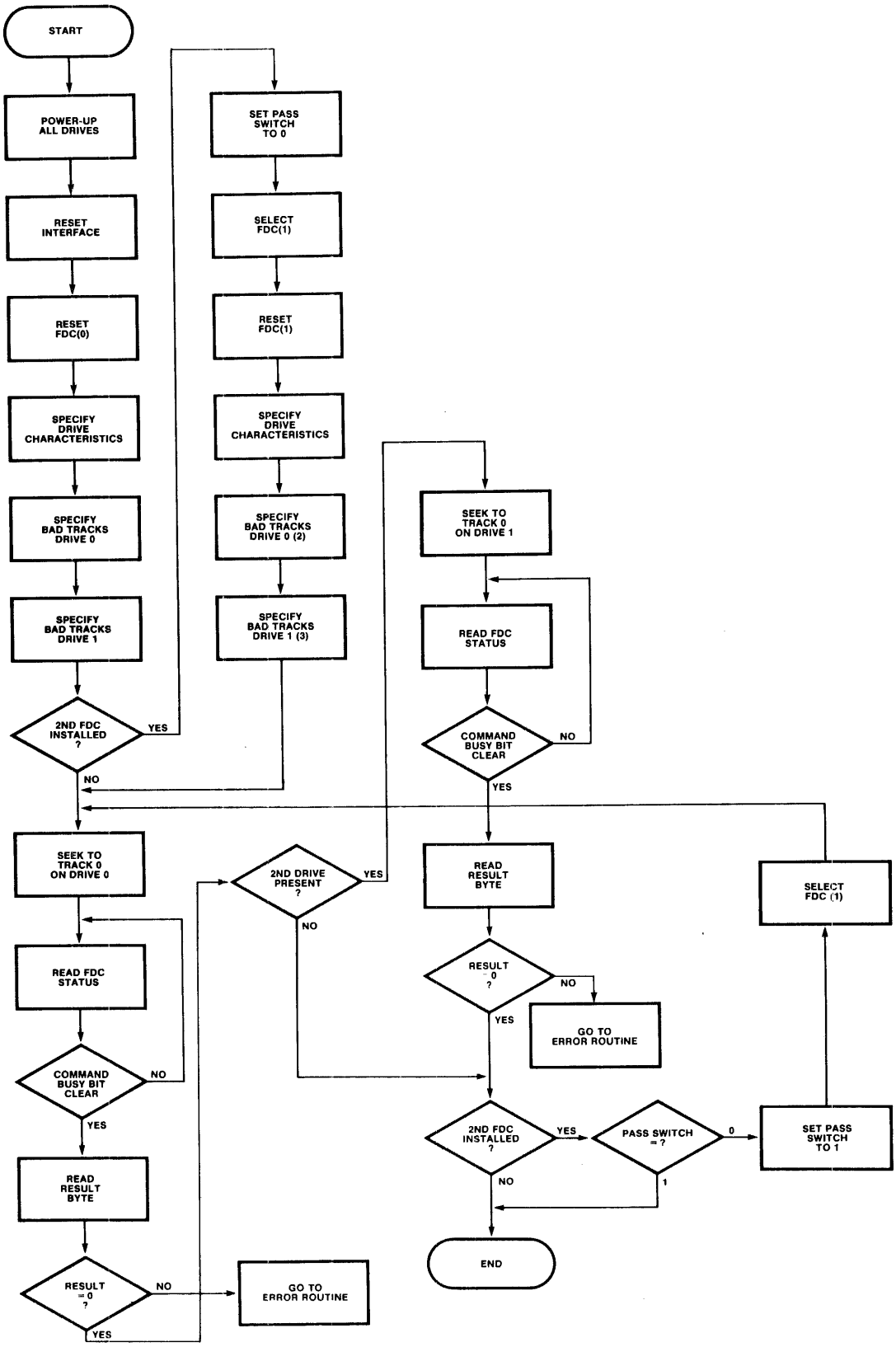


Figure 3-13. Initialization Flow Chart

Table 3-13. Typical Initialization Routine

```
;INIT RESETS THE INTERFACE AND FDC.
;SPECIFIES BAD TRACKS AND DRIVE CHARACTERISTICS.
;RECALIBRATES TWO DRIVES (SEEK TO TRACK 0).
;EXAMPLE USES ONE FDC AND TWO SHUGART 800 DRIVES.

BASE    EQU     00H
        PUBLIC  INIT
        EXTRN   ERR1
        EXTRN   ERR2
        EXTRN   DMAI
        EXTRN   FORM

        CSEG
; ***********************************************************************************************
;                                        RESET THE INTERFACE.
; ***********************************************************************************************
INIT:   OUT     BASE+0FH    ;INITIALIZE THE INTERFACE
; ***********************************************************************************************
;                                        RESET THE FDC CHIP.
; ***********************************************************************************************
        MVI     A,01H       ;LOAD A REGISTER WITH A ONE
        OUT     BASE+2      ;WRITE A ONE BYTE TO FDC TEST REGISTER
; ***********************************************************************************************
;                                        RESET TEST REGISTER.
; ***********************************************************************************************
        XRA     A           ;CLEAR A REGISTER
        OUT     BASE+2      ;WRITE A ZERO BYTE TO FDC TEST REGISTER
; ***********************************************************************************************
;                                     SPECIFY DRIVE CHARACTERISTICS.
; ***********************************************************************************************
        MVI     A,35H       ;LOAD SPECIFY COMMAND
        OUT     BASE+0      ;WRITE SPECIFY OP CODE TO COMMAND REGISTER
INIT1:  IN      BASE+0      ;READ STATUS REGISTER
        ANI     20H         ;CHECK PARAMETER FULL BIT SET (D5)
        JNZ     INIT1       ;WAIT FOR NOT FULL
        MVI     A,0DH       ;LOAD SPECIFY COMMAND (DEFINE DRIVE CHARACTERISTICS) IN A REGISTER
INIT2:  OUT     BASE+1      ;WRITE SPECIFY COMMAND TYPE TO COMMAND REGISTER
        IN      BASE+0      ;READ STATUS REGISTER
        ANI     20H         ;CHECK PARAMETER FULL BIT SET (D5)
        JNZ     INIT2       ;WAIT FOR NOT FULL
        MVI     A,08H       ;LOAD STEP RATE (8 MS) IN A REGISTER
INIT3:  OUT     BASE+1      ;WRITE STEP RATE TO PARAMETER REGISTER
        IN      BASE+0      ;READ STATUS REGISTER
        ANI     20H         ;CHECK PARAMETER FULL BIT SET (D5)
        JNZ     INIT3       ;WAIT FOR NOT FULL
        MVI     A,08H       ;LOAD HEAD SETTLING TIME (8 MS) IN A REGISTER
INIT4:  OUT     BASE+1      ;WRITE HEAD SETTLING TIME TO PARAMETER REGISTER
        IN      BASE+0      ;READ STATUS REGISTER
        ANI     20H         ;CHECK PARAMETER FULL BIT SET (D5)
        JNZ     INIT4       ;WAIT FOR NOT FULL
        MVI     A,59H       ;LOAD INDEX COUNT (5)/HEAD LOAD TIME (36 MS)
                            ;IN A REGISTER
        OUT     BASE+1      ;WRITE INDEX COUNT/HEAD LOAD TIME TO PARAMETER REGISTER
INIT5:  IN      BASE+0      ;READ STATUS REGISTER
        ANI     80H         ;CHECK COMMAND BUSY BIT SET (D7)
        JNZ     INIT5       ;WAIT FOR NOT BUSY
```

Table 3-13. Typical Initialization Routine (Cont'd)

```
; ********************************************************************************
;                    INITIALIZE BAD TRACK REGISTERS FOR ,DRIVE 0 AND 1, TO FF.
; ********************************************************************************
          MVI     A,35H       ;LOAD SPECIFY COMMAND IN A REGISTER
          OUT     BASE+0      ;WRITE SPECIFY OP CODE TO COMMAND REGISTER
INIT6:    IN      BASE+0      ;READ STATUS REGISTER
          ANI     20H         ;CHECK PARAMETER FULL BIT SET (D5)
          JNZ     INIT6       ;WAIT FOR NOT FULL
          MVI     A,10H       ;LOAD SPECIFY COMMAND TYPE (BAD TRACKS DRIVE 0)
                              ;IN A REGISTER
          OUT     BASE+1      ;WRITE SPECIFY COMMAND TYPE TO COMMAND REGISTER
          MVI     B,3H        ;SET B REGISTER TO THREE FOR COUNTING LOOPS
INIT7:    IN      BASE+0      ;READ STATUS REGISTER
          ANI     20H         ;CHECK PARAMETER FULL BIT SET (D5)
          JNZ     INIT7       ;WAIT FOR NOT FULL
          MVI     A,0FFH      ;LOAD FF IN A REGISTER
          OUT     BASE+1      ;WRITE FF TO PARAMETER REGISTER
          DCR     B           ;DECREMENT COUNT IN B REGISTER
          JNZ     INIT7       ;WAIT FOR ZERO (THREE LOOPS)
INIT8:    IN      BASE+0      ;READ STATUS REGISTER
          ANI     80H         ;CHECK COMMAND BUSY BIT SET (D7)
          JNZ     INIT8       ;WAIT FOR NOT BUSY
          MVI     A,35H       ;LOAD SPECIFY COMMAND IN A REGISTER
          OUT     BASE+0      ;WRITE SPECIFY OP CODE TO COMMAND REGISTER
INIT9:    IN      BASE+0      ;READ STATUS REGISTER
          ANI     20H         ;CHECK PARAMETER FULL BIT SET (D5)
          JNZ     INIT9       ;WAIT FOR NOT FULL
          MVI     A,18H       ;LOAD SPECIFY COMMAND TYPE (BAD TRACKS DRIVE 1)
                              ;IN A REGISTER
          OUT     BASE+1      ;WRITE SPECIFY COMMAND TYPE TO COMMAND REGISTER
          MVI     B,3H        ;SET B REGISTER TO THREE FOR COUNTING LOOPS
INIT10:   IN      BASE+0      ;READ STATUS REGISTER
          ANI     20H         ;CHECK PARAMETER FULL BIT SET (D5)
          JNZ     INIT10      ;WAIT FOR NOT FULL
          MVI     A,0FFH      ;LOAD FF IN A REGISTER
          OUT     BASE+1      ;WRITE FF TO PARAMETER REGISTER
          DCR     B           ;DECREMENT COUNT IN B REGISTER
          JNZ     INIT10      ;WAIT FOR ZERO (THREE LOOPS)
INIT11:   IN      BASE+0      ;READ STATUS REGISTER
          ANI     80H         ;CHECK COMMAND BUSY BIT SET (D7)
          JNZ     INIT11      ;WAIT FOR NOT BUSY
; ********************************************************************************
;                          RECALIBRATE DRIVE 0 (SEEK TO TRACK 0)
; ********************************************************************************
          MVI     A,69H       ;LOAD SEEK OP CODE IN A REGISTER
          OUT     BASE+0      ;WRITE SEEK COMMAND TO COMMAND REGISTER
INIT12:   IN      BASE+0      ;READ STATUS REGISTER
          ANI     20H         ;CHECK PARAMETER FULL BIT SET (D5)
          JNZ     INIT12      ;WAIT FOR NOT FULL
          XRA     A           ;CLEAR A REGISTER
          OUT     BASE+1      ;WRITE TRACK ZERO TO PARAMETER REGISTER
INIT13:   IN      BASE+0      ;READ STATUS REGISTER
          ANI     80H         ;CHECK COMMAND BUSY BIT SET (D7)
          JNZ     INIT13      ;WAIT FOR NOT BUSY
          IN      BASE+1      ;READ RESULT BYTE
          ORA     A           ;CHECK FOR ZERO RESULT BYTE
```

Table 3-13. Typical Initialization Routine (Cont'd)

```
            JNZ     ERR1        ;IF NON-ZERO GO TO ERROR ROUTINE
INIT14:     IN      BASE+0      ;READ STATUS REGISTER
            ANI     80H         ;CHECK COMMAND BUSY BIT SET (D7)
            JNZ     INIT14      ;WAIT FOR NOT BUSY
; ************************************************************************************
;                          RECALIBRATE DRIVE 1 (SEEK TO TRACK 0)
; ************************************************************************************
            MVI     A,0A9H      ;LOAD SEEK OP CODE IN A REGISTER
            OUT     BASE+0      ;WRITE SEEK COMMAND TO COMMAND REGISTER
INIT15:     IN      BASE+0      ;READ STATUS REGISTER
            ANI     20H         ;CHECK PARAMETER FULL BIT SET (D5)
            JNZ     INIT15      ;WAIT FOR NOT FULL
            XRA     A           ;CLEAR A REGISTER
            OUT     BASE+1      ;WRITE TRACK ZERO TO PARAMETER REGISTER
INIT16:     IN      BASE+0      ;READ STATUS REGISTER
            ANI     80H         ;CHECK COMMAND BUSY BIT SET (D7)
            JNZ     INIT16      ;WAIT FOR NOT BUSY
            IN      BASE+1      ;READ RESULT BYTE
            ORA     A           ;CHECK FOR ZERO RESULT BYTE
            JNZ     ERR2        ;IF NON-ZERO, GO TO ERROR ROUTINE
            JMP     FORM
            END
```

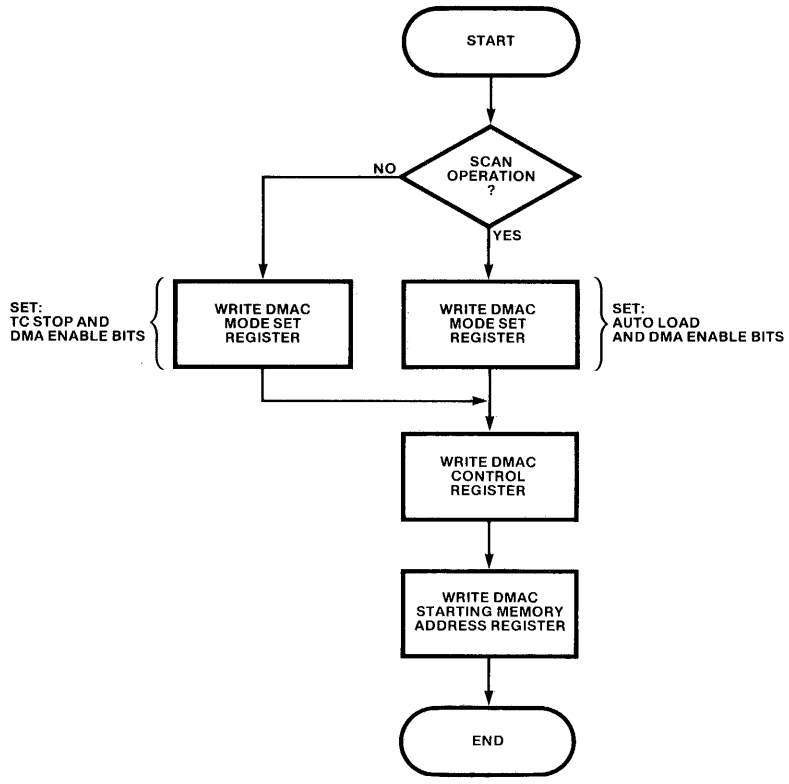


Figure 3-14. DMA Channel Initialization Flow Chart

Table 3-14. DMA Channel Initialization

```
;INITIALIZE DMAC FOR A WRITE OPERATION FROM MEMORY STARTING AT 4000
;TERMINAL COUNT SET FOR 128 BYTE SECTORS

BASE    EQU     00H
        PUBLIC  DMAI
        EXTRN   FORM

        CSEG
; ***************************************************************************
;                          WRITE DMAC MODE SET REGISTER
; ***************************************************************************
DMAI:   MVI     A,04H       ;MOVE MODE WORD INTO A REGISTER
        OUT     BASE+8      ;WRITE MODE SET REGISTER
; ***************************************************************************
;                          WRITE DMAC CONTROL REGISTER
; ***************************************************************************
        MVI     A,127D      ;CONTROL REGISTER LSB VALUE (128-1)
        OUT     BASE+5      ;WRITE CONTROL REGISTER LSB
        MVI     A,80H       ;MSB VALUE, WRITE DISK
        OUT     BASE+5      ;WRITE CONTROL REGISTER MSB
; ***************************************************************************
;                    WRITE DMAC STARTING MEMORY ADDRESS REGISTER
; ***************************************************************************
        MVI     A,00H       ;STARTING MEMORY ADDRESS LSB
        OUT     BASE+4      ;WRITE STARTING MEMORY ADDRESS LSB
        MVI     A,40H       ;MSB VALUE
        OUT     BASE+4      ;WRITE STARTING MEMORY ADDRESS MSB
        RET
        END
```

CHAPTER 4
THEORY OF OPERATION

4-1. INTRODUCTION

This chapter explains the circuit operation of the iSBC 204 Interface. 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 interface operation.

4-2. SCHEMATIC INTERPRETATION

The interface pc board schematic (drawing number 2002031) consists of five individual sheets which are labeled Sheet 1 of 5, Sheet 2 of 5, etc. These drawings (figure 5-2) and the pc board assembly drawing (figure 5-1), are located in Chapter 5.

Schematics are drawn to standard drafting conventions with input signals entering from the left and output signals exiting from the right. Input or output signals between individual sheets of the schematic include a location coordinate code immediately preceding (input signals) or following (output signals) the signal name. This code defines the schematic location for the origin or destination of the signal. The first digit of the code is the schematic sheet number, and the last two characters specify the zone (Z) as defined by the horizontal and vertical grid location coordinates appearing around the perimeter of the schematic. For example, the code "2ZB8" indicates that the origin or destination of the associated signal appears on sheet 2 of the schematic within the zone defined by schematic grid coordinates "B" and "8".

Schematic logic symbols follow the active state convention in the positioning of the inversion symbol. A gate with an inversion symbol at its output indicates that the output is active in its low state, and a gate without an inversion symbol at its output indicates that the output 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.

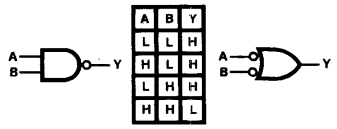


Figure 4-1. Logic Conventions

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 logic 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" which comprise the iSBC 204 Interface. Figure 4-7, located at the conclusion of this chapter, is the functional block diagram of the interface and shows the interrelationship of the various blocks.

4-4. CLOCK AND TIMER

The clock and timer circuit generates the asynchronous master timing signals MCLK (memory clock) and DCLK (data clock) and also generates the XACK/ (transfer acknowledge) signal on the Multibus interface in response to I/O port read and write commands addressed to the interface.

The clock and timer circuitry is shown on figure 5-2, sheet 2. The clock circuit is an Intel 8224 clock generator/timer circuit (A18) that uses a 16 MHz series resonant crystal as its reference source. The constant 16 MHz clock signal, at the OSC output, is routed to 4-bit synchronous counter A16 where it is divided down to the required clock frequencies. The OSC output is also routed to 8-bit parallel shift register A17 where it is used to derive the time-delayed control signals.

Depending on the type of drive supported (standard- or mini-sized), either the 16 MHz output signal from the clock generator/timer at wire-wrap post 38 or the 8 MHz output signal from the QA output of synchronous counter A16 at wire-wrap post 36 is connected to wire-wrap post 37 as the clock signal for the window generator circuit. Standard-sized drives require a 16 MHz clock signal (wire-wrap posts 38 and 37 connected), and mini-sized drives require an 8 MHz clock signal (wire-wrap posts 36 and 37 connected). The 2 MHz signal at the QC output of synchronous counter A16 is routed to the DMA controller as the MCLK signal. Again depending on the type of drive supported, either the 4 MHz output signal at the wire-wrap post 24 (standard-sized drives) or the 2 MHz output signal at wire-wrap post 22 (mini-sized drives) is connected to wire-wrap post 23 as the DCLK signal to the FDC circuit(s).

The control signal timer logic is based on 8-bit parallel shift register A17. This register is held in its clear state (all outputs logically low) until an I/O read or write command (IORC/ or IOWC/) is present on the Multibus interface. When either command signal is active, the clear input to the register is removed and, since the register's A and B inputs are permanently enabled, a logic high state is shifted into the parallel shift register on the positive transition of each 16 MHz clock pulse. When the I/O read or write command is addressed to the interface (I/O port address and switches agree), the LBDC (lower byte decode) and the UBDC/ (upper byte decode, used with 12-bit addressing) inputs will be active. These two inputs being active, allow the QB output from shift register A17-3 to control AND gate A30-6. On the second 16 MHz clock pulse, the register's QB output shifts to a logic high level which generates the BDSEL/ (board select) signal from A30-6. This signal performs the following functions:

1. Enables the tri-state buffer (A28) on the IORC/ and IOWC/ Multibus interface input lines to generate the DIOR/ (delayed I/O read) or DIOW/ (delayed I/O write) signal.
2. Enables the bidirectional bus drivers to accept or deposit the associated data byte on the Multibus interface.
3. Enables the XACK/ (transfer acknowledge) Multibus interface output line's tri-state buffer (A29).

On the eighth clock pulse to shift register A17, the QH output shifts to a logic high level. This output is inverted (A15) and buffered (A29) to become the XACK/ output signal to the CPU. Since the clock period is 0.0625 microseconds, the XACK/ signal is generated between 0.5 and 0.5625 microseconds following receipt of the I/O read or write command.

4-5. BASE ADDRESS DECODE

The base address decode circuit compares Multibus interface address bits 4 through 7 (4 through B for 12-bit addressing) with the interface's switch-selected base address. This comparison discriminates between I/O port instructions addressed to the interface and I/O port instructions addressed to other devices on the bus. When the address and switch values are equal (indicating an I/O port instruction addressed to the interface), the LBDC and UBDC/ (12-bit addressing) signals are generated. These signals enable both the command decode circuits and the output from the control signal timer.

The base address decode logic appears on sheet 1 of the schematic and consists of switch S2 and magnitude comparators A54 and A38. The ADEN/ (address and data enable) signal from the bus controller circuit is applied directly to the A=B input at pin 3 of the comparator A54. Since this input is inactive (logically high) during I/O port operations, the comparator is arranged to perform an "equals" comparison. Note that ADEN/ is active only during DMA transfer operations (when the interface has control of the bus) and consequently restricts the comparator from acknowledging a memory address containing a value that coincides with the base address switch settings. Multibus interface address lines ADR7/ through ADR4/ are applied directly to the A inputs of comparator A54, and address lines ADR8/ through ADRB/ are applied directly to the B inputs of comparator A38.. When the address bits of the I/O port read or write instruction coincide with the switch contact inputs, the LBDC and UBDC/ signals at the A=B outputs of the comparators shift to an active level.

4-6. COMMAND DECODE

The command decode circuits decode the four least significant address bits of the I/O port instruction (port address) to generate the individual chip select and control signals shown on the block diagram (figure 4-7).

The command decode logic appears on sheet 2 of the schematic and consists of two Intel 8205, 1-of-8 binary decoders (A52 and A53) and miscellaneous gating logic. The decoders are enabled by the LBDC and UBDC/ inputs from the base address decode circuits when an I/O port instruction is addressed to the interface. Table 4-1 defines the I/O port address decoding.

Note that it only is possible to write to port 9 (select FDC) and port F (reset interface) as their corresponding decoder outputs are gated with DIOW/ (delayed I/O write). Port 9 is unique in that it is contained

Table 4-1. I/O Port Address Decoding

ADDRESS BITS				PORT NUMBER (HEXADECIMAL)	DECODER OUTPUT ACTIVE	OUTPUT SIGNAL(S) GENERATED	FUNCTION
ADR3/	ADR2/	ADR1/	ADR0/				
H	H	H	H	0	A52, Pin 12	CSB/ and SELB or CSC/ and SELC	Select FDC
H	H	H	L	1	A52, Pin 12	CSB/ and SELB or CSC/ and SELC	Select FDC
H	H	L	H	2	A52, Pin 12	CSB/ and SELB or CSC/ and SELC	Select FDC
H	H	L	L	3	A52, Pin 12	CSB/ and SELB or CSC/ and SELC	Select FDC
H	L	H	H	4	A52, Pin 13	CSA/	Select DMAC
H	L	H	L	5	A52, Pin 13	CSA/	Select DMAC
H	L	L	H	6	A52, Pin 13	CSA/	Select DMAC
H	L	L	L	7	A52, Pin 13	CSA/	Select DMAC
L	H	H	H	8	A53, Pin 7	CSA/	Select DMAC
L	H	H	L	9	A53, Pin 9	None	*
L	H	L	H	A	A53, Pin 10	BASE/	Select Base Reg
L	L	L	L	F	A53, Pin 15	RST and RST/	Reset Interface

* I/O port 9 is used to enable CSB/ and SELB or CSC/ and SELC output signals for ports 0 through 3. CSB/ and SELB select first FDC circuit, CSC/ and SELC select second (optional) FDC circuit.

completely within the command decode circuit. Referring to sheet 2 of the schematic, the D0 output (which corresponds to bit 0 of the data byte associated with the select FDC command) from the bidirectional bus drivers is applied directly to the D input of flip-flop A26. The state of the D0 input is clocked into the flip-flop on the fifth 16 MHz clock pulse following receipt of the command. If the D0 input is inactive (if bit 0 of the associated data byte is clear), the CSB/ and SELB outputs will be enabled to select the first FDC circuit. Conversely, if the D0 input is active, the CSC/ and SELC outputs will be enabled to select the second (optional) FDC circuit. Clocking of the flip-flop occurs when the QE output of shift register A17 shifts to a logic high level (fifth input count pulse). The high level signal from the shift register (A17) is applied to NAND gate A27. The high level input causes the output to go high which in turn causes the output of AND gate A11-10 to go low. The low output from A11-10 is inverted by A13-4 to produce the required positive transition at the clock input to flip-flop A26.

4-7. BIDIRECTIONAL BUS DRIVERS

The bidirectional bus drivers accept or deposit a byte of data on the Multibus interface. Referring to sheet 1 of the schematic, the bus driver logic consists of two Intel 8226 tri-state bus driver circuits (A36 and A37). The bus drivers are enabled by either the BDSEL/ signal from the clock and timer circuit (active during I/O port operations) or by the DMAC/ (DMA control) signal from the bus controller (active during memory read/write operations). When both signals are inactive, the bus drivers are held in their high-impedance state. The direction of data flow through the bus drivers is determined by the state of the DIOR/ signal at the DIEN input to the drivers. Note that during I/O port operations (when the interface is functioning as a bus slave), DIOR/ is the delayed IORC/ command input from the CPU. During memory read/write operations (when the interface is functioning as a bus master), DIOR/ is generated by the DMA controller. When DIOR/ is active (low), data is transferred from the interface to the Multibus interface (memory write), and when DIOR/ is inactive (high) data is transferred from the Multibus interface to the interface (memory read).

4-8. BUS CONTROLLER

The bus controller which operates in conjunction with the DMA controller, both resolves bus contention among other master devices sharing the Multibus interface and generates the required control and timing signals responsible for meeting the Multibus interface address and data line set-up and hold times.

Bus controller operation is initiated by an FDC circuit through the DMA controller. The FDC requests a DMA transfer by activating its request output line (REQ1 or REQ2). Referring to sheet 3 of the schematic and figure 4-2, the DMA controller (A5), in response to the request at its DRQ2 input, activates its HRQ (hold request) output. HRQ, at the BCR1 (bus control request 1) input and OVRD input of A31, enables the bus controller's (A31) bus arbitration logic. OVRD (override) prevents the loss of the bus prior to transfer. On the succeeding cycle of the CPU's bus clock (BCLK/) signal, the bus controller activates its BREQ/ (bus request) output and inactivates its BPRO/ (bus priority out) output. BREQ/ is used to request bus access in a parallel priority environment, and BPRO/ is used in a serial priority environment to prevent lower-

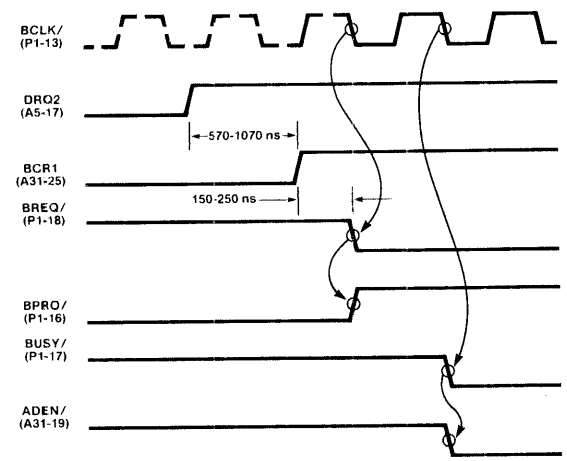


Figure 4-2. Bus Access Timing

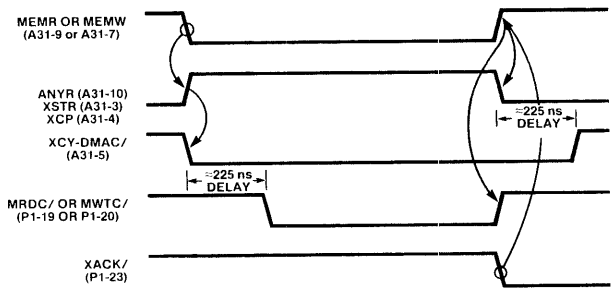


Figure 4-3. Set-Up and Hold Timing

priority bus masters from gaining bus access. If bus priority has been granted to the bus controller (noted by an active BPRN/ input and an inactive BUSY/ input), the bus controller activates BUSY/ on the next BCLK/ cycle to lock the bus controller onto the bus and to prevent any other bus master from accessing the bus. Coincident with activating BUSY/, the bus controller activates ADEN/ (address and data enable). This signal, in addition to disabling the base address decode comparator, enables the address register/drivers and, at the HLDA (hold acknowledge) input to the DMA controller (A5), indicates that the bus controller has gained control of the bus.

The data set-up and hold times are determined by the RC time constant established by C24 and R19 at the DLYADJ input to the bus controller A31. Typically, this RC delay is 225 nanoseconds. Circuit operation begins when the HLDA input to the DMA controller goes active. Referring to sheet 5 of the schematic and figure 4-3, depending on the DMA mode specified, either the MEMR (memory read) or MEMW (memory write) output is activated. At the bus controller, either one of these signals generates ANYR (any request). The ANYR output is routed to both the XSTR (transfer start request) and XCP (transfer complete) inputs. The positive-going, leading-edge of ANYR, at the XSTR input, initiates the transfer. Activating XSTR causes the XCY (transfer cycle) output to go active which, routed to sheet 1 as DMAC/, enables the bidirectional bus drivers. XSTR also triggers the bus controller's internal one-shot which, following the RC delay, activates the corresponding bus command (MRDC/ or MWTC/).

In response to the MRDC/ or MWTC/ command, the memory device activates its XACK/ output to acknowledge the transfer. XACK/, at the Ready input to the DMA controller, terminates the cycle and causes the MEMR or MEMW output to return to an inactive level. When the command goes inactive, the bus controller's ANYR and command (MRDC/ or MWTC/) outputs return to an inactive level. The negative-going transition of ANYR, at the XCP input, triggers the one-shot. When the one-shot times-out, the XCY (DMAC/) output returns to an inactive level to disable the bidirectional bus drivers.

4-9. DMA CONTROLLER

DMA controller operation with the bus controller is described in paragraph 4-8. As mentioned in Chapter 3, the reading and writing of individual registers within the DMA controller are accomplished through a set of I/O port commands. The actual register addressed is determined both by the four low-order address bits and by which of the I/O command inputs (IORC/ or IOWC/) is active. Referring to sheet 3 of the schematic, when the interface does not have access to the bus, the ADEN/ output from bus controller A31 is inactive, arranging bidirectional bus driver A51 in the input mode and routing the four, low-order address bits (ADR0/-ADR3/) of the I/O port command from the bus to the A0 through A3 inputs of the DMA controller. Table 4-2 defines the internal register selection.

When an I/O port command addressed to the interface is decoded, the BDSEL/ output is activated and enables the bidirectional bus drivers. When one of the DMA ports (ports 4-8) is addressed, the CSA/ (chip select A) output from the command decode logic (sheet 2) is activated and enables the 8257 DMA controller A5 (sheet 3). Depending on the state of the DIOR/ input at the bidirectional bus dirvers (sheet 1), the assoicated data byte is written into or read from the register addressed.

Table 4-2. DMA Controller Register Addressing

PORT ADDRESS	ADDRESS INPUT				I/O COMMAND ACTIVE	REGISTER ADDRESSED
	A3	A2	A1	A0		
4	0	1	0	0	IORC/ or IOWC/	DMAC Starting Memory Address
5	0	1	0	1	IORC/ or IOWC/	DMAC Control
6	0	1	1	0	IORC/ or IOWC/	DMAC Scan Pattern Address
7	0	1	1	1	IORC/ or IOWC/	DMAC Scan Control
8	1	0	0	0	IOWC/	DMAC Mode Set
8	1	0	0	0	IORC/	DMAC Status

As previously stated, a DMA transfer cycle is initiated when an FDC activates the DRQ2 input to the DMA controller. In response, the DMA controller activates the HRQ output to the bus controller. When the bus controller has accessed the bus, the HLDA input to the DMA controller is activated and the address register/drivers are enabled. In response to HLDA, the DMA controller enables the low-order memory address byte at the A0–A7 outputs, enables the high-order memory address byte at the D0–D7 outputs and activates the ADDSTB (address strobe) output (A5–8, sheet 3) to load the high-order memory address byte into the address register/drivers. The DMA controller then activates DACK2 (DMA acknowledge 2) which is gated to the selected FDC circuit. As shown in figure 4-4, the DMA controller next activates the appropriate read/write command outputs (MEMR and IOW or MEMW and IOR). MEMR or MEMW initiates the bus controller's data set-up cycle, the state of the IOR output at the bidirectional bus drivers selects the direction for the memory data transfer, and the IOR or IOW output at

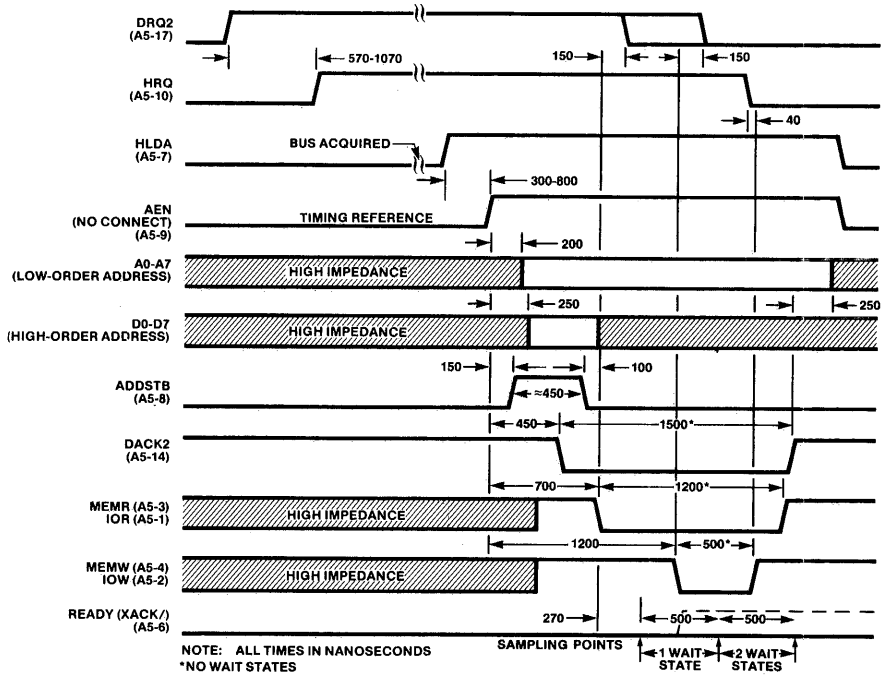


Figure 4-4. DMA Timing

the selected FDC circuit, with DACK2 active, causes the FDC to deactivate its request output (DRQ2 input at DMA controller goes inactive).

When the data byte has been read from or written to the memory location addressed, the memory device activates XACK/. At the Ready input to the DMA controller, this signal terminates the cycle. Note that depending on the acknowledge response time from the memory device, one or more wait states may be inserted and that each wait state extends the cycle by 500 nanoseconds.

4-10. ADDRESS REGISTER/ DRIVERS (16-BIT)

The address register/drivers accept the 16-bit memory address from the DMA controller and, when enabled, place the address on the Multibus interface as ADR0/ through ADRF/.

Referring to sheet 3 of the schematic, 8-bit input/ output port circuits A33 and A34 are not used with 16-bit addressing. The A0 through A3 outputs from the DMA controller are applied directly to the D inputs of 4-bit bidirectional bus driver A51. During DMA transfers, this circuit places the four, low-order address buts (ADR0/-ADR3/) on the Multibus interface. The A4 through A7 outputs from the DMA controller are applied to the B inputs of 4-bit binary full adder A49. Since input/output port circuit A34 is not used (its outputs are zero), the adder is transparent, and the A4 through A7 inputs are applied, intact, to output buffer line driver/receiver A50. The D0 through D7 outputs from the DMA controller are applied directly to octal type D latch/flip-flop A47. These inputs, which correspond to the high-order memory address byte, are enabled while ADDSTB is active and are latched into the circuit when ADDSTB returns to an inactive level. Again, since the outputs from the two input/output port circuits are zero, full adders A48 and A46 are transparent and the outputs from the latch/flip-flop are routed, intact, through the adders to their corresponding inputs at octal buffer line driver/ receivers A50 and A44.

ADEN/ from the bus controller block enables the line driver/receivers and the bidirectional bus driver to place the memory address on the Multibus interface.

4-11. ADDRESS REGISTER/ DRIVERS (20-BIT)

The base address register (A33-A34, sheet 3) is loaded with a 16-bit segment address value by two I/O write commands addressed to port A. When an I/O write comand to port A is decoded, BASE/ (2ZB1) from the command decode circuit (A53) goes low. BASE/ going low enables one leg of NAND gate A27-2 (3ZB7). On the second clock pulse to timer A17 (2ZA5), the IOWC/ (figure 4-5) signal conditions one leg of NAND gate A27-3 (2ZC5), the other leg is conditioned until the fifth clock pulse is received by the timer. The output of NAND gate A27-11 going low, enables NAND gate A27-3 (3ZB7). This output, at the S1 enable input to A34-1 (3ZD5), loads the associated (low-order segment address) data byte on the bus into address register A34. The other enable input to A34-13, on S2, is conditioned because the first/last flip-flop is in its clear (first) state following reset or the prior loading of a two byte segment address. On the fifth clock pulse, IOWB/ returns to an inactive level (high) which clocks the first/last flip-flop to the last position. When the second I/O write command to port A is decoded, the same steps take place with the high-order segment address being loaded into address register A33. On the fifth clock pulse to the timer the first/last flip-flop is set back to the first position.

The A0 through A3 outputs from DMA controller A5 (3ZD7) are applied directly to the D inputs of 4-bit bidirectional bus driver A51 (3ZD3) as the four low-order Multibus interface address bits (ADR0/-ADR3/). The A4 through A7 outputs are applied to the B inputs of 4-bit binary full adder A49 (3ZD4), but since the base address register contains a segment address value, the low-order four bits of the segment address are added to the high-order four bits of the low-order offset address byte to form the ADR4/ through ADR7/ Multibus interface address bits. The high-order offset address byte from the DMA controller is loaded into octal latch A47, and are combined with the high-order four bits of the low-order segment address byte and the low-order four bits of the high-order segment address byte within adders A48 and A46 to form the ADR8/ through ADRF/ Multibus interface address bits. The high-order four bits of the segment address, since adder A45 is transparent, are routed intact through the adder as the ADR10/ through ADR13/ Multibus interface address bits. Note that adders A49, A48, A46 and A45 are cascaded and that if the C4 (carry) output from a lower-ordered adder is active, one count is added to the combined segment-offset output value of the next, higher-ordered adder.

4-12. WINDOW GENERATOR

The window generator circuit uses the composite clock and data pulses originating from the drive and the separator clock signal from the clock and timer to generate the END/ (window) signal. This signal is, in turn, used by the FDC circuit with the composite clock and data input from the drive to determine

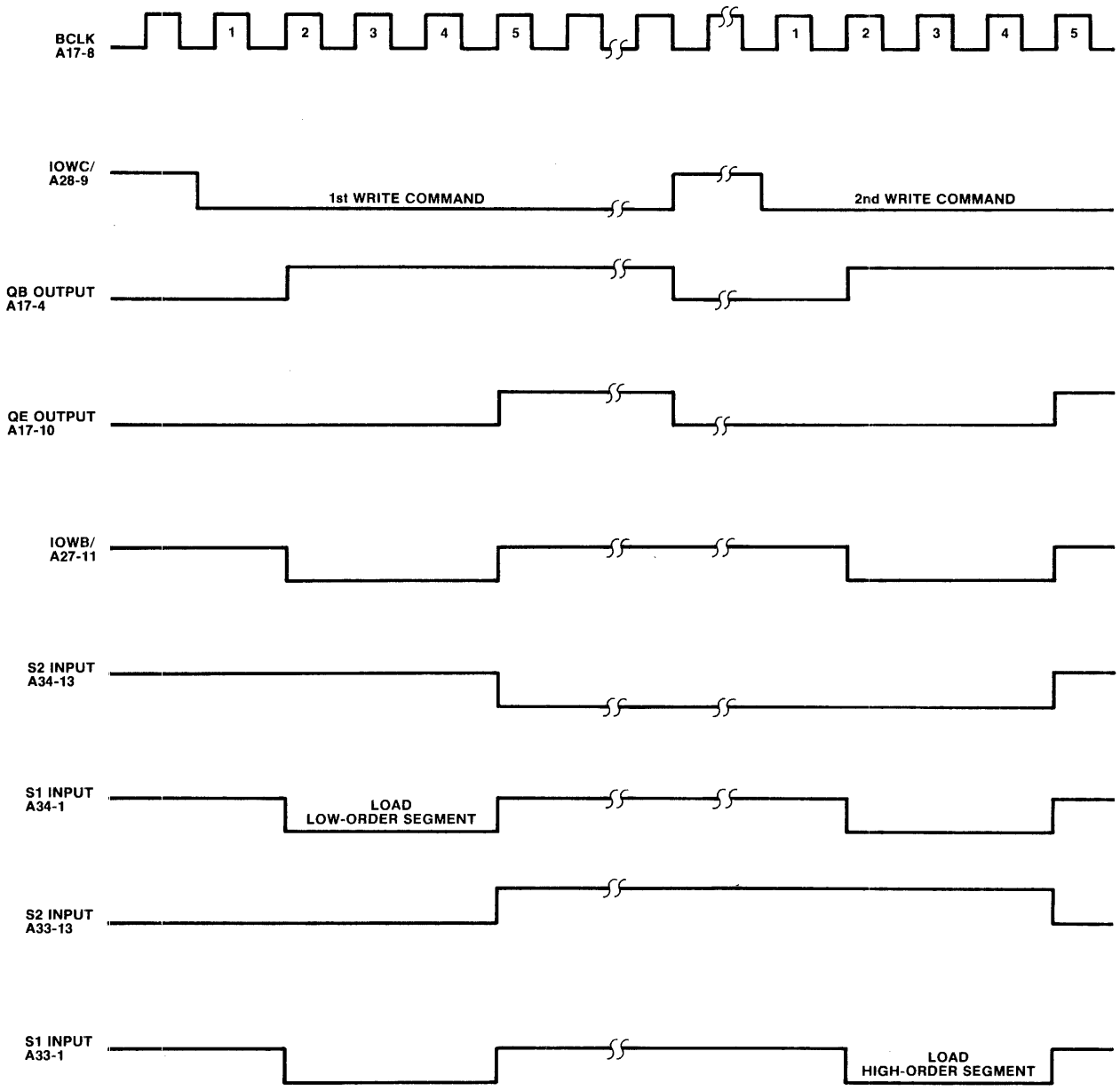


Figure 4-5. Low- and High-Order Segment Address Loading Timing

when a "zero" or "one" bit is encountered. The window generator is a digital, retriggerable one-shot circuit which times out only when the time interval between pulses is equal to a full bit cell. The FDC circuit examines the state of WND/ on the leading edge of each clock pulse. If the time interval from the previous pulse is equal to a full bit cell, the WND/ signal is inactive to indicate a "zero" bit. Conversely, if the interval is equal to half a bit cell, the WND/ signal is active to indicate a "one" bit.

Referring to sheet 2 of the schematic, the window generator consists principally of two D-type flip-flops (A10) and two synchronous 4-bit counters (A23 and A24). The composite clock and data input signal from a drive is inverted and routed to either A11-12 as DDAT1 (drives 0 and 1) or to A11-11 as DDAT2 (drives 2 and 3). The inverted output at A11-13 is coupled directly to the D input of the first flip-flop. The flip-flops and the two synchronous counters are clocked simultaneously by either the 16 MHz clock signal from clock generator/driver circuit A18 (standard-sized drives) or by the 8 MHz clock signal from the QA output of synchronous counter A16 (mini-sized drives). As shown in figure 4-6, the Q output from the first flip-flop is clocked to a logic high

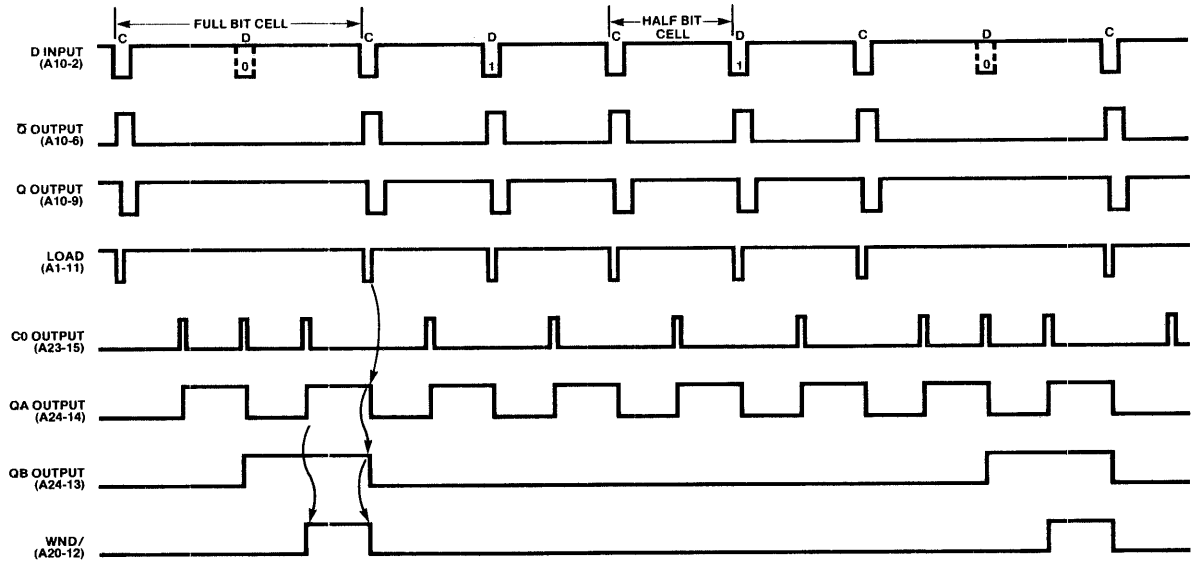


Figure 4-6. Data Separator Timing

level by each clock or data pulse. The Q output from the first flip-flop is coupled to the D input of the second flip-flop which, on the next clock pulse, produces a logic low level Q output from the second flip-flop. The Q output from the first flip-flop and the Q output from the second flip-flop are ANDed to produce the negative-going load pulse from A1-11. This pulse, which is generated on the leading edge of each clock or data pulse from the composite input signal, presets synchronous counter A23 to 0 and presets sysnchronous counter A24 to 12. Since the QA and QB outputs from synchronous counter A24 are preset to their inactive (low) state, a logic high level output is present at A25-3. This output, at the count enable inputs (EN P and EN T) to synchronous counter A23, enables the circuit's count function. On the count of 15, counter A23's CO output shifts to a logic high level for one clock cycle. At the count enable inputs to synchronous counter A24, the CO output, from A23 enables the counter to accumulate one count for every 16 cycles of the clock. On the first count, the QA output from A24 is active. When the interval between pulses is a half bit cell (data pulse present), a subsequent load pulse is generated to preset both counters before another count can be accumulated in A24, and the WND/ output at A20-12 remains active. When the interval between pulses is a full bit cell, two subsequent counts are accumulated in A24 before the next load pulse, and both the QA and QB outputs are permitted to go active. When active, the resultant logic low level output at A25-3 disables synchronous counter A23 and causes the WND/ output to shift to an inactive level until the next load pulse is generated to again preset the counters.

4-13. FDC CIRCUIT

The FDC circuit performs both the parallel-to-serial and serial-to-parallel conversion of data written to or read from the drive and all drive positioning and control functions. Since the operation of the two FDC circuits is identical, the following description is applicable to both FDC circuits. FDC circuit 1 appears on sheet 4 of the schematic and FDC circuit 2 appears on sheet 5.

The reading and writing of the individual registers within the FDC circuit is accomplished through a unique set of I/O port commands. The actual register addressed is determined both by the two low-order address bits (A0 and A1) of the I/O port command and by which of the I/O command inputs (IORC/ or IOWC/) is active. Table 4-3 defines the register selection.

When an I/O port command addressed to the interface is decoded, the BDSEL/ (2ZB1) output from the clock and timer circuit is activated and enables the bidirectional bus drivers. When one of the FDC I/O ports (ports 0, 1, 2) is addressed, the CSB/ (2ZB1) or CSC/ (2ZB1) output from the command decode circuit is activated to enable the corresponding FDC circuit. Depending on the state of the IORC/ command input at the bidirectional bus drivers, the

Table 4-3. FDC Register Addressing

PORT ADDRESS	ADDRESS INPUT		I/O COMMAND ACTIVE	REGISTER ADDRESSED
	A1	A0		
0	0	0	IOWC/	Command
0	0	0	IORC/	Status
1	0	1	IOWC/	Parameter
1	0	1	IORC/	Result
2	1	0	IOWC/	*Test

*Test register used with the Reset FDC I/O port command.

associated data byte is written into or read from the register addressed.

Following command and parameter entry, the FDC circuit enters the execution phase. Before a disk read or write operation is initiated, the FDC initiates a seek to the track addressed, monitors several of the drive's status lines, loads the head and, following the head load/head settling delay, begins reading the ID fields on the addressed track to locate the sector addressed. During the transfer, the FDC activates its DMA request output (REQ1 or REQ2) with each data byte to initiate a DMA cycle. Following the transfer operation, the FDC activates its interrupt request output (DINT1 or DINT2) which is available to the host processor through the interrupt matrix. The host processor then must examine the FDC's result register to determine the outcome of the operation.

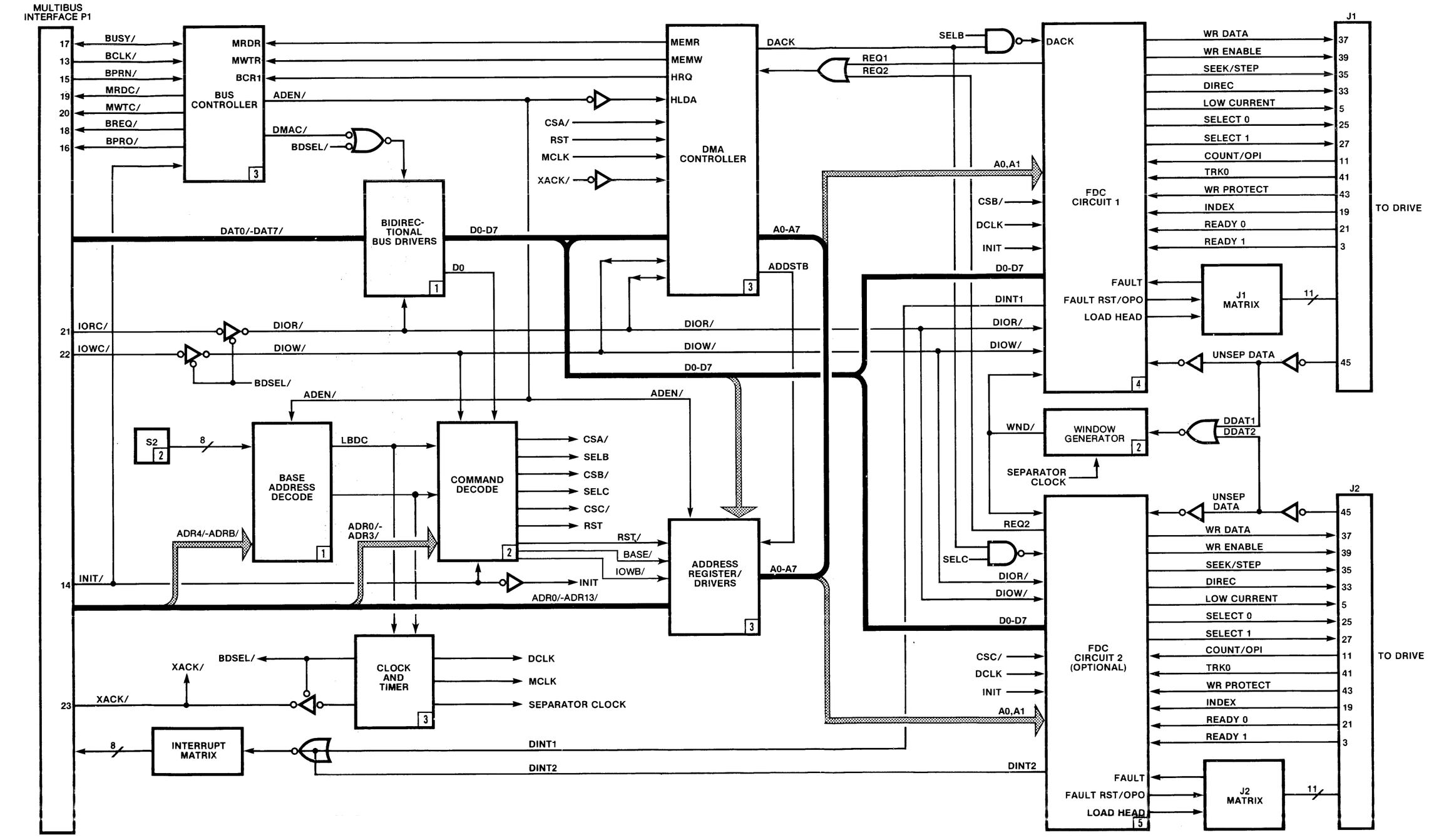


Figure 4-7. iSBC 204™ Functional Block Diagram

CHAPTER 5
SERVICE INFORMATION

5-1. INTRODUCTION

This chapter provides service and repair assistance instructions, service diagrams and the complete electronic parts list for the iSBC 204 Interface.

5-2. SERVICE DIAGRAMS

The iSBC 204 component location and schematic diagrams (figures 5-1 and 5-2, respectively) are included at the end of this chapter.

5-3. SERVICE AND REPAIR ASSISTANCE

United States customers can obtain service and repair assistance from Intel by contacting the MCSD Technical Support Center in Santa Clara, California at one of the following numbers:

Telephone:

From Alaska or Hawaii call—
(408) 987-8080

From locations within California call toll free—
(800) 672-3507

From all other U.S. locations call toll free—
(800) 538-8014

TWX: 910-338-0029 or 910-338-0255
TELEX: 34-6372

Always contact the MCSD Technical Support Center before returning a product to Intel for service or repair. You will be given a "Repair Authorization Number", shipping instructions, and other important information which will help Intel provide you with fast, efficient service. If the product is being returned because of damage sustained during shipment from Intel, or if the product is out of warranty, a purchase order is necessary in order for the MCSD Technical Support Center to initiate the repair.

In preparing the product for shipment to the MCSD Technical Support Center, use the original factory packaging material, if available. If the original packaging is not available, wrap the product in a cushioning material such as Air Cap TH-240 (or equivalent) manufactured by the Sealed Air Corporation, Hawthorne, N.J., and enclose in a heavy-duty corrugated shipping carton. Seal the carton securely, mark it "FRAGILE" to ensure careful handling, and ship it to the address specified by MCSD Technical Support Center personnel.

Customers outside of the United States should contact their sales source (Intel Sales Office or Authorized Intel Distributor) for directions on obtaining service or repair assistance.

5-4. REPLACEABLE COMPONENTS

This section contains the information necessary for the procurement of replacement components directly from commercial sources. Component manufacturers have been abbreviated in the parts list with either a two, three or four character code. Table 5-1 cross-references the manufacturer's code with the name and location of the prime commercial source. Table 5-2 is the list of replaceable components for the iSBC 204 Interface. Note that components that are available commercially are listed in the "MFR CODE" column as "COML" and that they are ordered by description (OBD). Every effort should be made to procure commercially-available components from a local distributor.

Table 5-1. Code of Manufacturers

MFR CODE	MANUFACTURER	LOCATION
CRY	Crystek Crystals Corp.	Fort Myers, FL
CTSB	CTS of Berne, Inc.	Berne, IN
CTSK	CTS Keene, Inc.	Paso Robles, CA
TI	Texas Instruments	Dallas, TX
COML	Any Commercial Source; Order By Description	

Table 5-2. Parts List

REFERENCE DESIGNATION	DESCRIPTION	MANUFACTURER PART NUMBER	MFR CODE	QUANTITY REQUIRED
A1,25,32	Integrated Circuit, 74LS00 Quad, 2-Input NAND Gate	SN74LS00N	TI	3
A2,3,7,8	Integrated Circuit, 7416 Hex Inverter Buffer/Driver	SN7416N	TI	4
A4,6*	Integrated Circuit, 8271 Floppy Disk Controller	Intel 8271	COML	1
A5	Integrated Circuit, 8257 DMA Controller	Intel 8257	COML	1
A9,39-43,55-61	Not Used			
A10	Integrated Circuit, 74S74 Dual Type D Flip-Flop	SN74S74N	TI	1
A11	Integrated Circuit, 74LS02 Quad, 2-Input NOR Gate	SN74LS02N	TI	1
A12-15	Integrated Circuit, 74LS04 Hex Interter	SN74LS04N	TI	4
A16	Integrated Circuit, 74LS193 Synchronous 4-Bit Counter	SN74LS193N	TI	1
A17	Integrated Circuit, 74LS164 8-Bit Shift Register	SN74LS164N	TI	1
A18	Integrated Circuit, 8224 Clock Generator/Driver	Intel 8224	COML	1
A19-22	Integrated Circuit, 7414 Hex Schmitt-Trigger Inverter	SN7414N	TI	4
A23,24	Integrated Circuit, 74LS161 Synchronous 4-Bit Counter	SN74LS161N	TI	2
A26	Integrated Circuit, 74LS74 Dual Type D Flip-Flop	SN74LS74N	TI	1
A27	Integrated Circuit, 74LS32 Quad, 2-Input OR Gate	SN74LS32N	TI	1
A28,29	Integrated Circuit, 74125 Quad, 3-State Bus Buffer	SN74125N	TI	2
A30	Integrated Circuit, 74LS10 Triple, 3-Input NAND Gate	SN74LS10N	TI	1
A31	Integrated Circuit, 8218 Bus Controller	Intel 8218	COML	1
A33,34	Integrated Circuit, 8212 8-Bit Input/Output Port	Intel 8212	COML	2
A35	Integrated Circuit, 74LS08 Quad, 2-Input AND Gate	SN74LS08N	TI	1
A36,37,51	Integrated Circuit, 8226 4-Bit Bidirectional Bus Driver	Intel 8226	COML	3

*Optional

Table 5-2. Parts List (Cont'd.)

REFERENCE DESIGNATION	DESCRIPTION	MANUFACTURER PART NUMBER	MFR CODE	QUANTITY REQUIRED
A38,54	Integrated Circuit, 74LS85 Magnitude Comparator	SN74LS85N	TI	2
A44,50	Integrated Circuit, 74LS240 Octal Buffer Line Driver/Receiver	SN74LS240N	TI	2
A45,46,48,49	Integrated Circuit, 74LS283 4-Bit Binary Full Adder	SN74LS283N	TI	4
A47	Integrated Circuit, 74LS373 Octal Type D Latch/Flip-Flop	SN74LS373N	TI	1
A52,53	Integrated Circuit, 8205 1-of-8 Binary Decoder	Intel 8205	COML	2
C1,2,6-13,15-22, 28,30-32,36,49	Capacitor, Ceramic Disc 0.01 μF, 25V, +80, −20%	OBD	COML	24
C3-5	Capacitor, Monolithic 0.1 μF, 50V, 20%	OBD	COML	3
C14	Capacitor, Dipped Mica 10 pF, 500V, 5%	OBD	COML	1
C24	Capacitor, Dipped Mica 390 pF, 500V, 5%	OBD	COML	1
C25,33	Capacitor, Electrolytic 22 μF, 35V, 10%	OBD	COML	2
R1,2,10,11	Resistor, Carbon Comp. 10k ohm, ¼W, 5%	OBD	COML	4
R3-8,15,17,18,20, 21	Resistor, Carbon Comp. 2.2k ohm, ¼W, 5%	OBD	COML	11
R9,16	Not Used			
R12,13,14	Resistor, Carbon Comp. 270 ohm, ¼W, 5%	OBD	COML	3
R19	Resistor, Carbon Comp. 470 ohm, ¼W, 5%	OBD	COML	1
RP1-4	Resistor Pack, 6-pin 150 ohm, 0.9W	750-61-R150Ω	CTSB	4
RP6,7	Resistor Pack, 6-pin 2.2k ohm, 0.9W	750-61-R2.2KΩ	CTSB	2
S2	Switch, SPST 8-Position	206-8	CTSK	1
Y1	Crystal, Series Resonant 16 MHz, HC18/U	CR60A/U	CRY	1

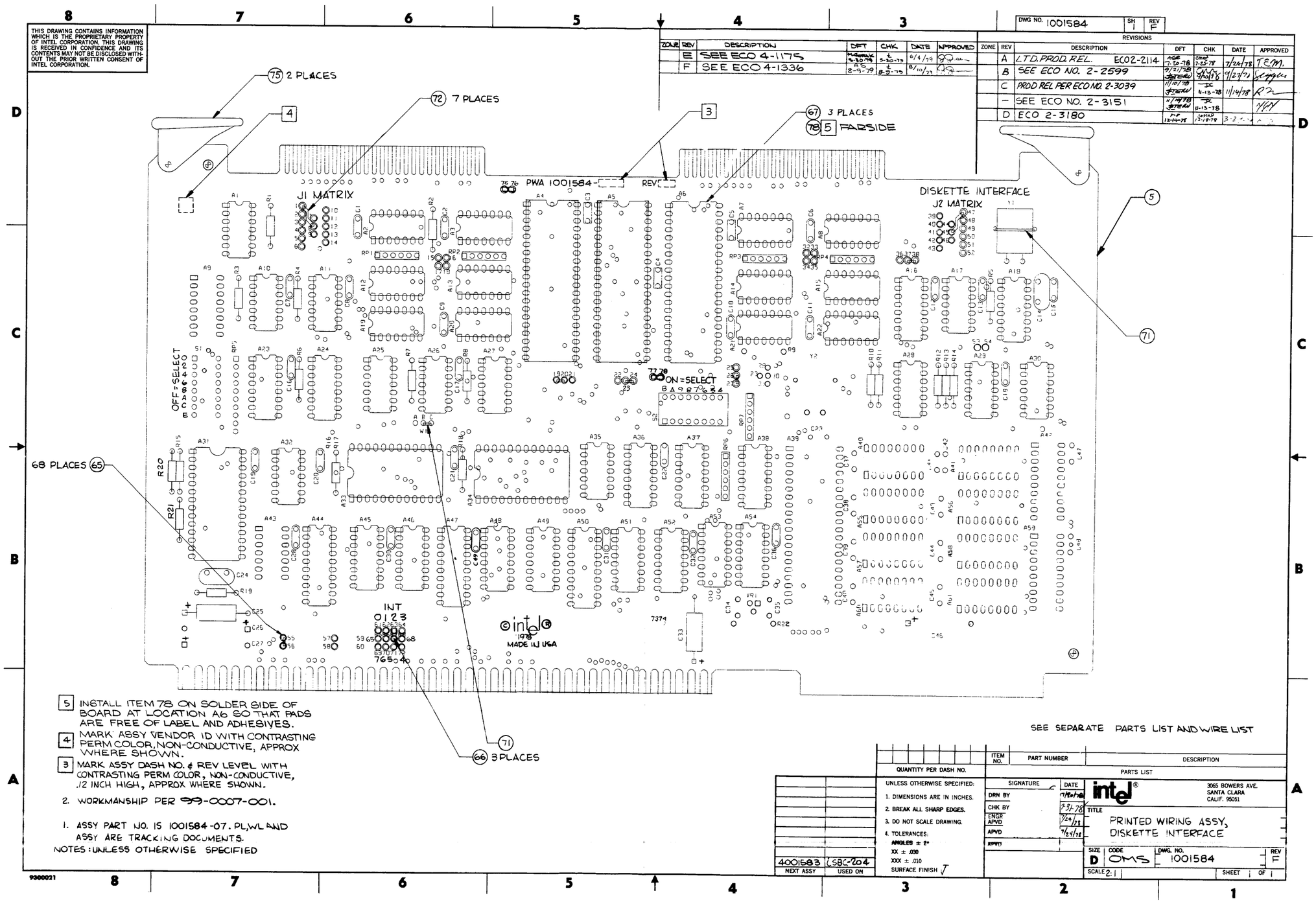
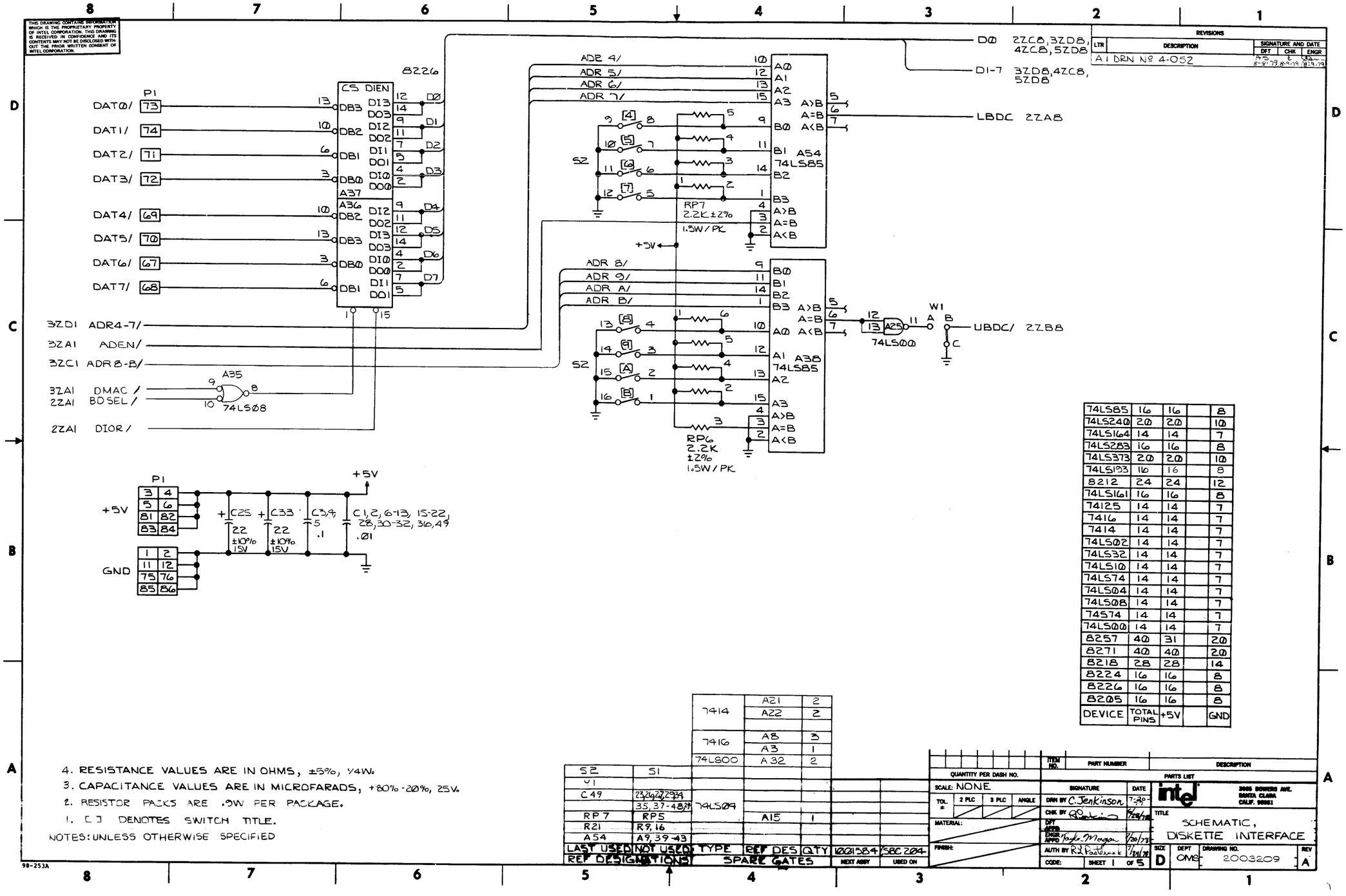


Figure 5-1. iSBC 204 Parts Location Diagram



74LS85	16	16		8
74LS240	20	20		10
74LS164	14	14		7
74LS283	16	16		8
74LS373	20	20		10
74LS193	16	16		8
8212	24	24		12
74LS161	16	16		8
74125	14	14		7
7416	14	14		7
7414	14	14		7
74LS02	14	14		7
74LS32	14	14		7
74LS10	14	14		7
74LS74	14	14		7
74LS04	14	14		7
74LS08	14	14		7
74S74	14	14		7
74LS00	14	14		7
8257	40	31		20
8271	40	40		20
8218	28	28		14
8224	16	16		8
8226	16	16		8
8205	16	16		8
DEVICE	TOTAL PINS	+5V		GND

		7414	A21	2			
			A22	2			
		7416	A8	3			
			A3	1			
		74LS00	A32	2			
S2	S1	74LS09					
Y1							
C49	23,26,27,29						
	35,37-48		A15	1			
RP7	RP5						
R21	R9,16						
A54	A9,39-43						
LAST USED	NOT USED	TYPE	REF DES	QTY	1001584	SBC 204	
REF DESIGNATIONS		SPARE GATES			NEXT ASSY	USED ON	

ITEM NO.	PART NUMBER	DESCRIPTION
PARTS LIST		

QUANTITY PER DASH NO.
SCALE: NONE
TOL ± 2 PLC 3 PLC ANGLE
MATERIAL:
FINISH:

	SIGNATURE	DATE
DRN BY	C. Jenkinson	7-20-73
CHK BY		7/24/73
DFT APPD		
ENGR APPD	Hayle Morgan	7/20/73
AUTH BY	R.J. Pasternak	7/24/73
CODE:	SHEET 1 OF 5	

intel 3065 BOWERS AVE. SANTA CLARA CALIF. 95051

TITLE: SCHEMATIC, DISKETTE INTERFACE

SIZE D | DEPT OMS | DRAWING NO. 2003209 | REV A

Figure 5-2. iSBC 204 Schematic Diagram (Sheet 1 of 5)

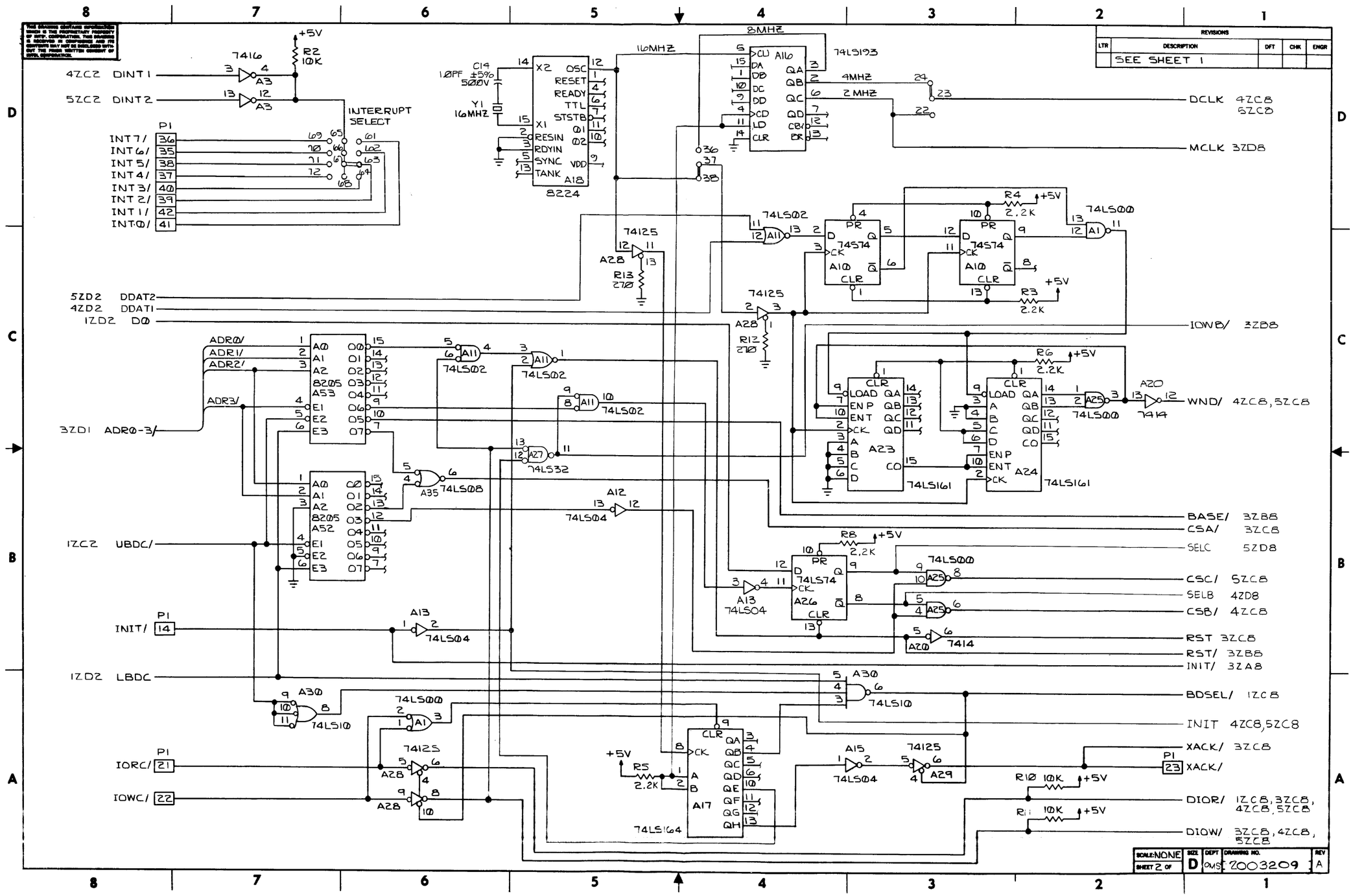


Figure 5-2. iSBC 204 Schematic Diagram (Sheet 2 of 5)

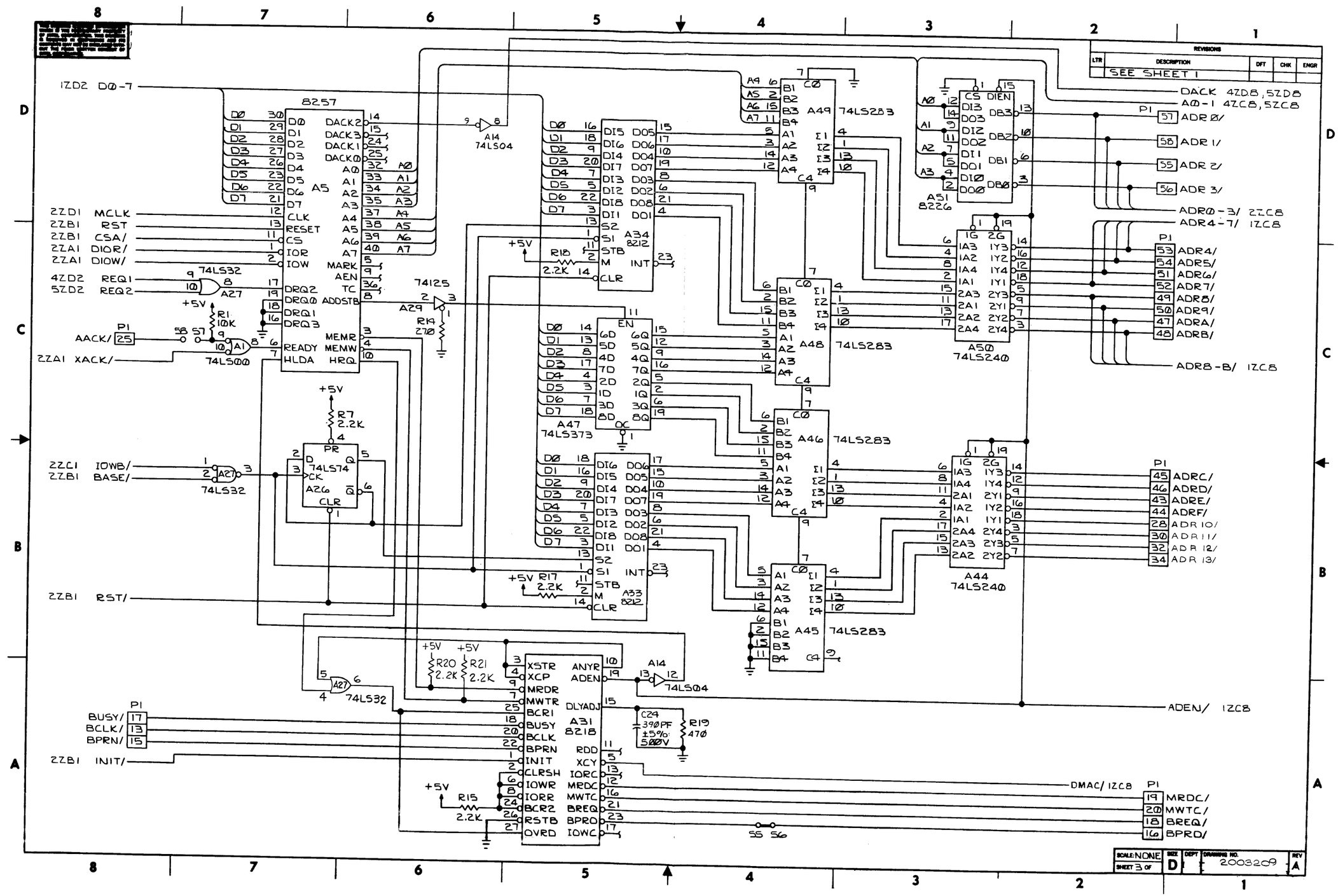


Figure 5-2. iSBC 204 Schematic Diagram (Sheet 3 of 5)

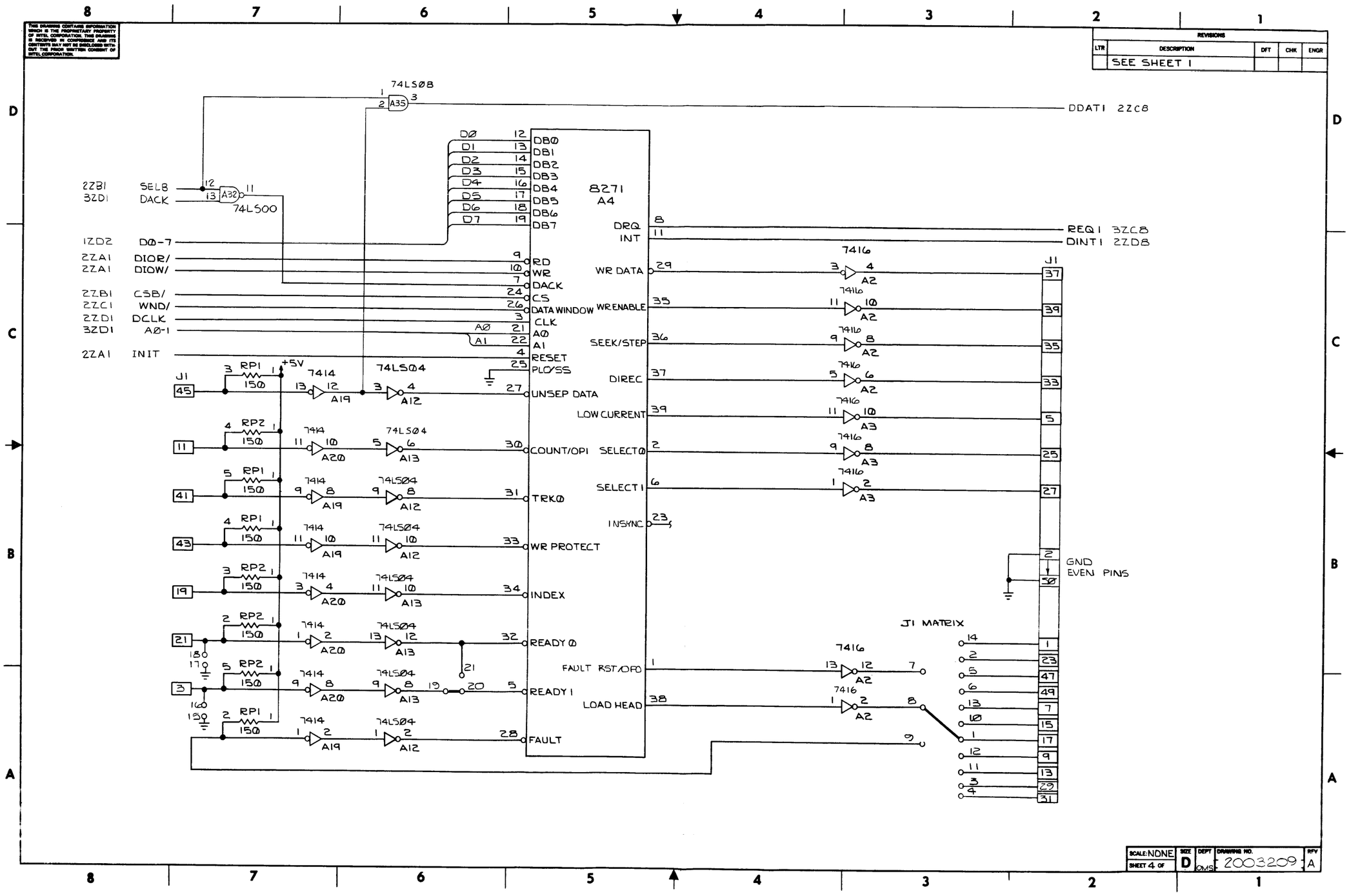


Figure 5-2. iSBC 204 Schematic Diagram (Sheet 4 of 5)

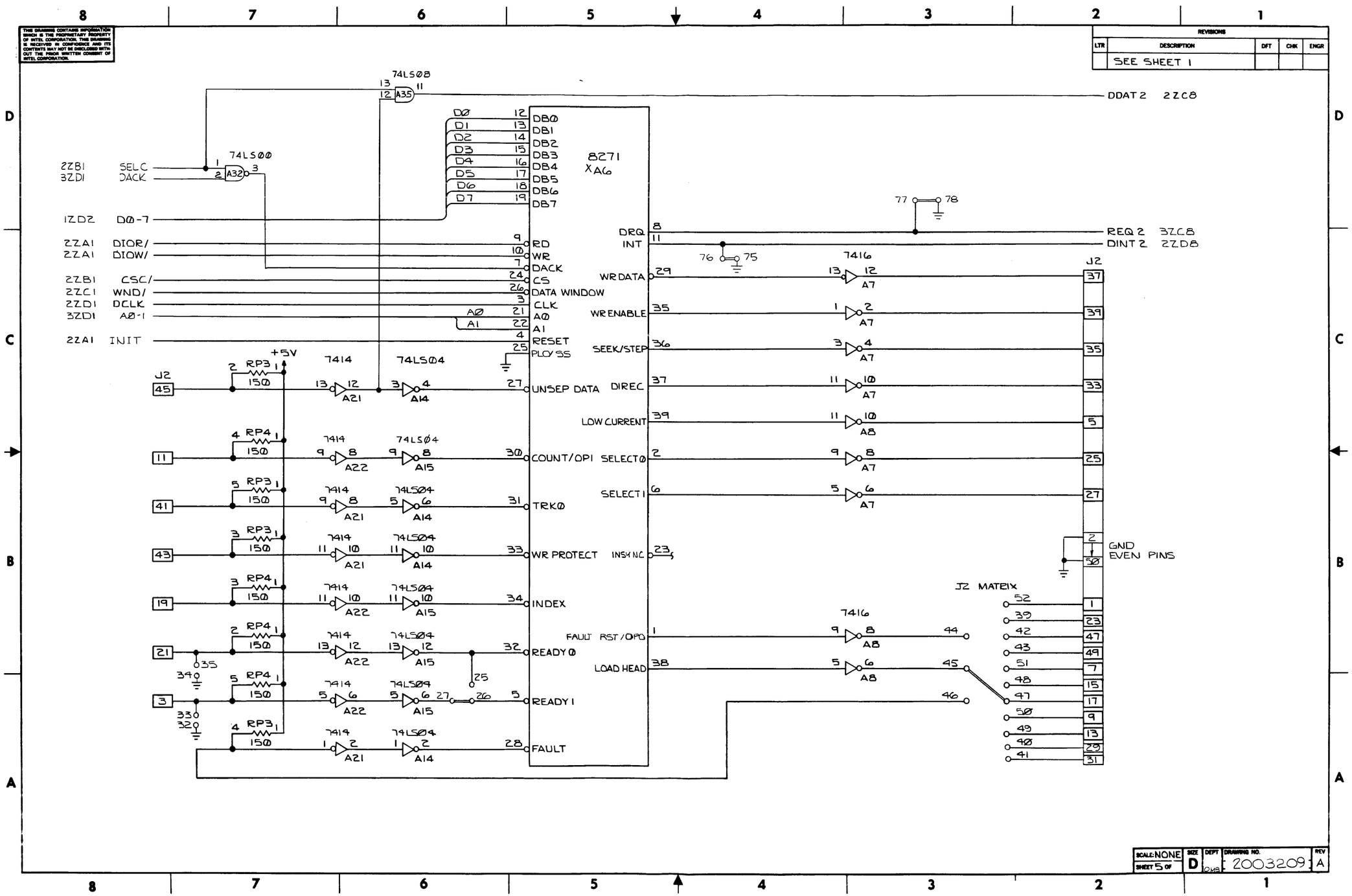


Figure 5-2. iSBC 204 Schematic Diagram (Sheet 5 of 5)

APPENDIX A
SCAN REGISTER INTERPRETATION

When scanning sectors of more than 128 bytes, the contents of both the scan block count register and the scan byte count register can be combined to determine the extent of the scan within a sector.

The scan block count register (special register 14_{16}) is initialized with a binary value which represents the sector length. This value is based on the expression $(2^n)-1$, where n is a number from 0 to 7. The initialization register values and the corresponding sector lengths for all values of n are shown.

n	INITIAL REGISTER VALUE		SECTOR LENGTH (IN BYTES)
	DECIMAL	HEXADECIMAL	
0	0	00	128
1	1	01	256
2	3	03	512
3	7	07	1024
4	15	0F	2048
5	31	1F	4096
6	63	3F	8192
7	127	7F	16384

The scan block count register is decremented with each block of 128 bytes transferred, and the scan byte count register, which is initialized to 128 at the beginning of each block, is decremented with each byte transferred. Consequently, the combined contents of the scan block count register and the scan byte count register (special register 13_{16}) represent the total number of unscanned bytes, plus one, remaining in the sector when the scan data pattern was located.

To simplify the 8080/8085 assembly language programming requirements to combine the contents of the two registers, the following calculation is used in the program example:

(Byte Count x 2 + Block Count x 256)/2

Program Example

```
                                BYTE REMAINDER SUBROUTINE ( BYTREM: )

                        KILLS   :       NOTHING

                        CALLS   :       RDSPRG ( USER'S READ SPECIAL REGISTER ROUTINE)

                        RETURNS :       END SECTOR OFFSET IN H&L REGISTERS

LOC  OBJ        SEQ             SOURCE STATEMENT        COMMENT

                68 BYTREM:
0000 F5         69              PUSH    PSW
0001 C5         70              PUSH    B
0002 0E13       71              MVI     C,13H           ; GET BYTE COUNT
0004 CD1600 C   72              CALL    RDSPRG
0007 07         73              RLC                     ; MULTIPLY BY 2
0008 6F         74              MOV     L,A             ; SAVE RESULT
0009 0E14       75              MVI     C,14H
000B CD1600 C   76              CALL    RDSPRG          ; GET BLOCK COUNT, RETURN WITH CARRY CLEAR
000E 1F         77              RAR                     ; DIVIDE BY 2  ( SAVE M.S. BIT IN CARRY )
000F 67         78              MOV     H,A             ; SAVE RESULT  ( MULTIPLY BY 256 )
0010 7D         79              MOV     A,L
0011 1F         80              RAR                     ; DIVIDE BY 2  ( APPEND TO L.S. BIT FROM M.S. BYTE )
0012 6F         81              MOV     L,A             ; CORRECT LSB
0013 C1         82              POP     B
0014 F1         83              POP     PSW
0015 C9         84              RET                     ; RETURNS NUMBER OF BYTES + 1 FROM END OF SECTOR
```

APPENDIX B
SUMMARY OF FDC AND I/O PORT COMMANDS

FDC COMMAND OP CODES		
Command	**Op Code (Hexadecimal)** Drive/ Surface 0 or 2	Drive/ Surface 1 or 3
Special Data Processing		
Specify	*35	*35
Format Track	63	A3
Read Sector ID	5B	9B
Special Drive		
Seek	69	A9
Read Drive Status	6C	AC
Read Special Register	7D	BD
Write Special Register	7A	BA
Routine Read/Write (Standard Format)		
Read Data	52	92
Read Data and Deleted Data	56	96
Write Data	4A	8A
Write Deleted Data	4E	8E
Verify Data and Deleted Data	5E	9E
(Special Format)		
Read Data	53	93
Read Data and Deleted Data	57	97
Write Data	4B	8B
Write Deleted Data	4F	8F
Verify Data and Deleted Data	5F	9F
Scan		
Scan Data	40	80
Scan Data and Deleted Data	44	84
*Does not involve diskette.		

FDC COMMAND FORMATS

Define Drive Characteristics

	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0	0	1	1	0	1	0	1	Write to Port 0
Parameter 0	0	0	0	0	1	1	0	1	Write to Port 1
Parameter 1	Step Rate*								Write to Port 1
Parameter 2	Head Settling Time*								Write to Port 1
Parameter 3	Index Count				Head Load Time*				Write to Port 1

*Mini-diskette parameters are doubled.

Load Bad Tracks

	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0	0	1	1	0	1	0	1	Write to Port 0
Parameter 0	0	0	0	1	1/0	0	0	0	Write to Port 1
Parameter 1	Bad Track No. 1								Write to Port 1
Parameter 2	Bad Track No. 2								Write to Port 1
Parameter 3	Current Track								Write to Port 1

D3=0, Define Bad Tracks Drive/Surface 0 or 2
D3=1, Define Bad Tracks Drive/Surface 1 or 3

Format Diskette

	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0/1	0/1	1	0	0	0	1	1	Write to Port 0
Parameter 0	Track Address								Write to Port 1
Parameter 1	Gap 3 Size Minus 6								Write to Port 1
Parameter 2	Record Length				No. of Sectors/Track				Write to Port 1
Parameter 3	Gap 5 Size Minus 6								Write to Port 1
Parameter 4	Gap 1 Size Minus 6								Write to Port 1

D7, D6=01, Select Drive/Surface 0 or 2
D7, D6=10, Select Drive/Surface 1 or 3

Read ID

	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0/1	0/1	0	1	1	0	1	1	Write to Port 0
	Track Address								Write to Port 1
	0	0	0	0	0	0	0	0	Write to Port 1
	Number of ID Fields								Write to Port 1

D7, D6=01, Select Drive/Surface 0 or 2
D7, D6=10, Select Drive/Surface 1 or 3

Seek

	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0/1	0/1	1	0	1	0	0	1	Write to Port 0
Parameter 0	Track Address 0-255*								Write to Port 1

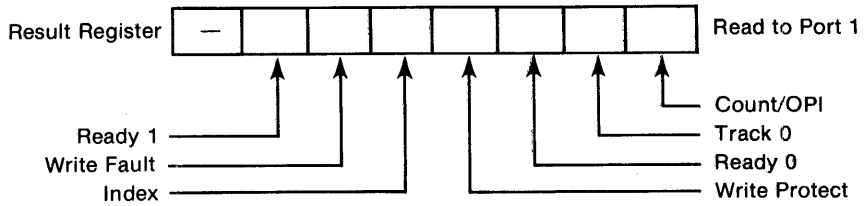
D7, D6=01, Select Drive/Surface 0 or 2
D7, D6=10, Select Drive/Surface 1 or 3

*Present diskette drives support up to 76 tracks.

Read Drive Status

	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0/1	0/1	1	0	1	1	0	0	Write to Port 0

D7, D6=01, Select Drive/Surface 0 or 2
D7, D6=10, Select Drive/Surface 1 or 3



Read Special Register*

	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0/1	0/1	1	1	1	1	0	1	Write to Port 0
Parameter	Register Address								Write to Port 1

D7, D6=01, Select Drive/Surface 0 or 2
D7, D6=10, Select Drive/Surface 1 or 3

* Data is returned in the Result Register.

Write Special Register

	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0/1	0/1	1	1	1	0	1	0	Write to Port 0
Parameter 0	Register Address								Write to Port 1
Parameter 1	Data								Write to Port 1

D7, D6=01, Select Drive/Surface 0 or 2
D7, D6=10, Select Drive/Surface 1 or 3

Routine Read Data

	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0/1	0/1	0	1	0	0	1	0	Write to Port 0
Parameter 0	Track Address								Write to Port 1
Parameter 1	Sector Address								Write to Port 1

D7, D6=01, Select Drive/Surface 0 or 2
D7, D6=10, Select Drive/Surface 1 or 3

Routine Read Data and Deleted Data

	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0/1	0/1	0	1	0	1	1	0	Write to Port 0
Parameter 0	Track Address								Write to Port 1
Parameter 1	Sector Address								Write to Port 1

D7, D6=01, Select Drive/Surface 0 or 2
D7, D6=10, Select Drive/Surface 1 or 3

Routine Write Data

	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0/1	0/1	0	0	1	0	1	0	Write to Port 0
Parameter 0	Track Address								Write to Port 1
Parameter 1	Sector Address								Write to Port 1

D7, D6=01, Select Drive/Surface 0 or 2
D7, D6=10, Select Drive/Surface 1 or 3

Routine Write Deleted Data

	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0/1	0/1	0	0	1	1	1	0	Write to Port 0
Parameter 0	Track Address								Write to Port 1
Parameter 1	Sector Address								Write to Port 1

D7, D6=01, Select Drive/Surface 0 or 2
D7, D6=10, Select Drive/Surface 1 or 3

Routine Verify Data and Deleted Data

	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0/1	0/1	0	1	1	1	1	0	Write to Port 0
Parameter 0	Track Address								Write to Port 1
Parameter 1	Sector Address								Write to Port 1

D7, D6=01, Select Drive/Surface 0 or 2
D7, D6=10, Select Drive/Surface 1 or 3

Special Read Data

	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0/1	0/1	0	1	0	0	1	1	Write to Port 0
Parameter 0	Track Address								Write to Port 1
Parameter 1	Sector Address								Write to Port 1
Parameter 2	Length				No. of Sectors				Write to Port 1

D7, D6=01, Select Drive/Surface 0 or 2
D7, D6=10, Select Drive/Surface 1 or 3

Special Read Data and Deleted Data

	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0/1	0/1	0	1	0	1	1	1	Write to Port 0
Parameter 0	Track Address								Write to Port 1
Parameter 1	Sector Address								Write to Port 1
Parameter 2	Length				No. of Sectors				

D7, D6=01, Select Drive/Surface 0 or 2
D7, D6=10, Select Drive/Surface 1 or 3

Special Write Data

	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0/1	0/1	0	0	1	0	1	1	Write to Port 0
Parameter 0	Track Address								Write to Port 1
Parameter 1	Sector Address								Write to Port 1
Parameter 2	Length				No. of Sectors				Write to Port 1

D7, D6=01, Select Drive/Surface 0 or 2
D7, D6=10, Select Drive/Surface 1 or 3

Special Write Deleted Data

	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0/1	0/1	0	0	1	1	1	1	Write to Port 0
Parameter 0	Track Address								Write to Port 1
Parameter 1	Sector Address								Write to Port 1
Parameter 2	Length				No. of Sectors				Write to Port 1

D7, D6=01, Select Drive/Surface 0 or 2
D7, D6=10, Select Drive/Surface 1 or 3

Special Verify Data and Deleted Data

	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0/1	0/1	0	1	1	1	1	1	Write to Port 0
Parameter 0	Track Address								Write to Port 1
Parameter 1	Sector Address								Write to Port 1
Parameter 2	Length				No. of Sectors				Write to Port 1

D7, D6=01, Select Drive/Surface 0 or 2
D7, D6=10, Select Drive/Surface 1 or 3

Scan Data

	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0/1	0/1	0	0	0	0	0	0	Write to Port 0
Parameter 0	Track Address								Write to Port 1
Parameter 1	Sector Address								Write to Port 1
Parameter 2	Length				No. of Sectors				Write to Port 1
Parameter 3	Scan Type				Step Size				Write to Port 1
Parameter 4	Field Length (Key)								Write to Port 1

D7, D6=01, Select Drive/Surface 0 or 2
D7, D6=10, Select Drive/Surface 1 or 3

Scan Data and Deleted Data

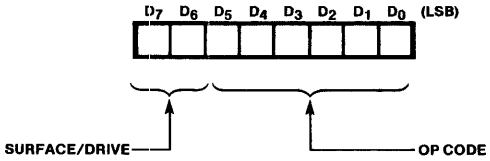
	D7	D6	D5	D4	D3	D2	D1	D0	
Command	0/1	0/1	0	0	0	1	0	0	Write to Port 0
Parameter 0	Track Address								Write to Port 1
Parameter 1	Sector Address								Write to Port 1
Parameter 2	Length				No. of Sectors				Write to Port 1
Parameter 3	Scan Type				Step Size				Write to Port 1
Parameter 4	Field Length (Key)								Write to Port 1

D7, D6=01, Select Drive/Surface 0 or 2
D7, D6=10, Select Drive/Surface 1 or 3

I/O PORT COMMANDS

Write FDC Command Register

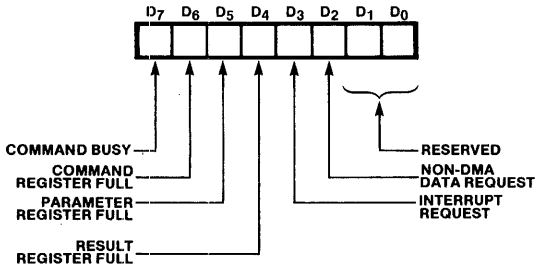
I/O Write to I/O Base Address+0



D7, D6=01, Select Drive/Surface 0 or 2
D7, D6=10, Select Drive/Surface 1 or 3

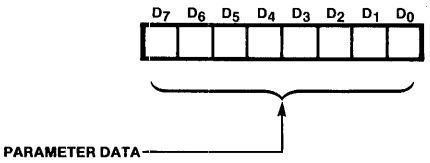
Read FDC Status Register

I/O Read to I/O Base Address+0



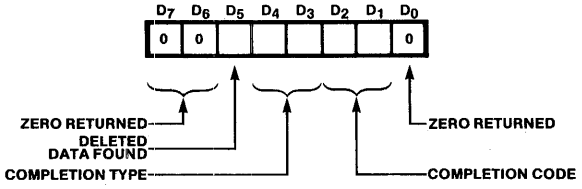
Write FDC Parameter Register

I/O Write to I/O Base Address+1



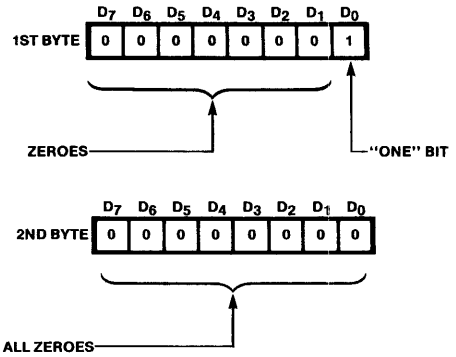
Read FDC Result Register (Delayed Result)

I/O Read to I/O Base Address+1



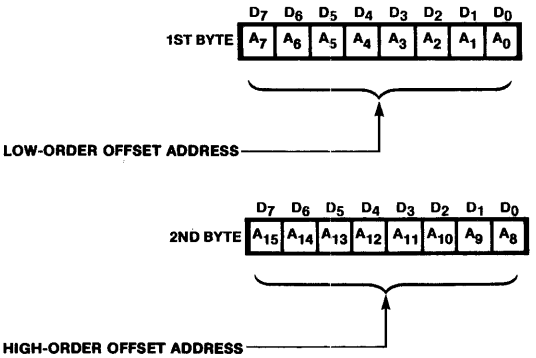
Reset FDC

Two I/O Writes to I/O Base Address+2



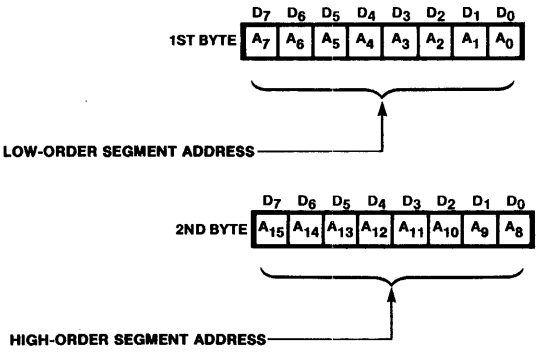
Read or Write DMAC Starting Memory Address Register (Offset)

Two I/O Reads or Writes to I/O Base Address+4



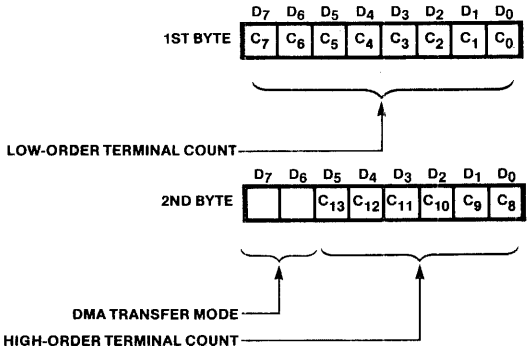
Write DMAC Starting Memory Address Register (Segment)

Two I/O Writes to I/O Base Address+A



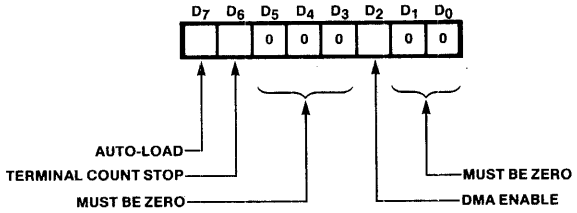
Read or Write DMAC Control Register

Two I/O Reads or Writes to I/O Base Address+5



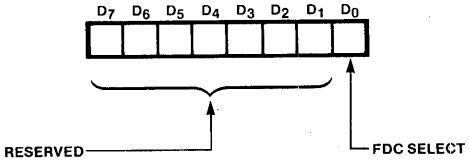
Write DMAC Mode Set Register

I/O Write to I/O Base Address+8



Select FDC Command

I/O Write to I/O Base Address+9



Reset Interface

I/O Write to I/O Base Address+F
(No data exchange)

REQUEST FOR READER'S COMMENTS

Intel Corporation attempts to provide documents that meet the needs of all Intel product users. This form lets you participate directly in the documentation process.

Please restrict your comments to the usability, accuracy, readability, organization, and completeness of this document.

1. Please specify by page any errors you found in this manual.

2. Does the document cover the information you expected or required? Please make suggestions for improvement.

3. Is this the right type of document for your needs? Is it at the right level? What other types of documents are needed?

4. Did you have any difficulty understanding descriptions or wording? Where?

5. Please rate this document on a scale of 1 to 10 with 10 being the best rating. _____

NAME _____ DATE _____

TITLE _____

COMPANY NAME/DEPARTMENT _____

ADDRESS _____

CITY _____ STATE _____ ZIP CODE _____

Please check here if you require a written reply. ☐

WE'D LIKE YOUR COMMENTS . . .

This document is one of a series describing Intel products. Your comments on the back of this form will help us produce better manuals. Each reply will be carefully reviewed by the responsible person. All comments and suggestions become the property of Intel Corporation.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1040 SANTA CLARA, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Intel Corporation
Attn: Technical Publications
3065 Bowers Avenue
Santa Clara, CA 95051

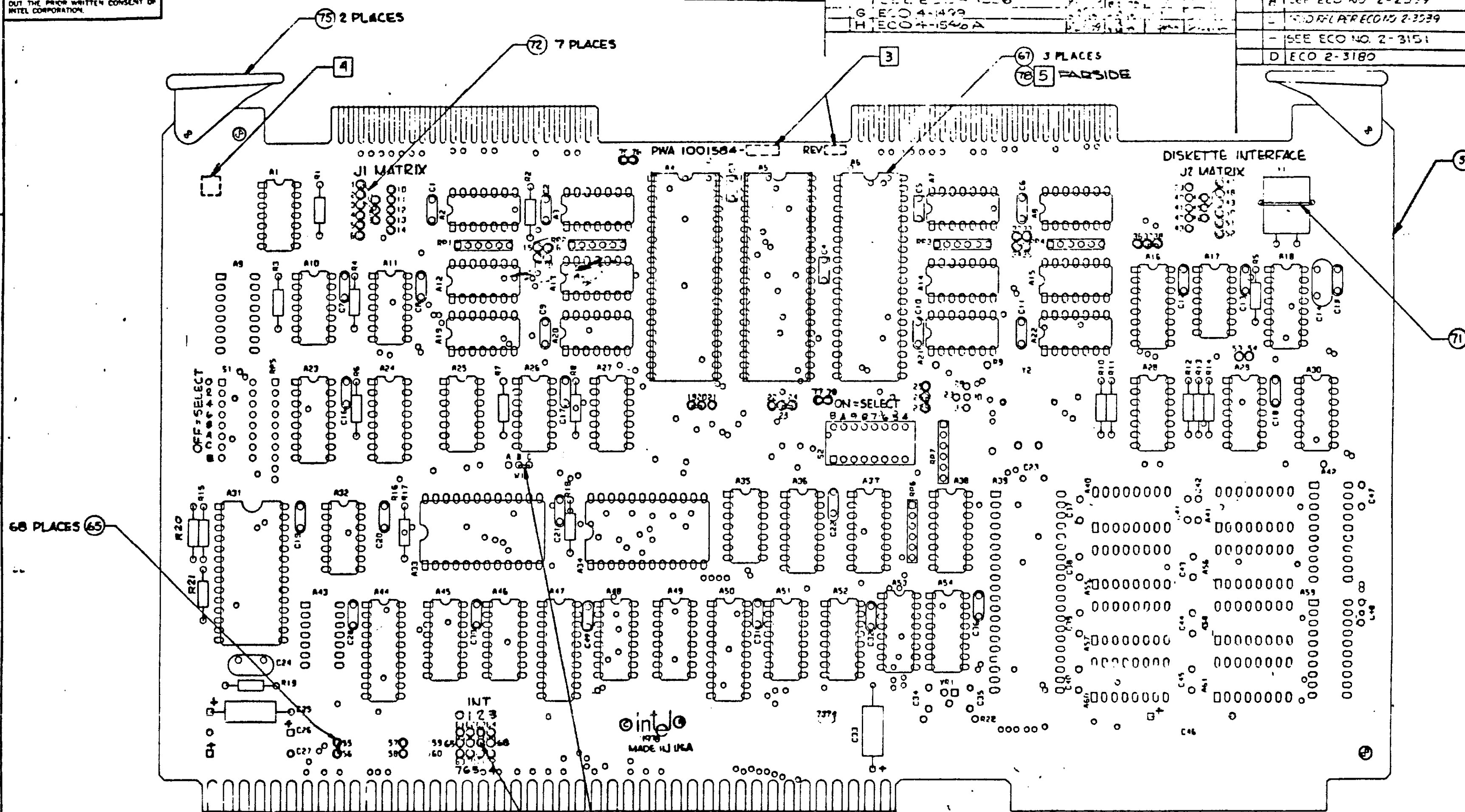
THIS DRAWING CONTAINS INFORMATION WHICH IS THE PROPRIETARY PROPERTY OF INTEL CORPORATION. THIS DRAWING IS RECEIVED IN CONFIDENCE AND ITS CONTENTS MAY NOT BE DISCLOSED WITHOUT THE PRIOR WRITTEN CONSENT OF INTEL CORPORATION.

J	ECO 40-1679
K	ECO 40-1844
L	ECO 40-2030

ZONE	REV	DESCRIPTION	DFT	CHK	DATE	APPROVED
	E	SEE ECO 4-1175				
	F	SEE ECO 4-1226				
	G	ECO 4-1429				
	H	ECO 4-1546A				

REVISIONS

ZONE	REV	DESCRIPTION	DFT	CHK	DATE	APPROVED
	A	ECO 2-2114				
	A	SEE ECO NO 2-2599				
		REDRAWN PER ECO NO 2-3539				
	–	SEE ECO NO. 2-3151				
	D	ECO 2-3180				



5 INSTALL ITEM 78 ON SOLDER SIDE OF BOARD AT LOCATION A6 SO THAT PADS ARE FREE OF LABEL AND ADHESIVES.

4 MARK ASSY VENDOR ID WITH CONTRASTING PERM COLOR, NON-CONDUCTIVE, APPROX WHERE SHOWN.

3 MARK ASSY DASH NO. & REV LEVEL WITH CONTRASTING PERM COLOR, NON-CONDUCTIVE, .12 INCH HIGH, APPROX WHERE SHOWN.

2. WORKMANSHIP PER 99-0007-001.

1. ASSEMBLY PART NO. IS 1001584-08, PL, WL AND ASSEMBLY ARE TRACKING DOCUMENTS.

NOTES: UNLESS OTHERWISE SPECIFIED

SEE SEPARATE PARTS LIST AND WIRE LIST

ITEM NO.	PART NUMBER	DESCRIPTION

QUANTITY PER DASH NO. — PARTS LIST

NEXT ASSY	USED ON
400583	1001-204

UNLESS OTHERWISE SPECIFIED
1. DIMENSIONS ARE IN INCHES
2. BREAK ALL SHARP EDGES
3. DO NOT SCALE DRAWING
4. TOLERANCES
ANGLES ± 2°
.XX ± .030
.XXX ± .010
SURFACE FINISH

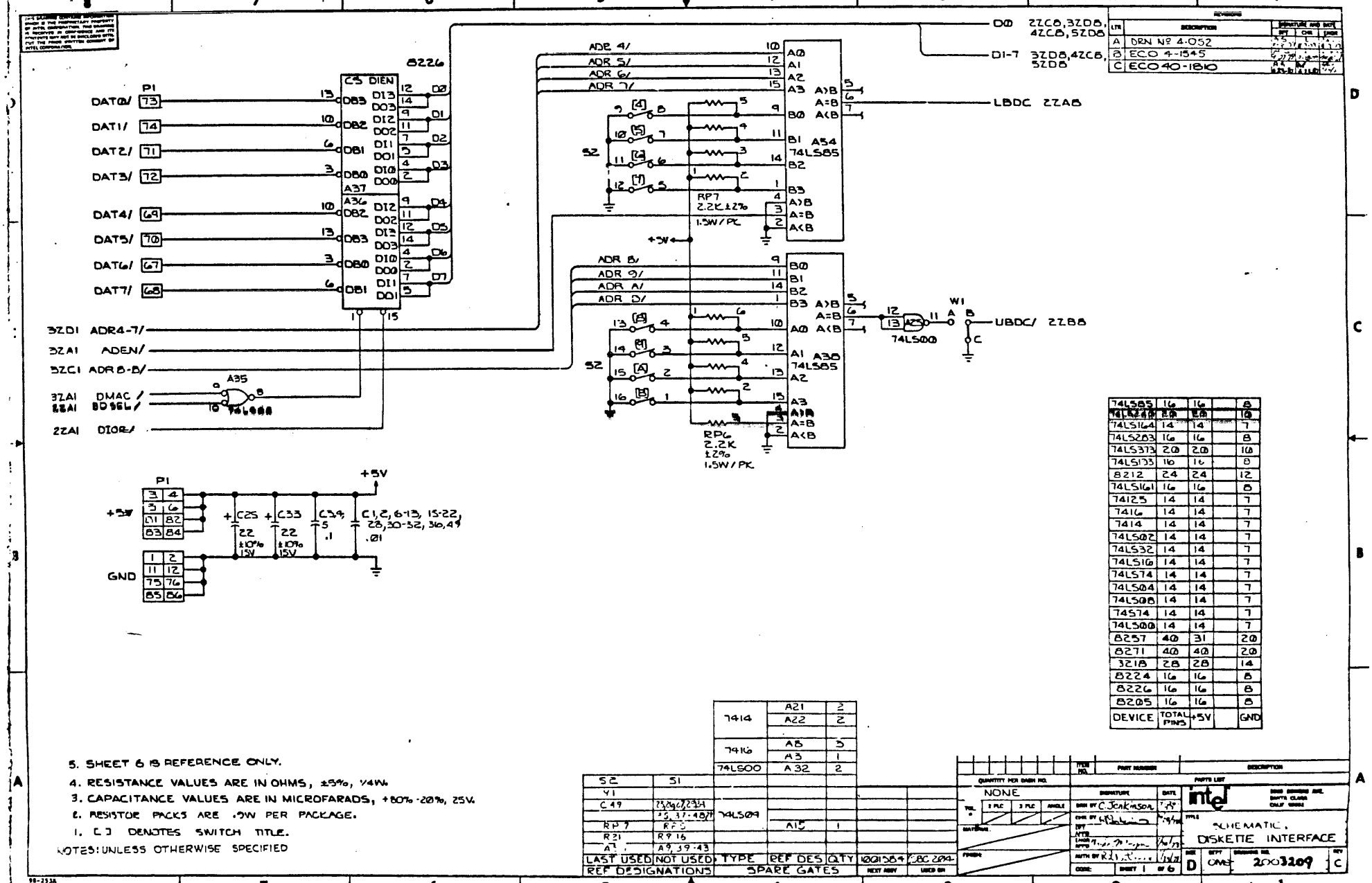
SIGNATURE	DATE
DRN BY	
CHK BY	
ENGR APVD	
APVD	
APVD	

intel 3065 BOWERS AVE. SANTA CLARA CALIF. 95051

TITLE: PRINTED WIRING ASSY, DISKETTE INTERFACE

SIZE	CODE	DWG NO	REV
D	CMS	1001584	L

SCALE 2:1 SHEET 1 OF 1



74LS85	16	16		8
74LS240	20	20		10
74LS164	14	14		7
74LS283	16	16		8
74LS373	20	20		10
74LS133	16	16		8
8212	24	24		12
74LS161	16	16		8
74125	14	14		7
7416	14	14		7
7414	14	14		7
74LS02	14	14		7
74LS32	14	14		7
74LS10	14	14		7
74LS74	14	14		7
74LS04	14	14		7
74LS08	14	14		7
7474	14	14		7
74LS00	14	14		7
8257	40	31		20
8271	40	40		20
3218	28	28		14
8224	16	16		8
8226	16	16		8
8205	16	16		8
DEVICE	TOTAL PINS	+5V		GND

7414	A21	2		
	A22	2		
7416	A8	5		
	A3	1		
74LS00	A32	2		
74LS09	A15	1		

S2	S1					
Y1						
C49	75pf/22pf					
RP7	RP5	74LS09	A15	1		
R21	R9 16					
A1	A9, 39-43					
LAST USED	NOT USED	TYPE	REF DES	QTY	1001364	28C 204
REF DESIGNATIONS		SPARE GATES			NEXT ASSY	USED ON

5. SHEET 6 IS REFERENCE ONLY.
4. RESISTANCE VALUES ARE IN OHMS, ±5%, ¼W.
3. CAPACITANCE VALUES ARE IN MICROFARADS, +80% -20%, 25V.
2. RESISTOR PACKS ARE .9W PER PACKAGE.
1. [ ] DENOTES SWITCH TITLE.

NOTES: UNLESS OTHERWISE SPECIFIED

LTR	DESCRIPTION	SIGNATURE AND DATE
A	DRN № 4-052	
B	ECO 4-1545	
C	ECO 40-1810	

QUANTITY PER DASH NO.: NONE

ITEM NO.	PART NUMBER	DESCRIPTION

PARTS LIST

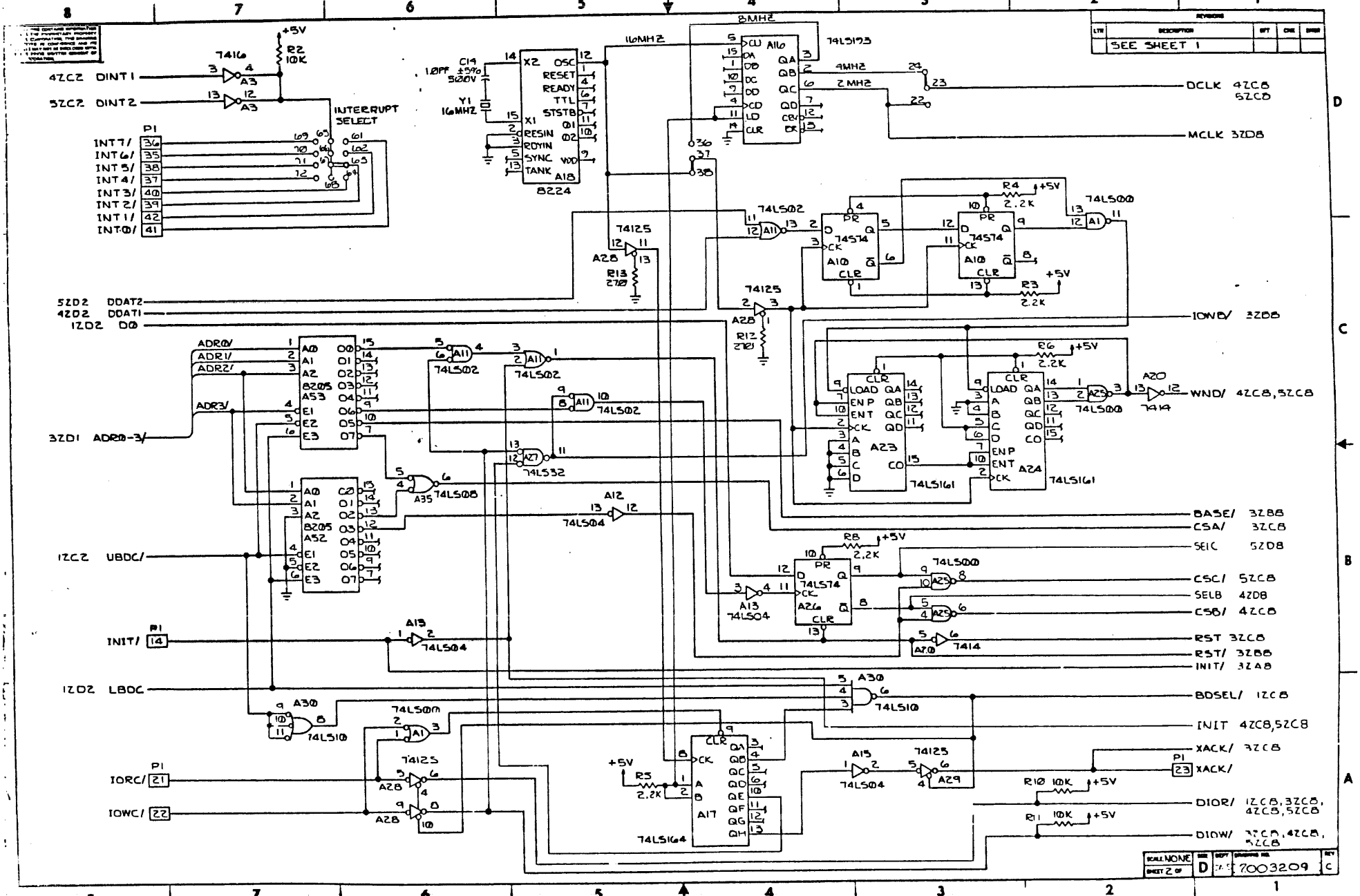
SIGNATURE	DATE
DRN BY C. Jenkinson	7/29
CHK BY	7/29/80
APP	
MFG	
ENGR APPR	7/31/80
AUTH BY R.L.	7/31/80
CODE:	SHEET 1 OF 6

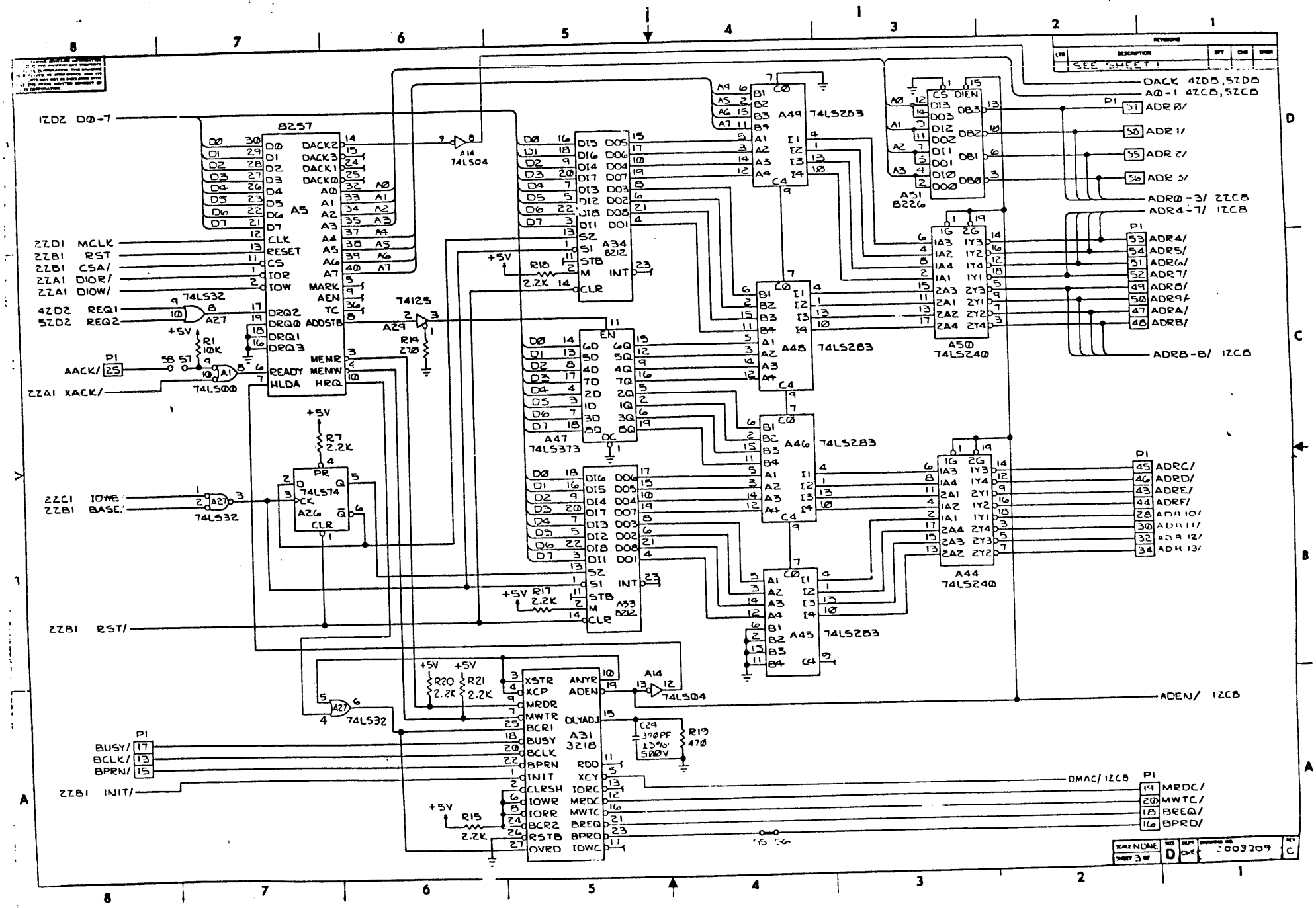
intel

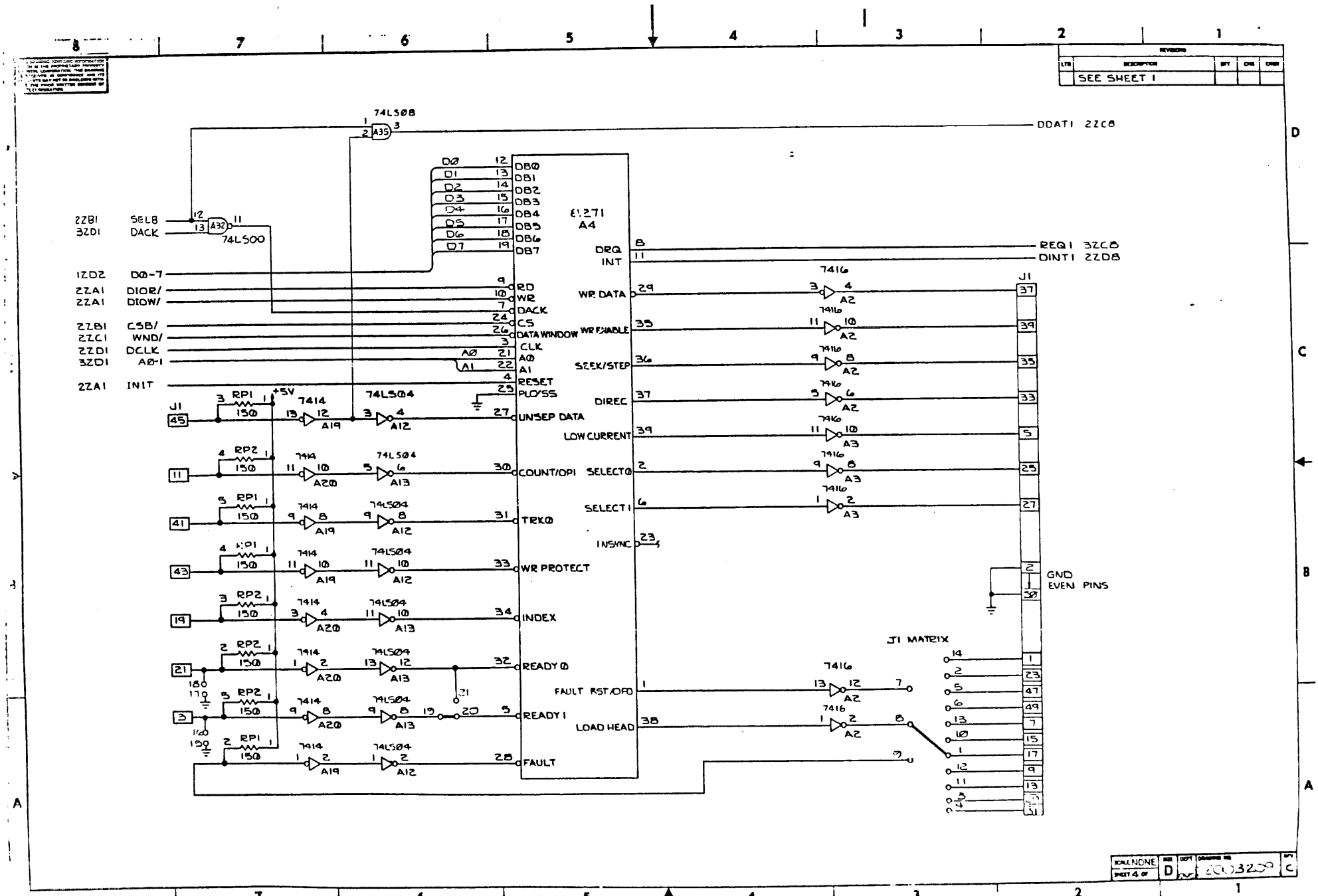
TITLE: SCHEMATIC, DISKETTE INTERFACE

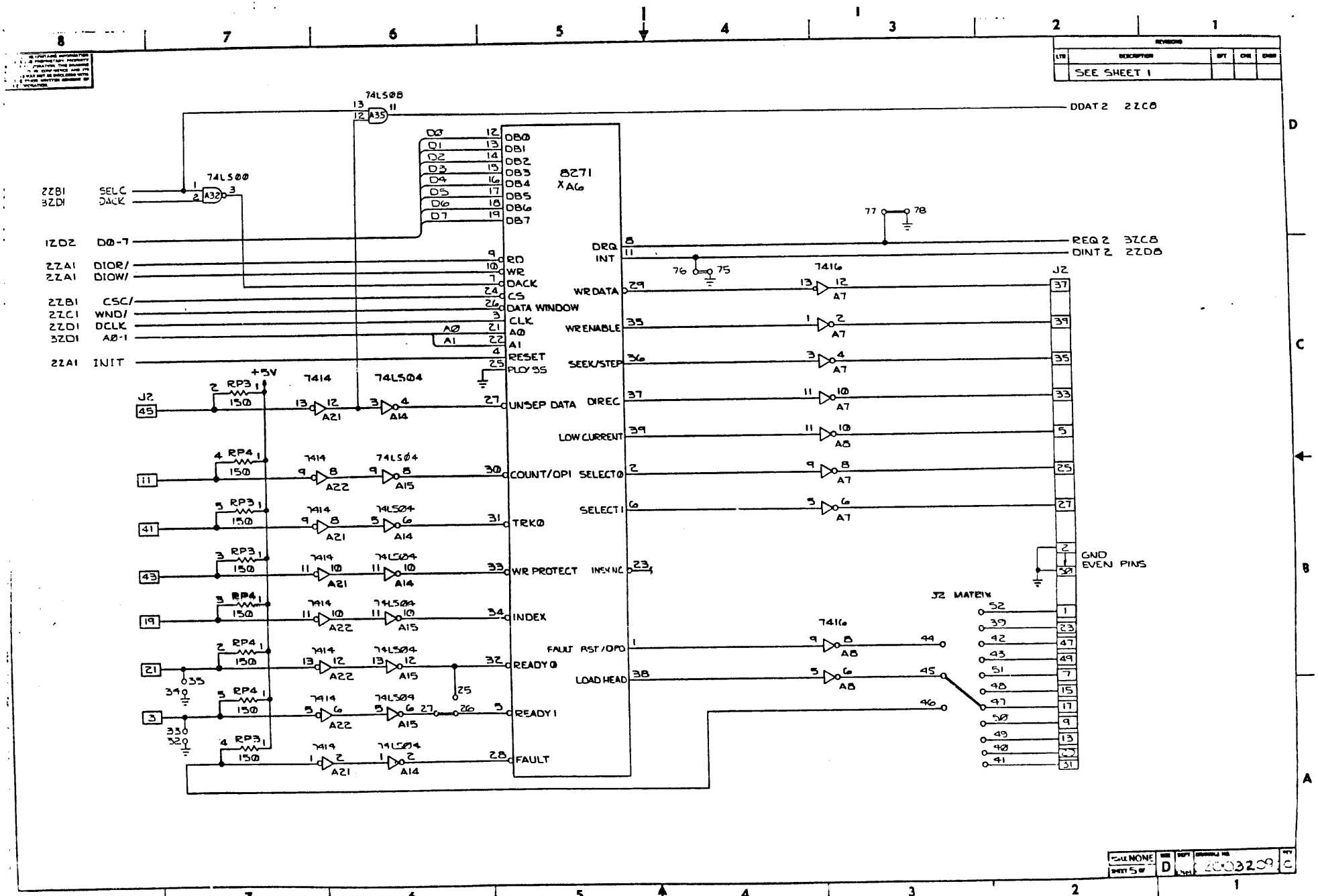
SIZE	DEPT	DRAWING NO.	REV
D	ONS	2003209	C

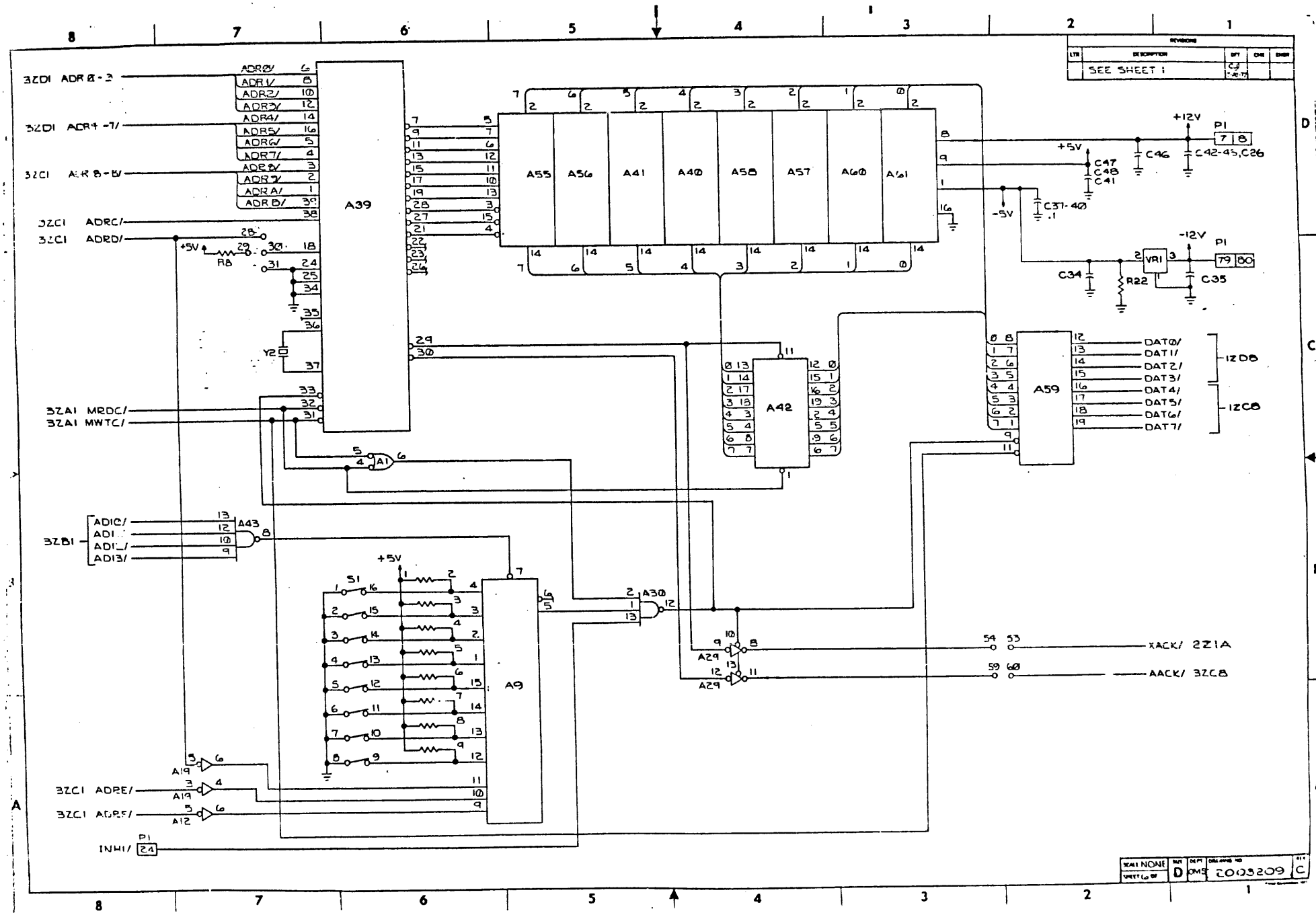
98-253A


SEE SHEET 1
4ZC2 DINT1
5ZC2 DINT2
INTERRUPT SELECT
INT7/
INT6/
INT5/
INT4/
INT3/
INT2/
INT1/
INT0/
8224
16MHZ
8MHZ
4MHZ
2MHZ
DCLK 4ZC8 5ZC8
MCLK 3ZD8
5ZD2 DDAT2
4ZD2 DDAT1
1ZD2 D0
IOWB/ 3ZB8
ADR0/
ADR1/
ADR2/
ADR3/
3ZD1 ADR0-3/
WND/ 4ZC8,5ZC8
1ZC2 UBDC/
BASE/ 3ZB8
CSA/ 3ZC8
SEIC 5ZD8
CSC/ 5ZC8
SELB 4ZD8
CSB/ 4ZC8
RST 3ZC8
RST/ 3ZB8
INIT/ 3ZA8
INIT/
1ZD2 LBDC
BDSEL/ 1ZC8
INIT 4ZC8,5ZC8
XACK/ 3ZC8
IORC/
IOWC/
DIOR/ 1ZC8,3ZC8, 4ZC8,5ZC8
DIOW/ 3ZC8,4ZC8, 5ZC8
74LS193
74LS574
74LS161
74LS00
74LS02
74LS04
74LS08
74LS10
74LS32
74LS164
74125
7414
74LS138
SCALE NONE
SHEET 2 OF
SIZE D
DRAWING NO. 7003209
REV C


8257
A5
74LS283
74LS240
74LS373
74LS574
74LS532
74LS00
74LS04
74125
3218
IZD2 D0-7
ZZD1 MCLK
ZZB1 RST
ZZB1 CSA/
ZZA1 DIOR/
ZZA1 DIOW/
4ZD2 REQ1
5ZD2 REQ2
AACK/
ZZA1 XACK/
ZZC1 IOWE
ZZB1 BASE/
ZZB1 RST/
ZZB1 INIT/
BUSY/
BCLK/
BPRN/
DACK 4ZD8,5ZD8
A0-1 4ZC8,5ZC8
ADR0/
ADR1/
ADR2/
ADR3/
ADR0-3/ ZZC8
ADR4-7/ IZC8
ADR4/
ADR5/
ADR6/
ADR7/
ADR8/
ADR9/
ADRA/
ADRB/
ADR8-B/ IZC8
ADRC/
ADRD/
ADRE/
ADRF/
ADR10/
ADR11/
ADR12/
ADR13/
ADEN/ IZC8
DMAC/ IZC8
MRDC/
MWTC/
BREQ/
BPRD/
SEE SHEET 1
SCALE NONE
SIZE D
SHEET 3 OF
C0093209
REV C


REVISIONS
LTR DESCRIPTION DFT CHK DATE
SEE SHEET 1
74LS08
DDAT1 2ZC8
2ZB1 SELB
3ZD1 DACK
74LS00
1ZD2 D0-7
2ZA1 DIOR/
2ZA1 DIOW/
2ZB1 CSB/
2ZC1 WND/
2ZD1 DCLK
3ZD1 A0-1
2ZA1 INIT
D0 D1 D2 D3 D4 D5 D6 D7
DB0 DB1 DB2 DB3 DB4 DB5 DB6 DB7
RD
WR
DACK
CS
DATA WINDOW
CLK
A0
A1
RESET
PLO/SS
UNSEP DATA
COUNT/OPI
TRK0
WR PROTECT
INDEX
READY 0
READY 1
FAULT
E1271
A4
DRQ
INT
WR DATA
WR ENABLE
SEEK/STEP
DIREC
LOW CURRENT
SELECT 0
SELECT 1
INSYNC
FAULT RST/OPO
LOAD HEAD
REQ1 3ZC8
DINT1 2ZD8
RP1
RP2
150
+5V
7414
74LS04
7416
A19
A12
A20
A13
A2
A3
J1
GND
EVEN PINS
J1 MATRIX
SCALE NONE
SHEET 4 OF
SIZE D
DRAWING NO 2003209
REV C


REVISIONS
LTR DESCRIPTION DFT CHK ENGR
SEE SHEET 1
74LS08
DDAT 2 2ZC8
74LS00
2ZB1 SELC
3ZD1 DACK
1ZD2 D0-7
2ZA1 DIOR/
2ZA1 DIOW/
2ZB1 CSC/
2ZC1 WND/
2ZD1 DCLK
3ZD1 A0-1
2ZA1 INIT
8271
XA6
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
RD
WR
DACK
CS
DATA WINDOW
CLK
A0
A1
RESET
PLOY SS
DRQ
INT
REQ 2 3ZC8
DINT 2 2ZD8
77
78
76
75
7416
J2
WR DATA
WR ENABLE
SEEK/STEP
DIREC
LOW CURRENT
SELECT 0
SELECT 1
GND
EVEN PINS
+5V
RP3
RP4
150Ω
7414
74LS04
A21
A14
A22
A15
A7
A8
UNSEP DATA
COUNT/OPI
TRK0
WR PROTECT
INDEX
READY 0
READY 1
FAULT
INSYNC
FAULT RST/OPO
LOAD HEAD
J2 MATRIX
035
330
320
25
26
27
SIZE NONE
D
SHEET 5 OF
2003209
REV
C


SEE SHEET 1
3ZDI ADR Ø-3
3ZDI ADR4-7/
3ZCI ADR8-B/
3ZCI ADRC/
3ZCI ADRD/
A39
A55
A56
A41
A40
A58
A57
A60
A61
+12V
+5V
-5V
-12V
C47
C48
C41
C46
C42-45,C26
C37-40
C34
R22
C35
VR1
R8
Y2
3ZAI MRDC/
3ZAI MWTC/
A42
A59
DAT0/
DAT1/
DAT2/
DAT3/
DAT4/
DAT5/
DAT6/
DAT7/
1ZDB
1ZCB
3ZBI
ADIC/
ADI
ADI_/
ADI3/
A43
A1
A30
A29
A9
S1
3ZCI ADRE/
3ZCI ADRF/
A19
A12
INHI/
XACK/ 2Z1A
AACK/ 3ZCB
SCALE NONE
SIZE D
DWG NO 2003209
REV C