iSBC 220™
SMD DISK CONTROLLER
HARDWARE REFERENCE MANUAL

Manual Order Number: 121597-001, REV A

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 its affiliates and may be used only to identify Intel products:

BXP	Intel	Megachassis
CREDIT	Intelevision	Micromap
i	Intellec	Multibus
ICE	iRMX	Multimodule
iCS	iSBC	PROMPT
im	iSBX	Promware
Insite	Library Manager	RMX/80
Intel	MCS	System 2000
		UPI
		μScope

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

CONTENTS

TABLES

ILLUSTRATIONS

PREFACE

This manual provides information regarding the installation, programming, operation, and servicing of the iSBC 220 SMD Disk Controller.

Related documents include:

- *The 8086 Family User's Manual,* Order No. 9800722
- *Intel MULTIBUS Specifications,* Order No. 9800683
- *Intel 8080/8085 Assembly Language Reference Manual,* Order No. 9800301
- *Intel 8086 Assembly Language Reference Manual,* Order No. 900640
- *MCS-80 User's Manual,* Order No. 9800153
- *MCS-85 User's Manual,* Order No. 9800722
- *Intel 8089 Assembly Language Reference Manual,* Order No. 9800938

CHAPTER 1
GENERAL INFORMATION

1-1. INTRODUCTION

The Intel iSBC 220™ SMD Disk Controller is designed to interface up to four Storage Module Device (SMD) Interface[1] compatible disk drives to any Intel Multibus™ interface compatible computer system. It can operate in a multiprocessor environment and is fully compatible with all Intel 8-bit and 16-bit computers. The Intel Multibus™ interface is the common interface between the iSBC 220 controller and the host computer, system memory, and other I/O boards. A typical drive subsystem is shown in figure 1-1.

Compatible disk drive storage range is from 12 to 600 megabytes. The number of tracks per surface, sectors per track, bytes per sector and alternate tracks per surface are software selectable for each drive unit.

The iSBC 220 controller's design is based on the Intel 8089 I/O Processor, which allows Direct Memory Access (DMA), error detecting and correction and data management. The single board assembly features automatic error recovery and retry, transparent data error correction and multiple sector transfers. Seek operations on multiple drives can be overlapped with a read/write operation on another drive. It is fully compatible with Intel 8086 CPU 20-bit addressing.

[1]*The Storage Module Device (SMD) Interface is a pending American National Standards Institute (ANSI) standard.*

1-2. DESCRIPTION

The iSBC 220 SMD Disk Controller is a single board assembly. It may reside in any Intel backplane or in a custom-designed configuration that is physically and electrically compatible with the Intel Multibus interface.

The host Central Processing Unit (CPU) communicates with the Disk controller via four blocks of information in host memory. Once the controller is

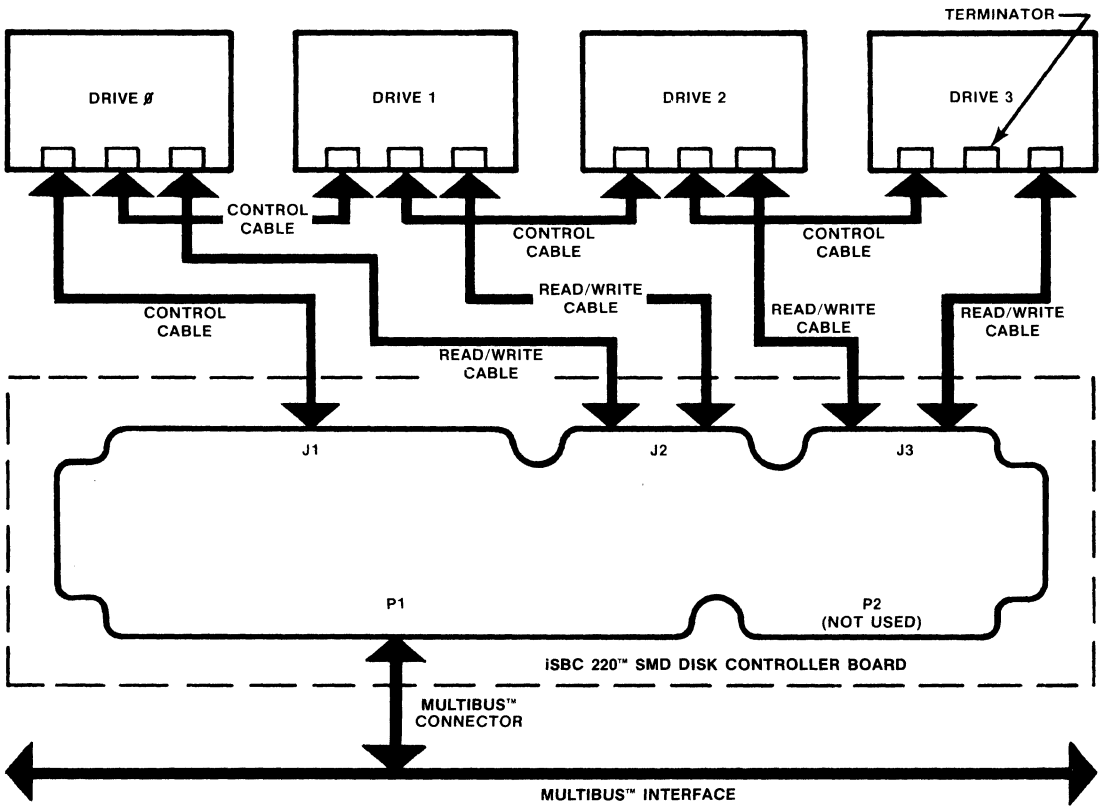


Figure 1-1. Typical Multiple Drive System

initialized, a CPU I/O write to the controller Wake-Up Address initiates disk activities. The controller accesses the four blocks in the host memory to determine the specific operation to be performed, fetches the required parameters and completes the specified operation without further CPU intervention.

The controller board generates all drive, control and data signals and receives the drive status and data signals required to perform the entire disk drive interfacing task. During a disk read operation, the controller accepts serial data from the disk, interprets sychronizing bit patterns, verifies validity of the data, performs a serial-to-parallel data conversion, and passes parallel data or error condition indications to host memory. During a disk write operation, the controller performs parallel-to-serial data conversion and transmits serial write data and the write clock to the drive.

The Intel 8089 I/O Processor provides optimum performance with minimum CPU overhead. An Intel 8288 Bus Controller and 8289 Bus Arbiter control access to the Multibus interface. Intel 2732 EPROMs and 2114 Static RAMs provide on-board local memory for storage of the controller I/O control program and a resident diagnostic exerciser, for data buffering and for temporary storage of read/write parameters.

To access system memory for a read or a write operation, the controller takes command of the Multibus interface and maintains control until the data transfer is complete. The buffer memory on the controller board limits this bus control time to a minimum.

1-3. SPECIFICATIONS

Table 1-1 lists the physical and performance specifications of the iSBC 220 SMD Disk Controller; table 1-2 lists typical characteristics of disk drives that are compatible with the iSBC 220 controller.

Table 1-1. Board Specifications

COMPATIBILITY	
CPU:	Any Intel mainframe or any Multibus™ interface compatible CPU. The controller can operate with either 16- or 20-bit addresses and with either 8- or 16-bit data bus widths.
Disk Drive:	SMD Interface compatible disk drive.
DATA ORGANIZATION AND CAPACITY	
Bytes per Sector:	128 / 256 / 512 / 1024 / Software Selectable
Sectors per Track (Maximum Allowable for Corresponding Selection of Bytes per Sector):	108 / 64 / 35 / 18 / Software Selectable
Disk Drives per Controller:	Up to four, daisy-chained.
Error Detecting and Correction:	Controller appends an Error Checking Code (ECC) at the end of each ID and data field. Using this ECC, the controller hardware can detect errors of up to 32 bits in length; controller firmware can correct errors of up to 11 bits in length.
CONTROLLER CHARACTERISTICS	
Mounting:	Occupies a card slot in iSBC 604/614 Modular Cardcage/Backplane or equivalent Multibus™ backplane connector.
Physical Characteristics:	
Width:	17.2 cm (6.8 inches)
Length:	30.5 cm (12.0 inches)
Height:	1.3 cm (0.5 inches)
Weight:	0.54 kg (19 ounces)
Power Requirements:	+5 Volts ±5% @ 3.25 amperes maximum; −5 Volts ±5% @ 0.75 amperes maximum.
	NOTE Jumper and on-board voltage regulator allow −5 Volts or −12 Volts from Multibus™ connector to be used as voltage source for −5 Volt.
Environmental:	
Temperature:	0°C to +55°C, operating (+32°F to +131°F). −55°C to +85°C, non-operating (−67°F to +185°F).
Humidity:	Up to 90%, non-condensing.

Table 1-2. Drive Characteristics (Typical)

Disk (spindle) Speed	3600 rpm
Tracks per Surface	823 typical
Servo Type	Closed loop, track following
Access Time	Track to track 6 ms Average 30 ms Maximum 55 ms
Data Transfer Rate	1.2 megabytes/second
Storage Capacity	12 to 600 megabytes

CHAPTER 2
PREPARATION FOR USE

2-1. INTRODUCTION

This chapter provides information for use in preparing and installing the iSBC 220 SMD Disk Controller. Included are instructions for unpacking and inspection, installation, setting switches, installing jumpers, and interfacing the controller board with the Multibus connector and disk drives.

2-2. UNPACKING AND INSPECTION

On receipt of the iSBC 220 controller from the carrier, immediately inspect the shipping carton for evidence of damage. If the shipping carton is damaged or water-stained, request that the carrier's agent be present when the carton is opened; if the carrier's agent is not present at the time of opening, keep the carton and packing materials for subsequent agent inspection.

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

Carefully unpack the shipping carton and verify that the following items are included:

- iSBC 220 SMD Disk Controller Printed Wired Assembly
- iSBC 220 SMD Disk Controller Schematic Diagram

2-3. BOARD INSTALLATION CONSIDERATIONS

The iSBC 220 controller can be installed in any Intel cardcage/backplane or any user-designed backplane that is compatible with the Multibus interface and meets the controller's power and Multibus connector dimensional requirements. The controller occupies one backplane slot.

When installing the controller in a serial priority environment (e.g., within any of the Intel system chassis), wiring modifications are required to support serial priority; a daisy-chain technique, see figure 2-1, establishes priority. The priority input (BPRN/) of the highest priority master is tied to ground. The priority output (BPRO/) of the highest priority master is then connected to the priority input (BPRN/) of the next lowest priority master, and so on. ("/" following the signal name indicates an active low). Slaves are jumpered as shown. This technique can accomodate a limited number of masters due to gate delays through the daisy-chain.

2-4. POWER REQUIREMENT

The board requires a +5 Volt ±5% power source at a maximum current of 3.25 amperes and a −5 Volt ±5% source at 0.75 amperes maximum, both supplied through the Multibus connector. An on-board voltage regulator allows the −5 Volt or −12 Volt voltage sources from the Multibus connector to be jumper selected for use as a voltage source for −5 Volts (refer to paragraph 2-13). Before installing the controller into a system chassis, make certain that the associated power supplies can supply the additional current that the controller board requires.

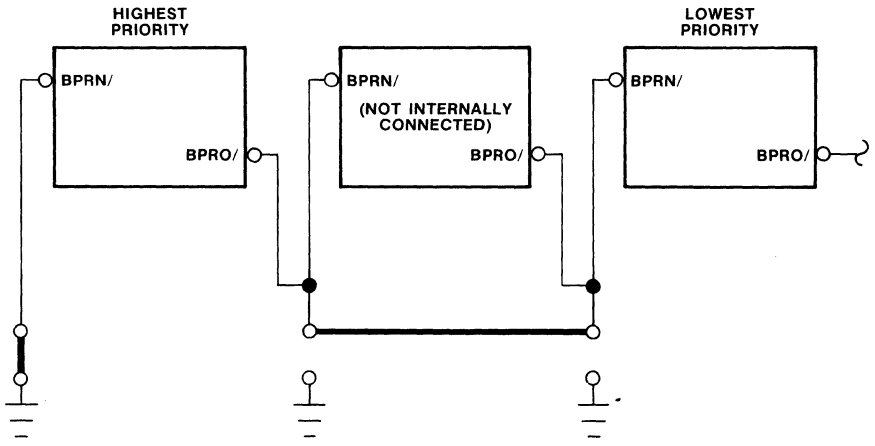


Figure 2-1. Serial Priority Resolution

2-5. COOLING REQUIREMENT

When the controller is installed in a high temperature environment, make certain the ambient operating temperature does not exceed +55°C.

2-6. MULTIBUS™ CONNECTOR

The controller communicates with the CPU and other boards via the Multibus interface. Table 2-1 lists the Multibus connector pin assignments; table 2-2 describes the controller/Multibus interface signals. Figure 2-2 provides a diagram of the controller/Multibus interface timing signals and a table of the timing requirements. Table 2-3 gives current requirements and other characteristics related to the controller/Multibus interface.

The controller is connected to the Multibus interface through connector P1, an 86-pin, double-sided, printed circuit edge connector with 3.96 mm (0.156 in) contact centers. Connector P2 is not used.

Table 2-1. Multibus™ Connector P1 Pin Assignment*

	P1 (Component Side)			P1 (Circuit Side)		
	Pin	Mnemonic	Description	Pin	Mnemonic	Description
Power Supplies	1	GND	Signal GND	2	GND	Signal GND
	3	+5V	+5Vdc	4	+5V	+5Vdc
	5	+5V	+5Vdc	6	+5V	+5Vdc
	7	+12V	+12Vdc	8	+12V	+12Vdc
	9	−5V	−5Vdc	10	−5V	−5Vdc
	11	GND	Signal GND	12	GND	Signal GND
Bus Controls	13	BCLK/	Bus Clock	14	INIT/	Initialize
	15	BPRN/	Bus Pri. In	16	BPRO/	Bus Pri. Out
	17	BUSY/	Bus Busy	18	BREQ/	Bus Request
	19	MRDC/	Mem Read Cmd	20	MWTC/	Mem Write Cmd
	21	IORC/	I/O Read Cmd	22	IOWC/	I/O Write Cmd
	23	XACK/	XFER Acknowledge	24	INH1/	Inhibit 1 disable RAM
Bus Controls and Address	25		Reserved	26	INH2/	Inhibit 2 disable PROM or ROM
	27	BHEN/	Byte High Enable	28	ADR10/	
	29	CBRQ/	Common Bus Request	30	ADR11/	Address
	31	CCLK/	Constant Clk	32	ADR12/	Bus
	33	INTA/	Intr Achknowledge	34	ADR13/	
Interrupts	35	INT6/	Parallel Interrput Requests	36	INT7/	Parallel Interrput Requests
	37	INT4/		38	INT5/	
	39	INT2/		40	INT3/	
	41	INT0/		42	INT1/	
Address	43	ADRE/	Address Bus	44	ADRF/	Address Bus
	45	ADRC/		46	ADRD/	
	47	ADRA/		48	ADRB/	
	49	ADR8/		50	ADR9/	
	51	ADR6/		52	ADR7/	
	53	ADR4/		54	ADR5/	
	55	ADR2/		56	ADR3/	
	57	ADR0/		58	ADR1/	
Data	59	DATE/	Data Bus	60	DATF/	Data Bus
	61	DATC/		62	DATD/	
	63	DATA/		64	DATB/	
	65	DAT8/		66	DAT9/	
	67	DAT6/		68	DAT7/	
	69	DAT4/		70	DAT5/	
	71	DAT2/		72	DAT3/	
	73	DAT0/		74	DAT1/	
Power Supplies	75	GND	Signal GND	76	GND	Signal GND
	77		Reserved	78		Reserved
	79	−12V	−12Vdc	80	−12V	−12Vdc
	81	+5V	+5Vdc	82	+5V	+5Vdc
	83	+5V	+5Vdc	84	+5V	+5Vdc
	85	GND	Signal GND	86	GND	Signal GND

* "/" following the signal name indicates an active low.

Table 2-2. iSBC 220™ Controller/Multibus Interface Signal Descriptions

Signal	Functional Description
ADR0/, ADRF/ ADR10/-ADR13/	*Address.* These 20 lines transmit the address of the memory location or I/O port to be accessed. For memory access, ADR0/ (when active) enables the even byte bank (DAT0/-DAT7/) on the Multibus™ connector; i.e., ADR0/ is active for all even addresses. ADR13/ is the most significant address bit.
BCLK/	*Bus Clock.* Used to synchronize the bus contention logic on all bus masters.
BHEN/	*Byte High Enable.* When active low, enables the odd byte bank (DAT8/-DATF/) onto the Multibus™ connector.
BPRN/	*Bus Priority In.* When low indicates to a particular bus master that no higher priority bus master is requesting use of the bus. BPRN/ is synchronized with BCLK/.
BPRO/	*Bus Priority Out.* In serial (daisy chain) priority resolution schemes, BPRO/ must be connected to the BPRN/ input of the bus master with the next lower bus priority.
BREQ/	*Bus Request.* In parallel priority resolution schemes, BREQ/ indicates that a particular bus master requires control of the bus for one or more data transfers. BREQ/ is synchronized with BCLK/.
BUSY/	*Bus Busy.* Indicates that the bus is in use and prevents all other bus masters from gaining control of the bus. BUSY/ is synchronized with BCLK/.
CBRQ/	*Common Bus Request.* Indicates that a bus master wishes control of the bus but does not presently have control. As soon as control of the bus is obtained, the requesting bus controller raises the CBRQ/ signal.
DAT0/-DATF/	*Data.* These 16 bidirectional data lines transmit and receive data to and from the addressed memory location or I/O port. DATF/ is the most-significant bit. For data byte operations, DAT0/-DAT7 is the even byte and DAT8-DATF/ is the odd byte.
INIT/	*Initialize.* Reset the entire system to a known internal state.
INT0/-INT7/	*Interrupt Request.* These eight lines transmit interrupt requests to the appropriate interrupt handler. INT0/ has the highest priority.
IOWC/	*I/O Write Command.* Indicates that the address of an I/O port is on the Multibus™ connector address lines and that the contents on the Multibus™ connector data lines are to be accepted by the addressed port.
MRDC/	*Memory Read Command.* Indicates that the address of a memory location is on the Multibus™ connector address lines and that the contents of that location are to be read (placed) on the Multibus™ connector data lines.
MWTC/	*Memory Write Command.* Indicates that the address of a memory location is on the Multibus™ connector address lines and that the contents on the Multibus™ connector data lines are to be written into that location.
XACK/	*Transfer Acknowledge.* Indicates that the address memory location has completed the specified read or write operation. That is, data has been placed onto or accepted from the Multibus™ connector data lines.

Table 2-3. iSBC 220™ Controller/Multibus Interface Signal Characteristics

Bus Signals	Driver 1, 3					Receiver 2, 3			
	Location	Type	I_{OL} Min_{ma}	I_{OH} $Min_{\mu a}$	C_O Min_{pf}	Location	I_{IL} Max_{ma}	I_{IH} $Max_{\mu a}$	C_I Max_{pf}
DAT0/-DATF/ (16 lines)	Masters	TRI	32	−5000	300	Masters and Slaves	−0.5	125	18
ADR0/-ADR13/, BHEN/ (21 lines)	Masters	TRI	32	−5000	300	Slaves	−0.8	90	18
MRDC/, MWTC/	Masters	TRI	32	−5000	300	Slaves	−0.7	50	18
IOWC/						Slaves	−0.4	20	5

Table 2-3. iSBC 220™ Controller/Multibus Interface Signal Characteristics (Continued)

Bus Signals	Driver 1, 3				Receiver 2, 3				
	Location	Type	I_{OL} Min_{ma}	I_{OH} $Min_{\mu a}$	C_O Min_{pf}	Location	I_{IL} Max_{ma}	I_{IH} $Max_{\mu a}$	C_I Max_{pf}
XACK/	Slaves	TRI	48	−2000	300	Masters	−1.2	60	18
BCLK/						Master	−0.5	60	18
BREQ/	Each Master	TTL	10	−400	60				
BPRO/	Each Master	TTL	10	−400	60				
BPRN/						Master	−0.5	60	18
BUSY/, CBRQ	All Masters	O.C.	20	-	250	All Masters	−0.5	60	18
INIT/						All	−0.5	60	18
INT0/-INT7/ (8 lines)	Slaves	O.C.	40	-	300				

Notes:

1. Driver Requirements:
   - I_{OH} = High Output Current Drive
   - I_{OL} = Low Output Current Drive
   - C_O = Capacitance Drive Capability
   - TRI = 3-State Drive
   - O.C.= Open Collector Driver
   - TTL = Totem-pole Driver
2. Receiver Requirements:
   - I_{IH} = High Input Current Load
   - I_{IL} = Low Input Current Load
   - C_I = Cap Active Load
3. Low and High Voltage Requirements:

   Receiver:

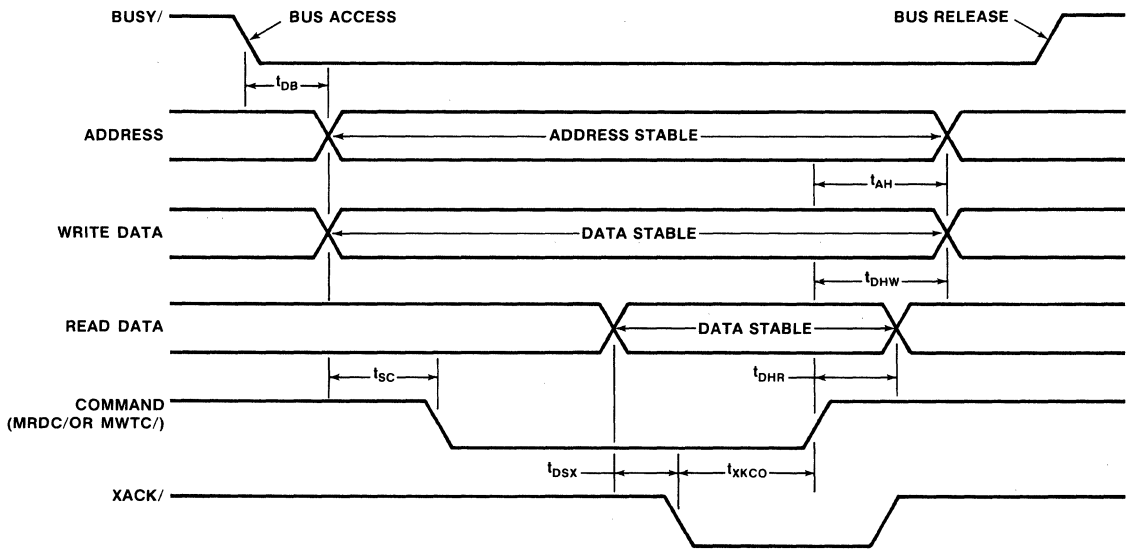
   $0 \leqq V_{IL} \leqq 0.8V$

   $2.0V \leqq V_{IH} \leqq 5.5V$

   Driver:

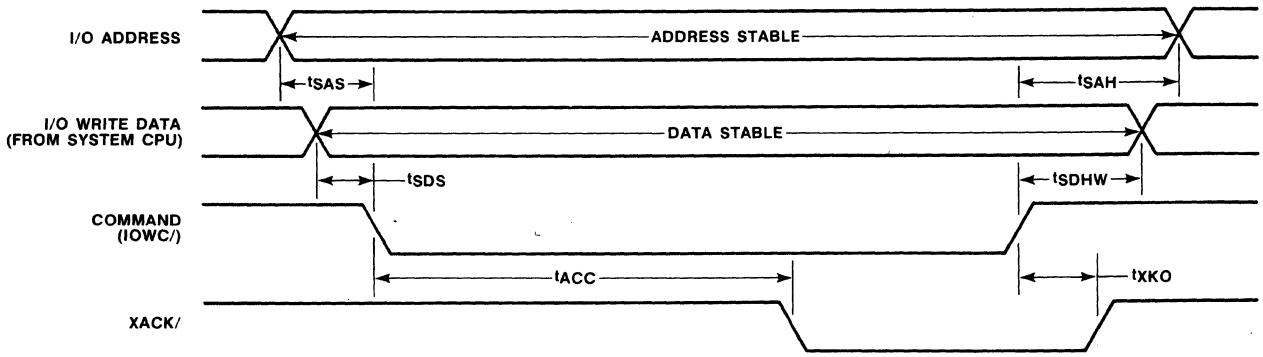
   $0 \leqq V_{OL} \leqq 0.5V$

   $2.4V \leqq V_{OH} \leqq 5.5V$

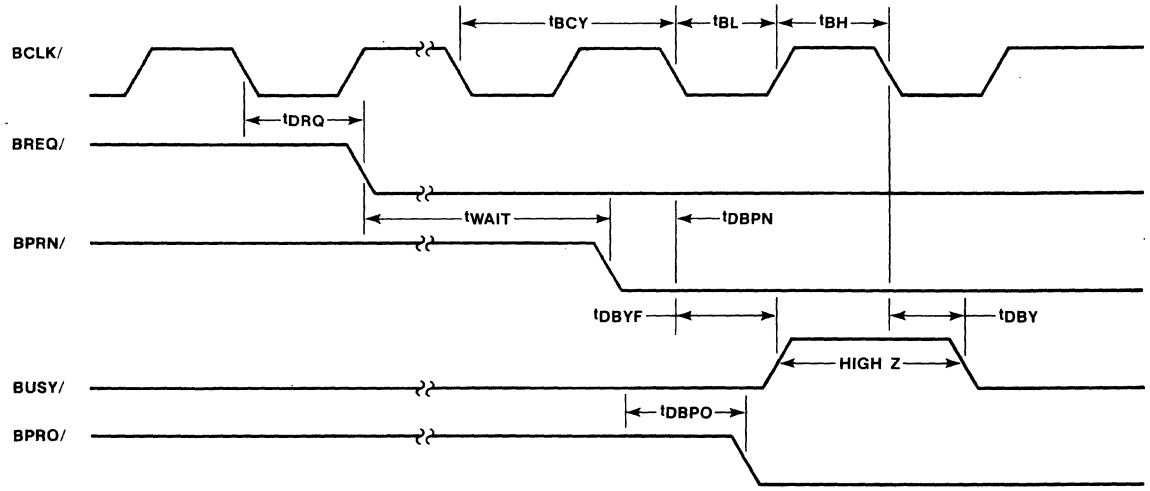


Master Command Access Timing

Figure 2-2. iSBC 220™ Controller/Multibus Interface Signal Timing



Slave Command Timing



Bus Exchange Timing

Parameter	Time in Nanoseconds		Description
	Minimum	Maximum	
t_{SAS}	50		Address Setup Time to I/O Command
t_{SDS}	0		Data Setup Time to I/O Command
t_{SAH}	15		Address Hold Time from I/O Command
t_{SDHW}	30		Data Hold Time from I/O Command
t_{ACC}		8000	I/O Access Time
t_{XKO}	100		XACK/Hold Time from I/O Command
t_{BCY}	125		Bus Clock Cycle Time
t_{BL}	65		Bus Clock Low
t_{BH}	35		Bus Clock High
t_{DRQ}		35	Bus Request Delay
t_{DBY}		60	Bus Busy Turn On Delay
t_{DBYF}		35	Bus Busy Turn Off Delay
t_{DBPN}	15		Priority Input Setup Time
t_{DBPO}		25	BPRO/Serial Delay from BPRN/
t_{WAIT}		∞	Requesting Master Bus Access Time
t_{DB}	50		Busy to Address/Data Delay
t_{SC}	50		Address/Data Setup to Command
t_{XKCO}		750	XACK/ to Command Turn Off
t_{AH}	50		Address Hold Time
t_{DHW}	50		Data Hold Time
t_{DHR}	0		Read Data Hold Time
t_{DSX}	0		Data Setup Time Before XACK/

Figure 2-2. iSBC 220™ Controller/Multibus Interface Signal Timing (Continued)

2-7. SWITCH/JUMPER CONFIGURATIONS

A number of switches and jumper links (see table 2-4) are provided on the controller board that allow the user to conveniently set the controller for the system environment in which it is to operate (8-bit or 16-bit system data bus, 8-bit or 16-bit I/O addressing, etc.). Figure 2-3 shows the location of these switches and jumpers on the board. They should be set, as described in the following paragraphs, prior to installing the board in a cardcage or backplane.

Table 2-4. Configuration Linkages and Switches

Function	Pin or Switch No.
Wake-Up Address	S1-1 through S1-8 S2-3 through S2-10
8-Bit or 16-Bit System Data Bus Compatibility	S2-1
8-Bit or 16-Bit Host I/O Processor Port Addressing	S2-2
Interrupt Priority Level	W4-C to W4-0 through W4-7
Any Request	W2
Voltage Selection	W1 and W5

2-8. WAKE-UP ADDRESS SELECTION

The controller communicates with the host CPU through four I/O communications blocks located in the host memory. When the controller is to receive instructions, it goes to the beginning address of the first I/O communication block. This address is called the wake-up address (WUA). The WUA may be any address in host memory. Sixteen WUA switches (S1-1) through S1-8 and S2-3 through S2-10, see figure 2-3) are provided on the controller board that allow the user to set the controller for the selected wake-up address. The switch labels in figure 2-3 correspond to hexadecimal address bits 0 through F. Any switch set to ON represents a logical 1.

The controller multiplies the settings of the WUA switches by 2^4 (shifts the number four places to the left) to create a 20-bit WUA. Note that due to this shift, the four least-significant bits of the selected WUA must be zeros. When accessing host memory, the controller transmits the entire 20-bit WUA through the Multibus interface. If the host memory uses 16-bit addressing, the four most significant bits of the 20-bit WUA must be zero. This is accomplished by setting the four most significant bits of the WUA switches (S1-1 through S1-4) to zero.

2-9. WAKE-UP I/O PORT ADDRESS SELECTION

The host processor communicates with the controller through an 8-bit I/O port. The WUA switches also set the address of this I/O port. For a host processor with 8-bit I/O port addressing, bits 0 through 7 of the unshifted WUA determine the wake-up I/O port address; for a host processor with 16-bit I/O port addressing, bits 0 through F determine the address.

I/O Address Selection switch S2-2 on the controller board (see figure 2-3) determines the type of I/O port addressing the host processor uses: ON for 16-bit addressing; OFF for 8-bit addressing.

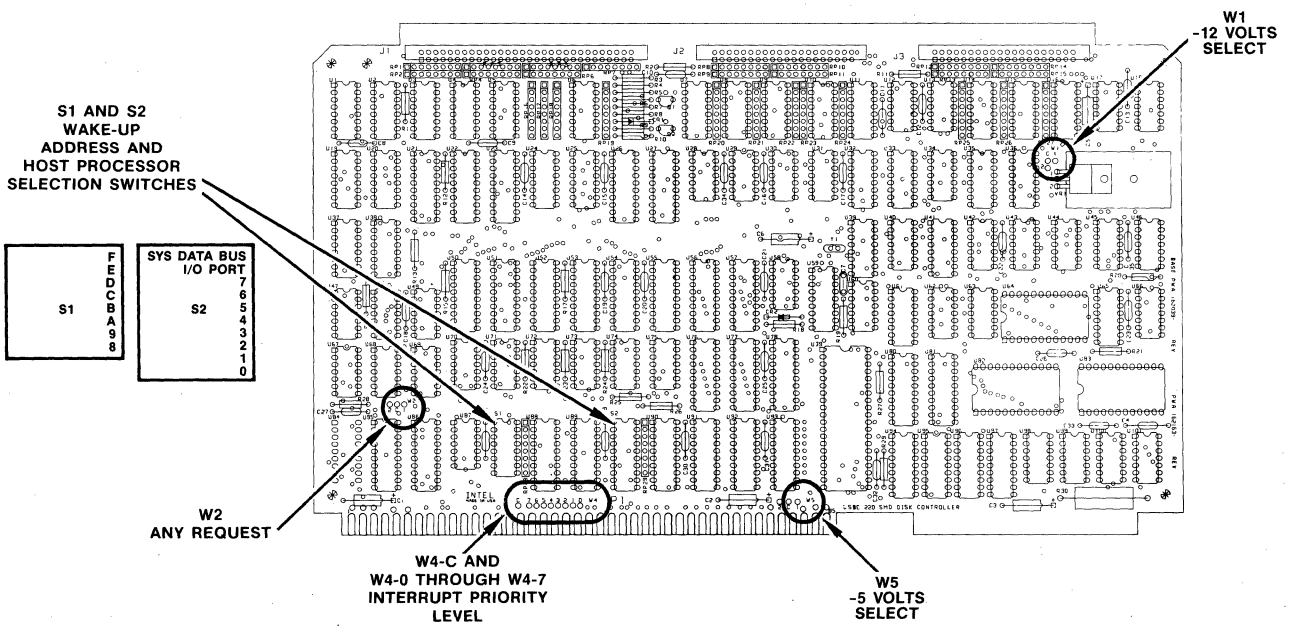


Figure 2-3. Location of Jumpers and Switches on Controller Board

2-10. SYSTEM DATA BUS SELECTION

Host processor selection switch S2-1 on the controller board (see figure 2-3) sets the controller for the type of system data bus with which the controller is to interface: ON for 16-bit bus, OFF for an 8-bit bus. This switch allows the controller to use its 16-bit data transfer mode to access the system bus (if the system memory supports 16-bit accesses), even though the host processor only supports 8-bit accesses.

2-11. INTERRUPT PRIORITY LEVEL

The controller's internal interrupt request signal can be assigned to any of eight interrupt priority levels (INT0/ to INT7/) on the Multibus connector. To select the interrupt request priority level, place a jumper link as shown in table 2-5 and figure 2-3.

2-12. ANY REQUEST SELECTION

The any request function allows the controller to be set to relinquish control of the Multibus interface following a request from:

1. A higher priority device only (jumper between pins W2-C and W2-1 on the controller board).
2. Any device, lower or higher priority, (jumper between pins W2-C and W2-2).

Figure 2-3 shows the location of the selection pins.

2-13. VOLTAGE SELECTION

Figure 2-3 shows the location on the controller board of the Voltage Source Selection pins for the −5 Volt power supply. Install a jumper at either W5 (−5 Volts) or W1 (−12 Volts) to select a voltage source for the on-board −5 Volt Supply.

2-14. DRIVE INTERFACE

The iSBC 220 SMD Disk Controller is designed for compatibility with SMD Interface compatible disk drives. Two interface cables per drive are required, one that daisy-chains command information (Control Cable) and another that provides the serial data (Read/Write Cable). The controller can support up to four drives. Refer to paragraphs 4-22 through 4-28 for a detailed description of the controller to drive interface signals.

2-15. CABLING REQUIREMENTS

Unless the drive manufacturer supplies them, the Interface cables between the controller and disk drives will have to be fabricated (see figure 2-4). Right-angle pin header connectors with ejector tabs are recommended for mating with each of the controller board's edge connectors. A 60-pin mass-terminated socket connector (3M 3334-6060 or equivalent) is recommended for mating with J1; a 40-pin connector (3M 3417-6040 or equivalent) is recommended for mating with J2 and J3. The mass terminated sockets are easily attached to flat ribbon cable using the jig that the connector manufacturer supplies. The Control Cable, which connects to J1, requires a 60-conductor ribbon cable; the Read/Write Cables, which connect to J2 and J3, each require one or two 20-conductor ribbon cables, depending on the number of drives in the installation (refer to paragraph 2-17 and 2-18 below). Cable length for the Control Cable cannot exceed a total length of 100 feet; total length for any Read/Write cable must not exceed 50 feet.

2-16. DRIVE PIN ASSIGNMENTS

Tables 2-6 and 2-7 list the pin assignments for the J1, J2, and J3 connectors of the controller and the J1 through J4 connectors of the drives.

2-17. SINGLE DRIVE INSTALLATIONS

For single drive installations, two controller-to-drive interface cables are required. The 60-conductor Control Cable connects between J1 of the controller and the corresponding control cable connector on the drive. The Control Cable lines must be terminated as described in the drive manufacturer's hardware reference manual at the drive terminator. The 20-conductor Read/Write cable connects between either J2 or J3 of the controller and the corresponding read/write cable connector on the drive. In addition, jumpers or switches within the drive must be set to assign the drive a logical address. Since the controller can communicate with up to four drives, the logical address can be set to 0, 1, 2 or 3.

2-18. MULTIPLE DRIVE INSTALLATIONS

For multiple drive installations, a common Control Cable bus transmits control data between the controller and the drives (see figure 1-1). The Control Cable is connected between J1 of the controller and the corresponding control cable connector on the drive at physical address 0. Drive-to-drive control cables are then installed to daisy-chain the control data to the other drives. The control lines must be terminated at the last drive on the bus.

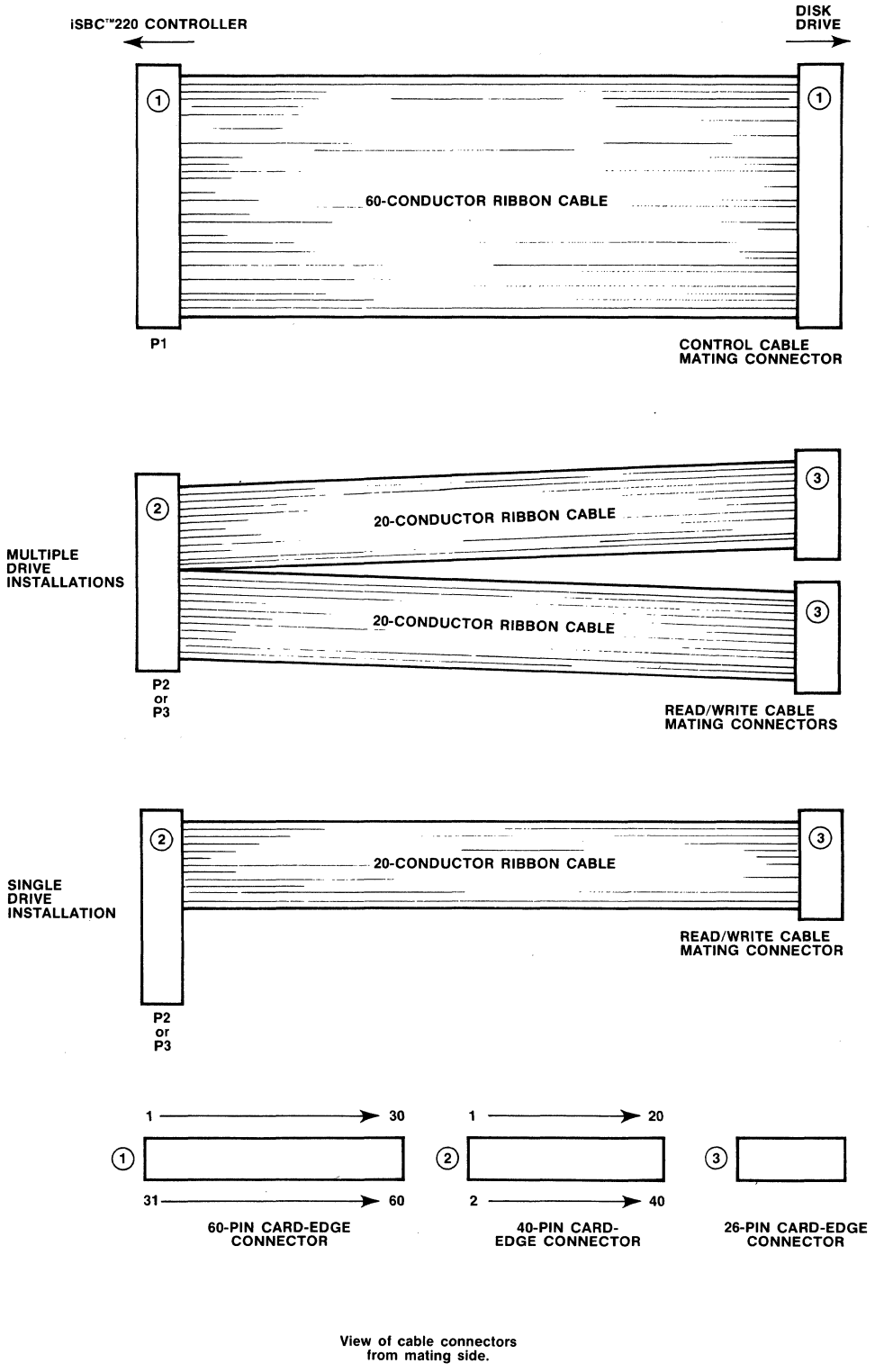


Figure 2-4. Interconnecting Cable Requirements

Table 2-5. Interrupt Priority Level Selection

Priority Level Selected	Wire Wrap	
	From Pin	To Pin
0	W4-C	W4-0
1	W4-C	W4-1
2	W4-C	W4-2
3	W4-C	W4-3
4	W4-C	W4-4
5	W4-C	W4-5
6	W4-C	W4-6
7	W4-C	W4-7

Table 2-6. Control Cable Signal/Pin List

Signal Name	J1 Asserted State Pin Polarity		Source	Drive Input Connector	
	−	+		−	+
Device Select 0	23	53	CU	23	53
Device Select 1	24	54	CU	24	54
Device Select 2	26	56	CU	26	56
Device Select 3	27	57	CU	27	57
Device Select Enable	22	52	CU	22	52
Set Cylinder (Tag 1)	01	31	CU	01	31
Set Head Address (Tag 2)	02	32	CU	02	32
Control Select (Tag 3)	03	33	CU	03	33
Bus 0	04	34	CU	04	34
Bus 1	05	35	CU	05	35
Bus 2	06	36	CU	06	36
Bus 3	07	37	CU	07	37
Bus 4	08	38	CU	08	38
Bus 5	09	39	CU	09	39
Bus 6	10	40	CU	10	40
Bus 7	11	41	CU	11	41
Bus 8	12	42	CU	12	42
Bus 9	13	43	CU	13	43
Interface Enable	14	44	CU	14	44
Index Mark	18	48	Drive	18	48
Sector Mark*	25	55	Drive	25	55
Fault	15	45	Drive	15	45
Seek Error	16	46	Drive	16	46
On Cylinder	17	47	Drive	17	47
Unit Ready	19	49	Drive	19	49
Write Protected	28	58	Drive	28	58
Address Mark	20	50	Drive	20	50
Reserved	21	51	-	21	51
Pick	29	-	CU	29	-
Sequence Disable	59	-	CU	59	-
Reserved	30	60	-	30	60

*Not Used

Table 2-7. Read/Write Cable Signal/Pin List

Signal Name	Controller I/O Connector	Drive I/O Conn.	
Servo Clock	J2/3-02,21	JXX-02,14	
Gnd	01	01	
Read Data	03,23	03,16	
Gnd	22	15	
Read Clock	05,24	05,17	
Gnd	04	04	Physical Drive 0 and 2
Write Clock	06,26	06,19	
Gnd	25	18	
Write Data	08,27	08,20	
Gnd	07	07	
Unit Selected	29,09	22,09	
Gnd	28	21	
Seek End	10,30	10,23	
Servo Clock	12,31	JXX-02,14	
Gnd	11	01	
Read Data	13,33	03,16	
Gnd	32	15	
Read Clock	15,34	05,17	
Gnd	14	04	Physical Drive 1 and 3
Write Clock	16,36	06,19	
Gnd	35	18	
Write Data	18,37	08,20	
Gnd	17	07	
Unit Selected	39,19	22,09	
Gnd	38	21	
Seek End	20,40	10,23	

Note: Each signal is a differential pair with pin numbers given by: JXX-1st pin, 2nd pin.

Two 20-conductor Read/Write Cables (see figure 2-4) connect between J2 and the corresponding read/write cable connectors on the drives at physical address 1 and 2. Read/write data can be transmitted between the controller and additional drives in the same manner through connector J3, with the two 20-conductor Read/Write Cables going to the drives at physical addresses 2 and 3. As has been described for single drive installations, the logical address of each drive is set at each drive.

If only two drives are to be connected to the controller, connect a single 20-conductor Read/Write Cable between J2 of the controller and the first drive and another cable between J3 and the second drive. This interconnection method eliminates the need to construct a split (double) Read/Write Cable.

2-19. POWER UP/DOWN CONSIDERATIONS

If power is applied to, or removed from, the system while a drive is READY, a spurious disk write operation could occur. To prevent this from happening always ensure that the drives are not spinning when system power to the controller is switched on or off.

2-20. DIAGNOSTIC CHECK

A PROM-resident self-diagnostic may be used to verify the controller operation. Instructions for execution of the diagnostic are given in chapter 3.

CHAPTER 3
PROGRAMMING INFORMATION

3-1. INTRODUCTION

This chapter describes the programming conventions that must be followed to initiate and monitor the transfer of data between the host memory and a disk drive. Included in this section are a discussion of: disk organization, track sectoring format, disk controller communications protocol, interrupt handling and the use of disk control functions.

3-2. DISK ORGANIZATION

The iSBC 220 SMD Disk Controller can communicate with from one to four disk drive units. Each drive has a number of disk surfaces, which are fixed, removable or both. In the following discussion, a head is assumed to be associated with a single disk surface. Each surface can have up to 1024 tracks (circular data paths numbered 0 through 1023). The set of tracks on multiple recording surfaces at a given head position or location is referred to as a "cylinder" (see figure 3-1). A drive that has 1024 tracks per surface thus has 1024 cylinders.

Each track is divided into equal-sized sectors. Each of these sectors includes a sector identification block with error checking information and a data block, also with error checking information. The iSBC 220 controller allows the user to select the size of the data block; the size of the data block then determines the maximum number of sectors permitted per track (as shown in Table 3-1).

Table 3-1. Data Block Length vs. Sectors Per Track

Bytes Per Data Block	Maximum Number of Sectors Per Track
128	108
256	64
512	35
1024	18

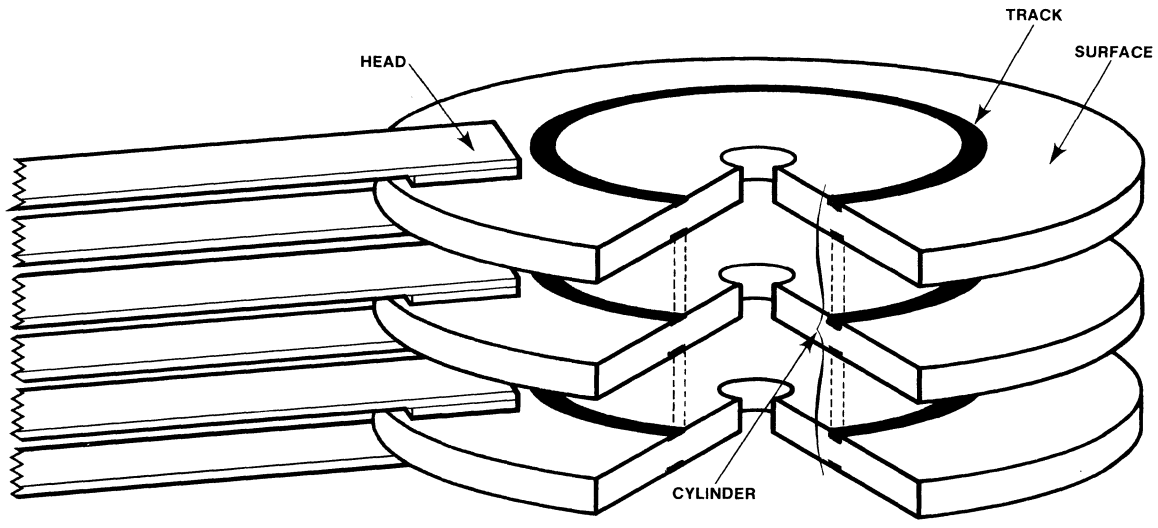


Figure 3-1. Disk Drive Organization and Terminology

3-3. TRACK SECTORING FORMAT

The controller generates the format of the sector identification block, the data block and the error checking fields of each sector of the disk, one track at a time. Figure 3-2 shows how the controller organizes this information. Refer to the paragraph 3-12 and 3-13 for further information on track formatting.

3-4. CONTROLLER I/O COMMUNICATIONS BLOCKS

The host processor and the disk controller use four blocks of host memory and one host I/O port to exchange instructions and status. The I/O communications blocks are titled: Wake-Up Block, Channel Control Block, Controller Invocation Block and I/O Parameter Block. Sixty-eight bytes of host memory must be dedicated to the I/O communications blocks.

NOTE

Following the initialization of the controller, the Wake-Up Block, Channel Control Block and Controller Invocation Block must be maintained at their assigned locations. The location of the I/O Parameter Block can be changed providing that the I/O Parameter Block Pointer in the Controller Invocation Block is changed to correspond to the new location.

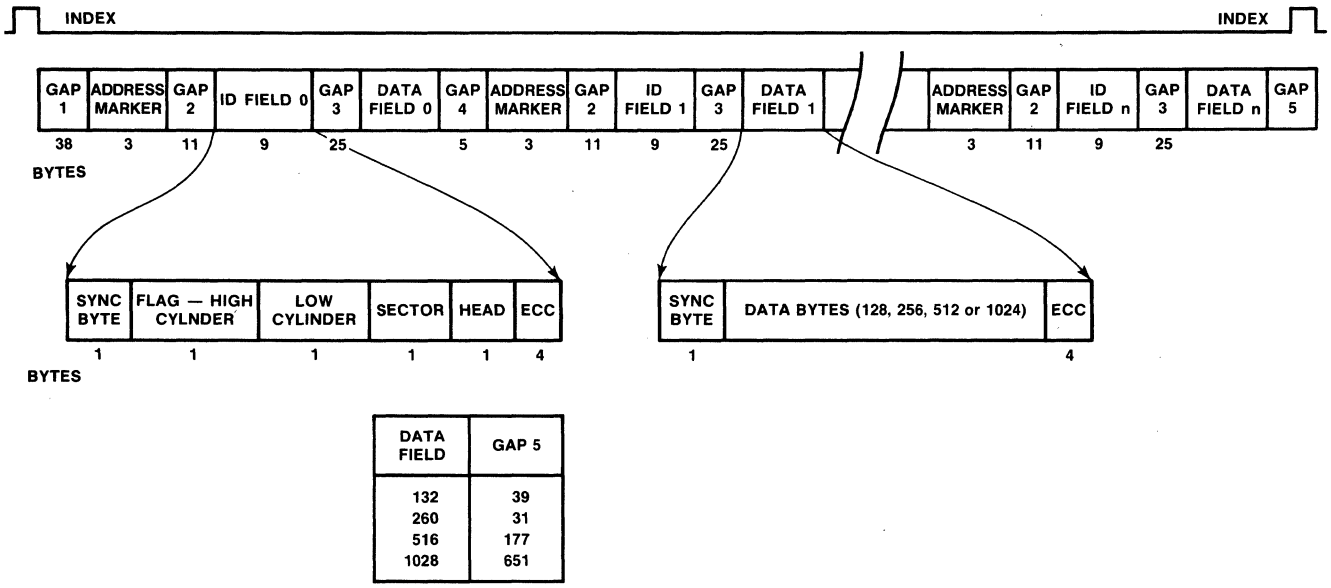
The controller uses these blocks to perform three basic functions: initialize the controller, check and transmit status, and obtain user selected disk access functions and parameters. In addition to these I/O communications blocks, certain controller functions (such as track formatting) also require data/parameter buffers in host memory. Dedicated locations in host memory, however, are not required for these buffers. One I/O port in the host processor's addressable I/O space is also required. The host uses this port, called the Wake-Up I/O Port, to initiate controller activity.

The sequence in which the controller accesses these blocks varies with the type of operation being performed, but for general data transfers (reads or writes), the blocks are accessed as follows:

(1) The host loads the I/O Parameter block in system memory with a command and parameters for the function the controller is to perform (for example read data). See Figure 3-3.

(2) The host then transmits a wake-up command (01H) to wake-up I/O port, signaling the controller to go to I/O communications blocks for instructions.

(3) The controller goes to the Channel Control Block and links its way through the Controller Invocation Block to the I/O Parameter Block. (The Wake-Up Block is used only during controller initialization and by 8089 firmware.)



DATA FIELD	GAP 5
132	39
260	31
516	177
1028	651

Figure 3-2. Sector Data Format.

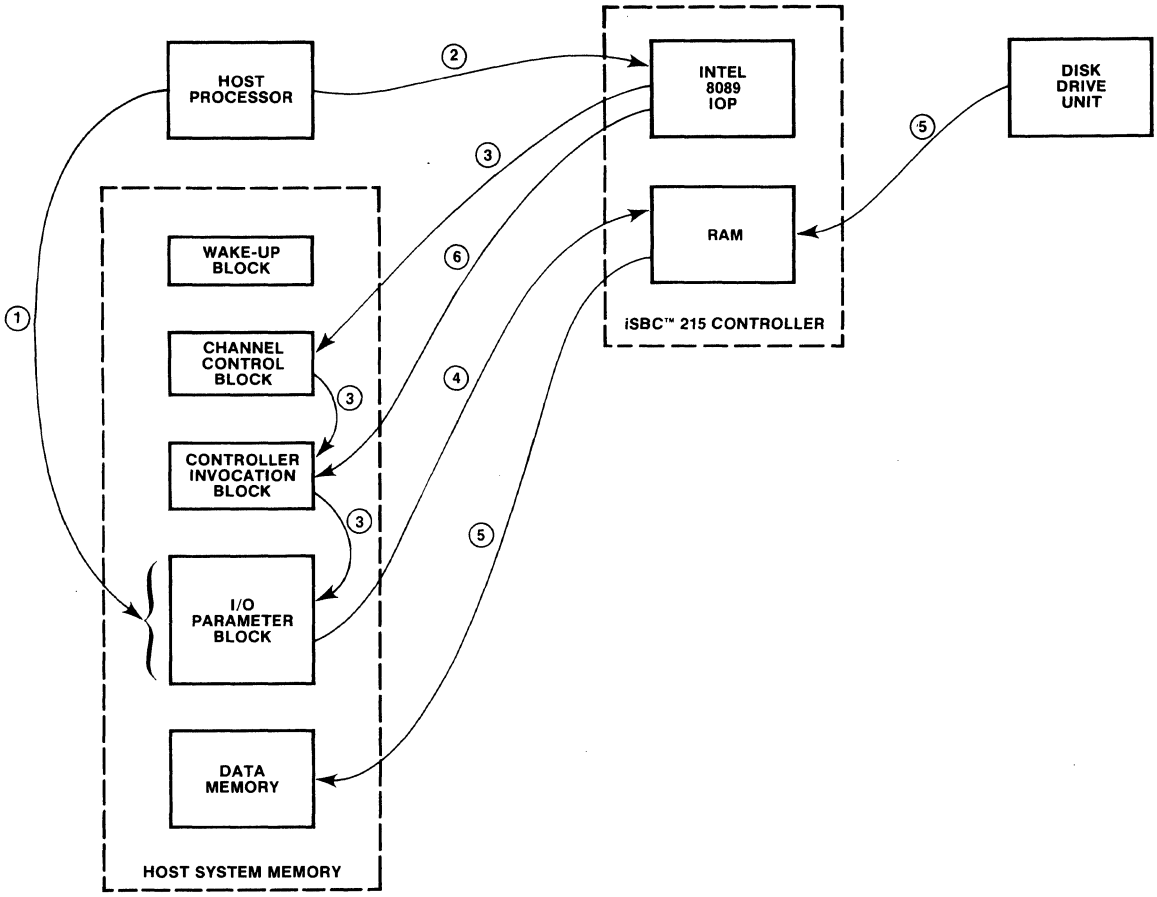


Figure 3-3. Host CPU-Disk Controller Interface Through the I/O Communications Blocks

④ At the I/O Parameter Block, the controller reads the command and parameter data into its RAM and begins the data transfer function.

⑤ The controller reads data from the selected drive into its RAM, then performs a DMA transfer of the data from RAM into system memory.

⑥ When the data transfer is complete, the controller posts the status in the Controller Invocation Block, sends an interrupt to the host and awaits further instructions.

These I/O communications blocks are accessed in a similar manner when performing a write function.

A detailed description of these blocks and the data required in each is provided in paragraphs 3-6 through 3-10. Refer to paragraphs 2-7 through 2-10 for a discussion of selecting the wake-up address, wake-up I/O port address and 8-bit or 16-bit host.

3-5. HOST CPU-CONTROLLER-DISK DRIVE INTERACTION

Figure 4-2 shows a simplified block diagram of the major hardware sections of the host CPU, host memory, controller and disk drives. The host system memory contains all the controller I/O communications blocks, as well as the data buffers. The host initiates controller activity through the wake-up I/O port, which it addresses through the Multibus interface. The Intel 8089 I/O processor (IOP) handles all communicates between the host CPU, host memory and disk drives, once the host has initiated controller activity. Controller oeprations software is contained in on-board PROM. RAM on the controller board facilitates intermediate data storage between the host and the disk drive.

3-6. WAKE-UP I/O PORT

To invoke controller activity, the host CPU transmits a wake-up command byte to the controller through

the wake-up I/O port. Three wake-up commands are allowed:

00H	CLEAR INTERRUPT — Controller to host interrupt is reset; controller reset is cleared.
01H	START OPERATION — Instructs controller to start the operation that the elements of the I/O parameter block define.
02H	RESET CONTROLLER — Performs hardware reset of controller. A clear interrupt (00H) must be initiated following this command. (Each time the controller is reset, the communications link between the controller and the host must be re-established through the Initializing function.)
03H through FFH	Reserved.

The sixteen wake-up address switches on the controller board determine the address of the wake-up I/O port.

3-7. WAKE-UP BLOCK

The Wake-Up Block is the first of the I/O communications blocks (see Figure 3-4). It is used to establish a link between the controller and the I/O communications blocks in host system memory.

3-8. CHANNEL CONTROL BLOCK

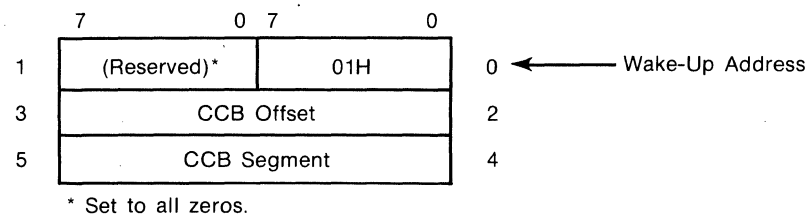
The controller uses the Channel Control Block to indicate the status of the internal processor (the Intel 8089 I/O Processor) and to invoke processor program operations. The Channel Control Block requires 16 bytes (see Figure 3-5). Except for the BUSY 1 FLAG (byte 1) and the Controller Invocation Block address (bytes 2 through 5), the information contained in this block is used to invoke controller operations that are transparent to the host.

3-9. CONTROLLER INVOCATION BLOCK

The controller uses the Controller Invocation Block (CIB) to post status to the host CPU and to locate the starting address for the controller's on-board disk interface program. The status sempahore byte (byte 3) has a special purpose. The host uses this byte to indicate to the controller whether it has read the current contents of the status byte and is ready for a status update. The Controller Invocation Block requires 16 bytes (see Figure 3-6).

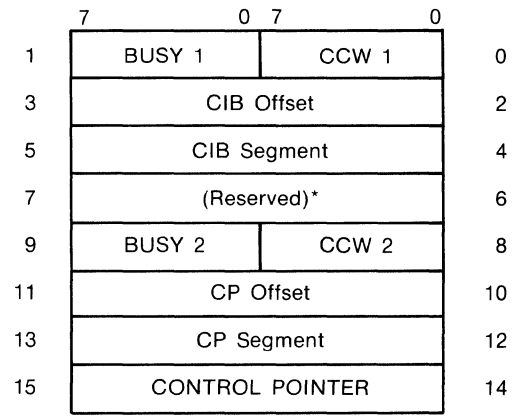
3-10. I/O PARAMETER BLOCK

The I/O Parameter Block (IOPB) contains the controller operating commands, which define the function the controller is to perform (read, write, etc.), and the parameters of the function (memory address, disk head and cylinder, etc.). The I/O Parameter Block requires 30 bytes of host memory space. Figure 3-7 describes the function of each byte.



Byte	Function
0	SYSTEM OPERATION COMMAND — Must be set to 01H.
1	Reserved.
2 through 5	CHANNEL CONTROL BLOCK (CCB) ADDRESS — Address of first byte of Channel Control Block. (Address = Offset + Segment $\times 2^4$).

Figure 3-4. Wake-Up Block



* Set to all zeros.

Byte	Function
0	CHANNEL CONTROL WORD 1 — Indicates location of Intel 8089 I/O Processor control store program. Must contain 01H.
1	BUSY 1 FLAG — Indicates whether controller is busy or idle. 00H — Idle FFH — Busy
2 through 5	CONTROLLER INVOCATION BLOCK (CIB) ADDRESS — Address of fifth byte of Controller Invocation Block.
6 and 7	Reserved.
8	CHANNEL CONTROL WORD 2 — Must contain 01H.
9	BUSY 2 WORD — Not meaningful to host CPU.
10 through 13	CONTROL POINTER ADDRESS — Address must point to the Control Pointer in the next sequential word.
14 and 15	CONTROL POINTER — Must be set to 0004H.

Figure 3-5. Channel Control Block

3-11. TYPICAL CONTROLLER OPERATIONS

The following section describes how to set up the I/O communications blocks in the host memory, how to initialize the controller and how to perform the various data transfer operations. It is assumed that the controller board has been properly installed as described in Chapter 2.

3-12. INITIALIZING THE CONTROLLER

The controller must be initialized before any data transfer activities between the host system memory and the disk drives can be initiated. Initialization of the controller involves:

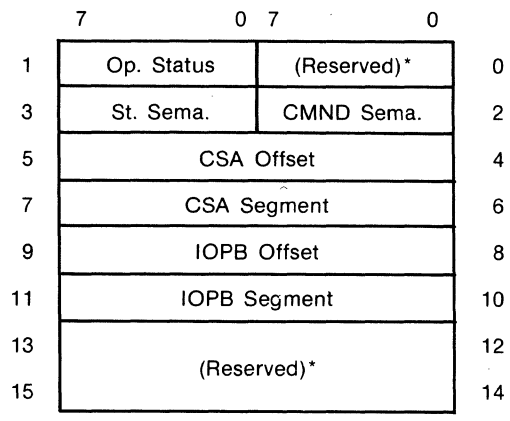
1. Establishing a link between the 8089 and the I/O communications blocks in host system memory.
2. Reading the parameters that describe the disk drives with which the controller is to interface into the controller's RAM buffer, using the Initialize function (FUNCTION = 00H).

This initialization must be performed following a:

1. Power-on event.
2. Controller reset (02H written to the wake-up I/O port).

After the controller has been initialized, any of the data transfer functions described in paragraphs 3-13 through 3-25 can be performed in any sequence. (Refer to paragraphs 4-12 through 4-15 for a detailed explanation of controller initialization.)

The following procedure gives the sequence in which the controller initializing activities must be performed. Prior to initializating the controller, check



* Set to all zeros.

Byte	Function
0	Reserved.
1	CONTROLLER OPERATION STATUS — Bits 0 through 2 indicate event completed. Bits 4 and 5 indicate drive associated with event. Bit 6 indicates error: soft (recoverable), 0, or hard, 1. Bit 7 indicates a summary error that can be checked through the transfer error status function (refer to paragraph 3-25).
2	COMMAND SEMPAHORE — Controller does not use this byte. It is provided for use as a multiprocessor interlock.
3	STATUS SEMPAHORE — Controller posts status only when this byte is 00H; when new status has been posted, controller sets byte to FFH. When host CPU has read status, it sets this byte to 00H.
4 through 7	CONTROL STORE PROGRAM ADDRESS — Starting address of controllers on-board disk interface program. Set to 00000H.
8 through 11	I/O PARAMETER BLOCK ADDRESS — Address of first byte of I/O parameter block.

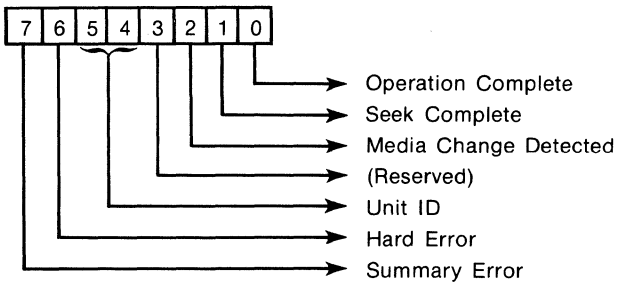


Figure 3-6. Controller Invocation Block

that the system data bus switch (S2-1), the host system I/O address switch (S2-2), the wake-up address switches (S1-1 through S1-8 and S2-3 through S2-10), and the interrupt level jumper has been set as described in the procedure titled Switch/Jumper Configurations in Chapter 2.

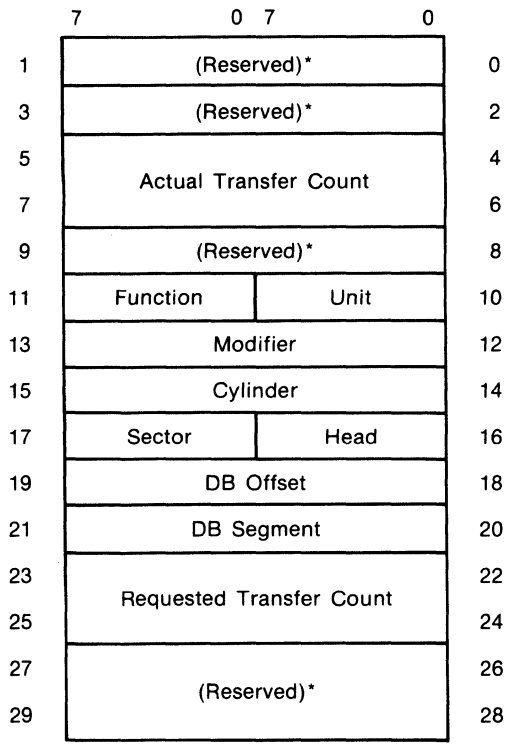
To initialize the controller, the host CPU must perform the following steps:

1. **Establish addresses for the four I/O communications blocks in host memory:**

Wake-Up Block	6 Bytes
Channel Control Block	16 Bytes
Controller Invocation Block	16 Bytes
I/O Parameter Block	30 Bytes

Remember that the address of the first byte of the Wake-Up Block must be equal to the wake-up address set in the controller's wake-up address switches times 2^4. For example, if the switches are set to 0673H, the address of byte 0 of the Wake-Up Block is:

06730H	20-Bit Addressing
6730H	16-Bit Addressing



* Set to all zeros.

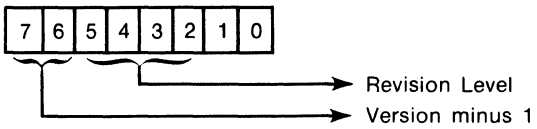
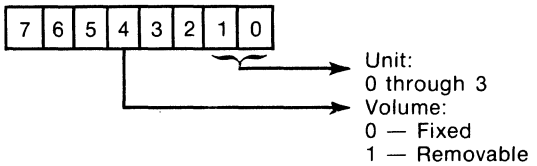
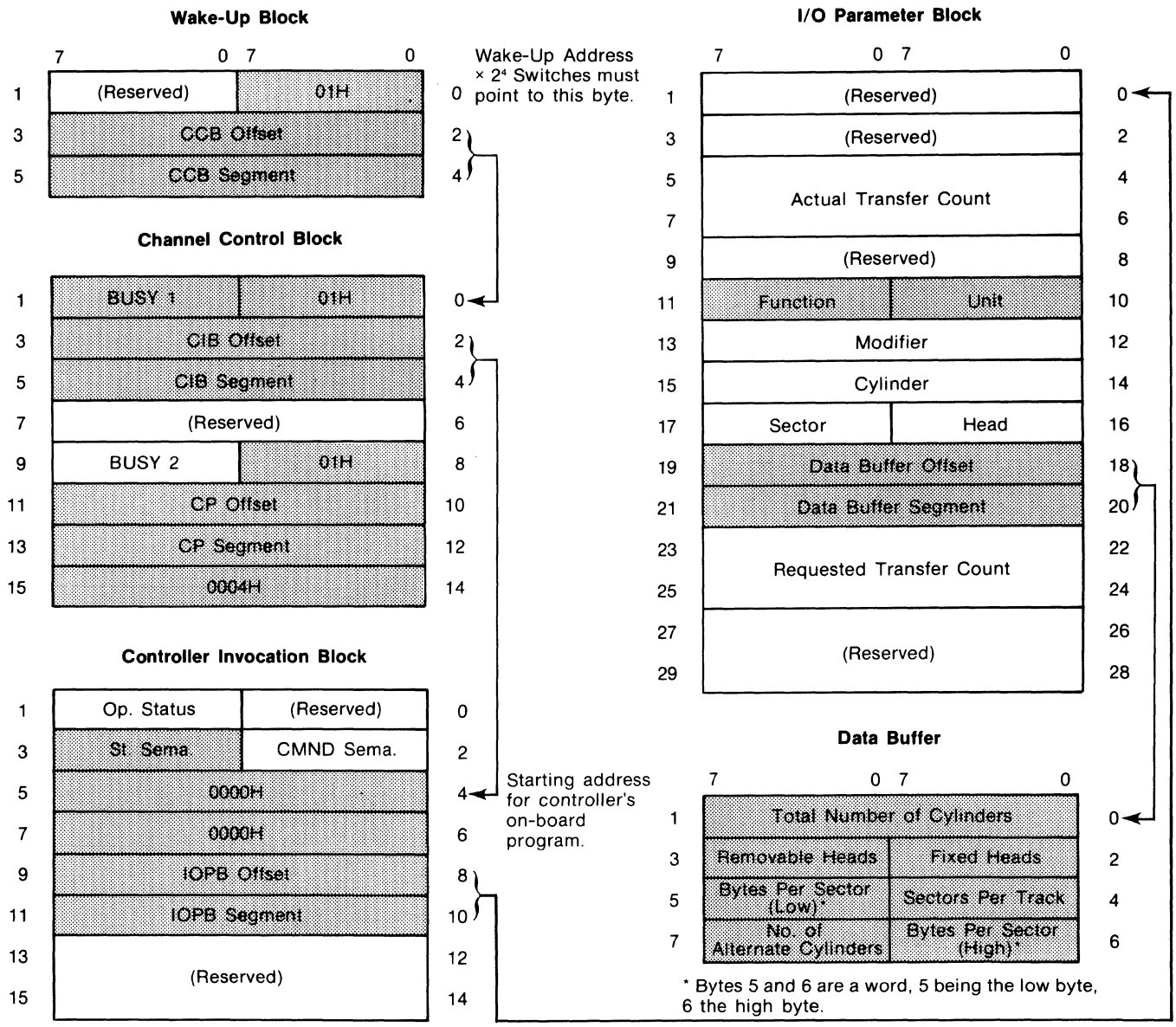
Byte	Name and Function
0 through 3	Reserved.
4 through 7	ACTUAL TRANSFER COUNT — Count of bytes actually transferred between the system and the disk or controller. Four-byte binary number, least significant bits in first byte. Controller writes count to IOPB following termination or completion of an operation. If count does not match requested transfer count, operation was prematurely terminated; check status. When performing the track formatting function, a count of 6 is set in the Actual Transfer Count word. When performing the status transfer function, the count is set to 12. When initializing drive 0, this word has a special function. It is used to display the controller firmware's version number as shown below: 
8 and 9	Reserved.
10	UNIT — Code for drive unit being accessed and its volume identification: bits 0 an 1 indicate unit ID codes; bit 4 indicates fixed or removable volume. If the drive unit being addressed uses either an all fixed or all removable volume, bit 4 must always be set to zero. 

Figure 3-7. I/O Parameter Block Description

Byte	Name and Function
11	FUNCTION — Code for operation to be performed. Refer to following discussion of typical controller operations for a detailed discussion of these operations: 00H INITIALIZE 01H TRANSFER STATUS 02H FORMAT 03H READ SECTOR ID 04H READ DATA 05H READ TO BUFFER AND VERIFY 06H WRITE DATA 07H WRITE BUFFER DATA 08H INITIATE TRACK SEEK 09H - 0DH Reserved 0EH BUFFER I/O 0FH DIAGNOSTIC
12 and 13	MODIFIER — Code to modify function codes. Bit 0 Suppresses interrupt on command completion when set to 1. Bit 1 Automatic retries for error recovery are inhibited when set to 1. Bits 2 through 15 Reserved.
14 and 15	CYLINDER — Binary number specifying logical cylinder code; bit 0 is least significant bit of number.
16	HEAD — Binary number specifying logical head code; bit 0 is least significant bit of number.
17	SECTOR — Binary number specifying logical sector code; bit 0 is least significant bit of number.
18 through 21	DATA BUFFER ADDRESS — Address of first byte in host system memory data (parameter) buffer.
22 through 25	REQUESTED TRANSFER COUNT — Count of bytes requested to be transferred between the system and the disk or controller. Four-byte binary number, least significant bits in first byte. See description of ACTUAL TRANSFER COUNT, bytes 4 through 7 in IOPB.
26 through 29	Reserved.

Figure 3-7. I/O Parameter Block Description (Continued)

2. **Set up the Wake-Up Block (see Figure 3-8).**
3. **Set BUSY 1 Flag (Optional).** Set the BUSY 1 flag (byte 1 of the Channel Control Block) to non-zero (FFH). This allows the host to monitor the BUSY 1 flag to find out when the initialization procedure is complete.
4. **Reset the controller.** Host writes a 02H to the wake-up I/O port.
5. **Clear the reset.** Host writes a 00H to the wake-up I/O port.
6. **Establish the host-controller communications link.** Write a 01H to the wake-up I/O port. The controller goes to the Wake-Up Block in host memory and records the address of the Channel Control Block, then goes to the Channel Control Block and clears the BUSY 1 FLAG. On all subsequent 01H commands to the wake-up I/O port, the controller will go to the Channel Control Block.
7. **Set up the Channel Control Block as shown in Figure 3-8.**
8. **Set up the Controller Invocation block as shown in Figure 3-8.** Be sure the STATUS SEMAPHORE, byte 3, is set to 00H.
9. **Set up the I/O Parameter Block as shown in Figure 3-8.** Be sure the UNIT, byte 10, is set for the correct unit number and the FUNCTION, byte 11, is set for the Initialize function (FUNCTION = 00H). Initialize unit 0 first.
10. **Establish parameter buffer.** Set up a disk drive parameter data buffer with the parameters for the drive to be initialized as shown in Figure 3-8. Be sure the data buffer address in the I/O Parameter Block points to the first address of this data buffer.
11. **Start initialize function.** Poll the BUSY 1 FLAG (Byte 1 of the CCB) and write a 01H to the wake-up I/O port when the flag is zero. The controller goes to the Channel Control Block, then links its way through the Controller Invocation Block and I/O Parameter Block and reads the disk drive parameters for the unit specified.
12. **Respond to and process the resulting interrupt or status or both.**
13. **Reset I/O Parameter Block.** Set the UNIT, byte 10, for the next unit to be initialized and set the data buffer address, byte 18 through 21, for the beginning address of the unit's disk parameters.
14. **Repeat steps 10 through 12 for each drive unit.** Note that the initialization procedure *MUST BE PERFORMED FOR ALL FOUR DRIVE UNITS*, starting with unit 0, even if one or more of the drives do not exist. Initialize all unattached drives with all zeros.



NOTE: Set up the shaded bytes in each of the I/O Communications Blocks and in the Data Buffer.

Figure 3-8. I/O Communications Blocks Linking

The controller is now initialized. This procedure need not be repeated except after a power-on or a controller reset. For all subsequent disk activities, the host communicates with the controller through the Channel Control Block, the Controller Invocation Block and the I/O Parameter Block.

3-13. TRACK FORMATTING

The Format Track function (FUNCTION = 02H) writes the gaps, sector headers and data fields (see Figure 3-2) on a track — one track per command. A track can be designated as a normal, assigned alternate or defective track. A defective track generally points to an assigned alternate track. Refer to the discussion of alternate and defective track handling in paragraph 3-14.

Use the following procedure to format a track.

1. **Set up the I/O Parameter Block as shown in Figure 3-9.**
2. **Set up a 6-byte data buffer for the type of track to be formated as shown in Figure 3-9.** A track can be designated as a data track, assigned alternate track or defective track. The user pattern is repeated throughout the data field of every sector. In the case of a defective track, the user pattern is a pointer to the alternate track. If the alternate track is defective, it can not be used to point to another alternate. An interleave factor of 1 corresponds to consecutive sectors.
3. **Initiate the format operation.** Write a 01H to the wake-up I/O port.
4. **Respond to and process the resulting interrupt or status or both.**

NOTE

Always format the last track on head 0 as a data track. This track should then be reserved for use by the on-board diagnostic.

3-14. ALTERNATE AND DEFECTIVE TRACK HANDLING

It is suggested that each disk surface be divided into two areas (see Figure 3-10), the data track area and the alternate track area. The user assigns the number of tracks in the alternate track area, typically 1 - 2% of the total number of available tracks on the surface. If a disk surface has 512 tracks, tracks 0 through 500 would constitute the data track area and tracks 501 through 510 would constitute the alternate track area. **The last track at Head 0 must be reserved for the diagnostic program.**

When a track within the data track area is deemed defective, the host reformats the track, giving it a defective track code and entering the address of the next available alternate track in the data fields. The alternate track that is selected must be formatted as an assigned alternate track.

When the controller accesses a track that has been previously marked defective, it will automatically invoke a seek to the assigned alternate track and use the alternate as if it were the data track area. This operation is automatic and is invisible to the user, except for the added time required to complete the operation.

3-15. DATA TRANSFER AND VERIFICATION

Seven data transfer and verification command functions are allowed, selected through the FUNCTION byte in the I/O Parameter Block: Read Sector ID, Read Data, Read Data to Buffer and Verify, Write Data, Write Data from Buffer, Initiate Track Seek and Buffer I/O.

NOTE

All data transfers between the host system memory and a disk drive unit are buffered through the controller's on-board RAM buffer. During a write, the controller performs a DMA transfer of a one-sector block of data from the host system memory to the RAM buffer. It then transfers the sector serially from the RAM buffer to the disk in two byte increments. When reading from the disk, the controller performs a serial transfer of a sector of data from the disk to the RAM buffer in two byte increments. When the entire sector has been read into the RAM and all error checking has been completed, the controller then performs a DMA transfer of the one-sector block from the RAM to host system memory.

The controller contains a burst error checking code (ECC) computing circuit that creates an error checking code for each sector ID and each data block written into disk memory. When reading data from the disk, the controller verifies the sector ID and the information in the data blocks using these error checking codes. If errors are detected that can be corrected (occur within an eleven-bit burst or less), they are corrected and the remainder of the operation is completed. If the error cannot be corrected, the sector is re-read. If after 27 retries the errors remain uncorrectable, the operation is terminated and a Hard Error is indicated in the operation status byte (byte 1) of the Controller Invocation Block. To obtain detailed information on the nature of the error,

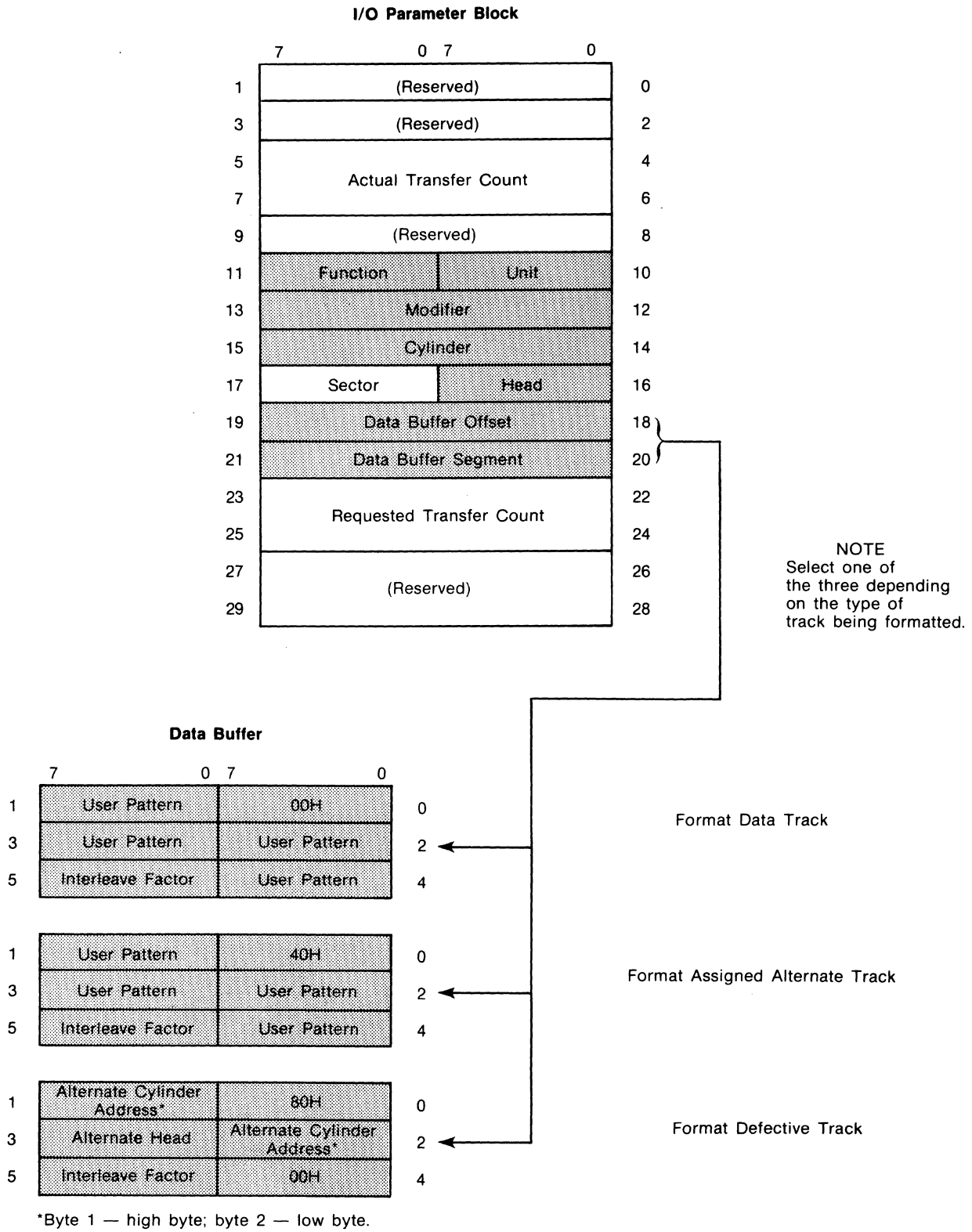


Figure 3-9. Track Formatting

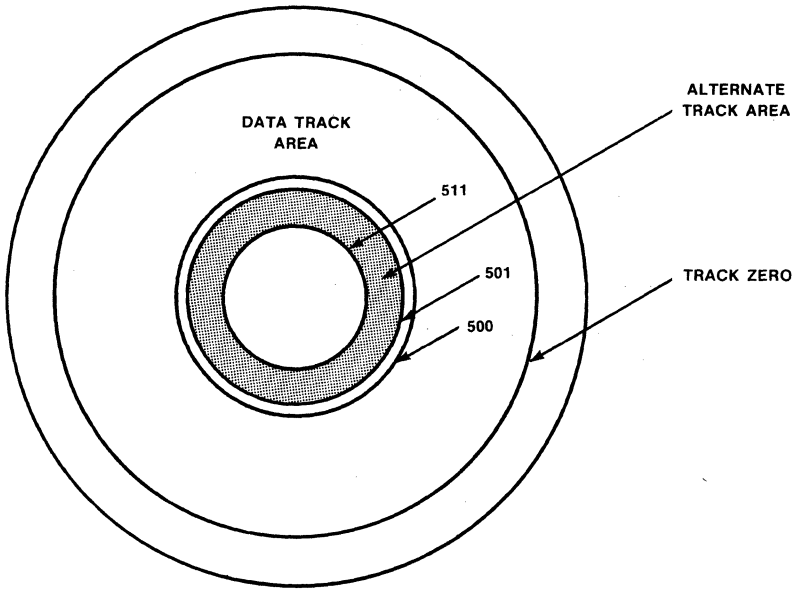


Figure 3-10. Alternate Track Formatting

perform the Transfer Error Status function (refer to paragraph 3-25).

Each of the data transfer and verification functions is described in detail in the following paragraphs. To use any one of these functions, the host CPU must perform the following steps:

1. **Set up the I/O parameter block as shown in the paragraph describing the function.**
2. **Initiate the operation.** Write a 01H to the wake-up I/O port.
3. **Respond to and process the resulting interrupt or status or both.**

3-16. READ SECTOR ID

The Read Sector ID function (FUNCTION = 03H) searches for the first error free sector on the selected track and writes the contents of the sector ID field into a 5-byte data buffer in host memory (see Figure 3-11). An implied seek, head select or volume change, *is not performed.* The Read Sector ID is performed on the cylinder, volume and head that the previous function selected. One use of this function is to search the alternate track area for tracks that have not been assigned as alternates.

To perform this function, set up the shaded bytes in the I/O parameter block as shown in Figure 3-11, and reserve a 5-byte data buffer in host system memory.

3-17. READ DATA

The Read Data function (FUNCTION = 04H) reads data from the disk into host system memory. It begins reading with the first byte of the selected sector and ends reading when the requested byte count is reached, end of media is reached or a hard failure is detected. If multi-sector data transfers are requested the controller automatically seeks to the next sector, the next head and the next cylinder, in that order. Automatic head increments are supported only within the volume, fixed or removable, but not between volumes, for example, fixed across to removable. The last sector, head and track address in the data track area defines the end of media. An implied seek is invoked if the current head position is different from the specified track identification. The DATA BUFFER address set in the I/O parameter block is the address in host system memory where the first data byte read from the disk is to be transferred. Since the data being transmitted from the disk drive is buffered in the controller's RAM, data overruns cannot occur. To perform this function, set up the shaded bytes in the I/O parameter block as shown in Figure 3-12.

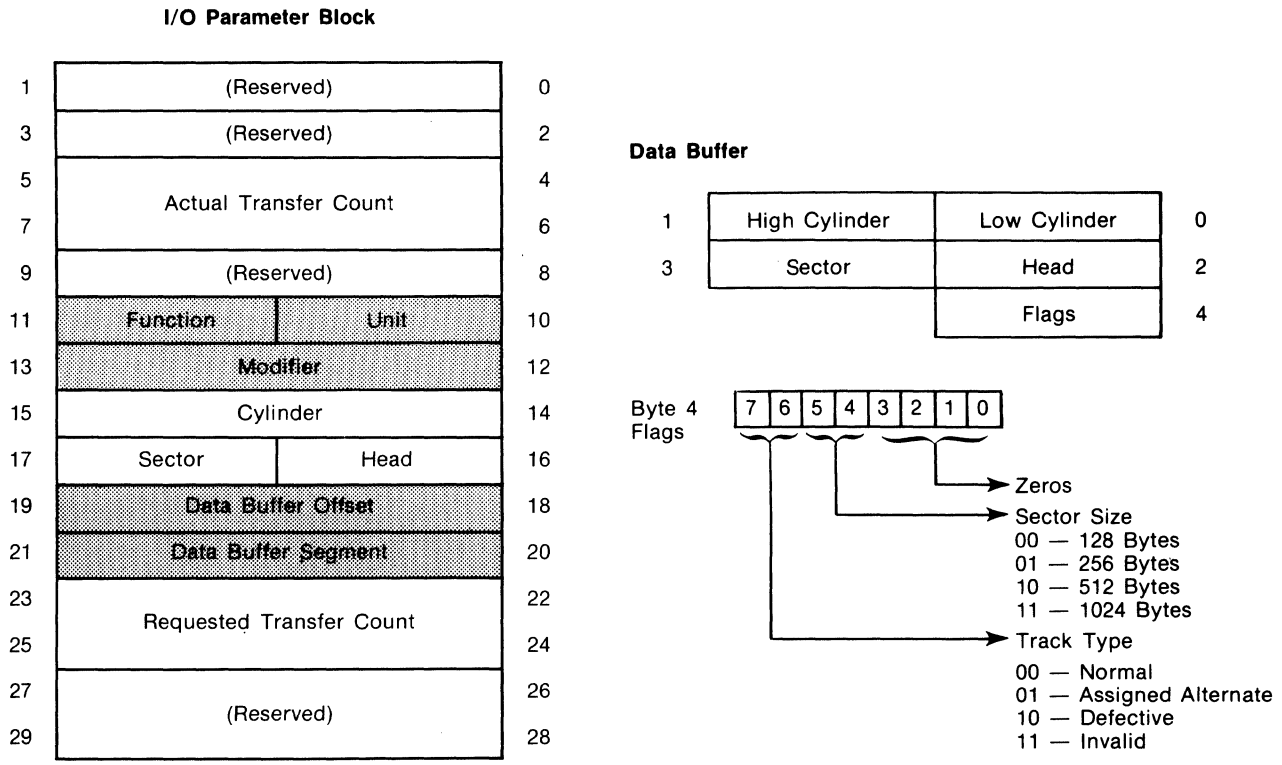


Figure 3-11. Read Sector ID

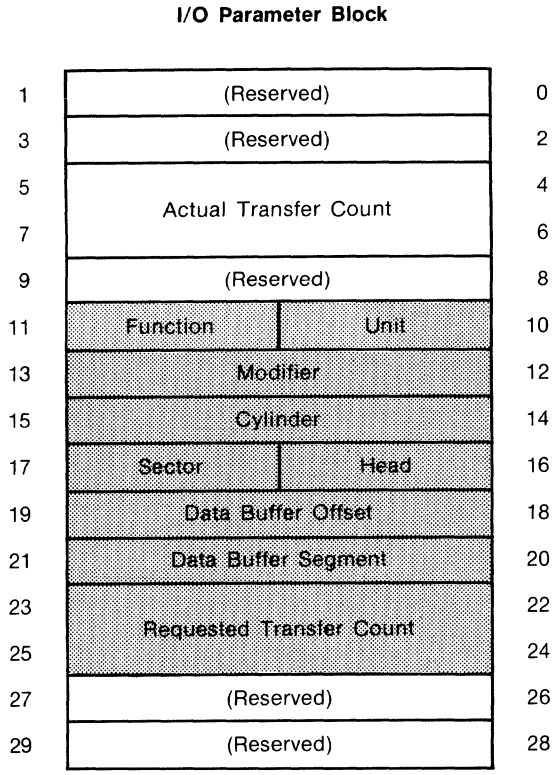


Figure 3-12. Read Data

3-18. READ DATA INTO CONTROLLER BUFFER AND VERIFY

The Read Data into Controller Buffer and Verify function (FUNCTION = 05H) reads data from the disk into the controller on-board RAM and checks the ECCs to verify the sector ID and data fields for all sectors affected. It begins reading with the first byte of the selected sector and ends reading when the requested byte count is reached, end of media is reached or a hard failure is detected. The multi-sector data verification is supported through the auto-sector, auto-head, auto-cylinder protocol described for Read Data function. End of media and implied seek are also supported as described for the Read Data functions.

The Read Data into Controller Buffer and Verify function has two applications:

1. Allows data to be verified after it has been written from host system memory to the disk.

2. Allows data to be transferred from one disk location to another by coupling this function with the Write Data from Controller Buffer function.

To perform the Read Data into Controller Buffer and Verify function, set up the shaded bytes in the I/O parameter block as shown in Figure 3-13.

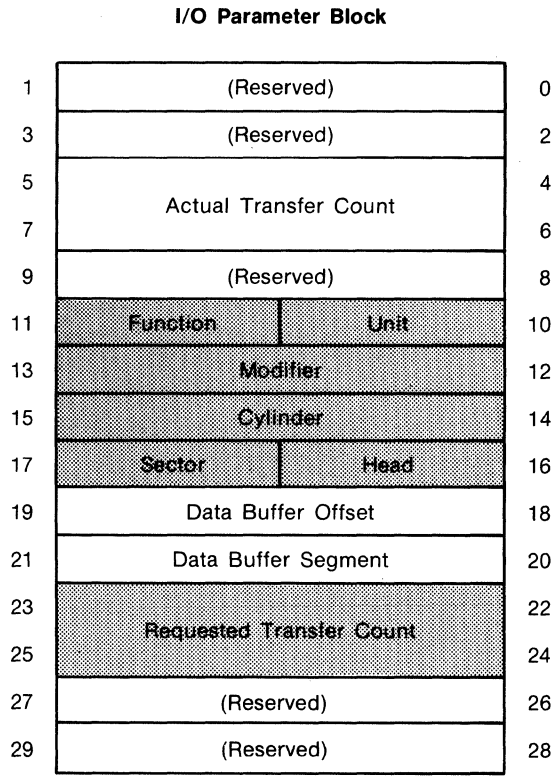


Figure 3-13. Read Data into Controller Buffer and Verify

3-19. WRITE DATA

The Write Data function (FUNCTION = 06H) writes data from host system memory onto the disk. It begins reading from the specified host data buffer address and writes to the first byte of the selected sector. It ends writing when the requested byte count is reached, end of media occurs (system memory or disk space) or a hard failure is detected. When writing to more than one sector, the sector selection is automatic as described for the Read Data function. Auto-head increments and implied seek are also supported as described for the Read Data function. If writing ends in the midst of a sector, the remaining area of the sector is filled with zeros.

To perform this function, set up the shaded bytes in the I/O parameter block as shown in Figure 3-14.

3-20. WRITE DATA FROM CONTROLLER BUFFER TO DISK

The Write Data from Controller Buffer to Disk (FUNCTION = 07H) writes data from the controller on-board RAM onto the disk. It begins reading from the first address of the controller's data buffer (4010H) and writes to the first byte of the selected disk sector. It ends writing when the requested byte count is reached, end of media occurs (controller memory or disk space) or a hard failure is detected. When writing to more than one sector, the sector selection is automatic as described for the Read Data function and the data in the buffer is repeated for each sector written. Auto-head increments, implied seek and end of media are also supported as is described for the Read Data function. If writing ends in the midst of a sector, the remaining area of the sector is filled with zeros.

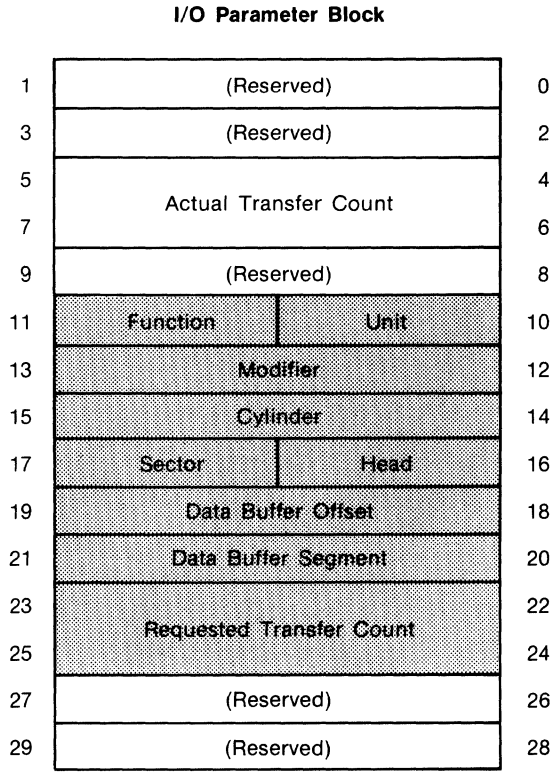


Figure 3-14. Write Data

To perform this function, set up the shaded bytes in the I/O parameter block as shown in Figure 3-15.

3-21. INITIATE TRACK SEEK

The Initiate Track Seek function (FUNCTION = 08H) positions the read/write head on a specified track, if the head is not already on that track. When issued sequentially to several drives, this command allows multiple disk drives to perform concurrent (overlapping) seeks. If a seek to a cylinder beyond the end of media (which includes alternates) is initiated, the heads are automatically returned to track zero, and an invalid address error is posted. If an operation complete interrupt is enabled, it is invoked when the seek command has been initiated and a seek complete interrupt (which is always enabled) is invoked when the seek is completed. The operation complete interrupt allows a function to be initiated on a second drive while the seek is being performed on the first drive.

To perform this function, set up the shaded bytes in the I/O parameter block as shown in Figure 3-16.

3-22. BUFFER I/O

The Buffer I/O function (FUNCTION = 0EH) transfers data between the host system memory and controller on-board RAM. Beginning addresses in the host system memory and controller buffer memory are specified. Data transfer begins at these addresses and ends when the requested byte count is reached. Since the controller has only 64K bytes of local memory address space, the most significant bytes of the REQUESTED TRANSFER COUNT (bytes 24 and 25) are ignored.

CAUTION

Data transfers from the host system memory to the controller-buffer must be written to addresses within the range of 4000H to 4600H.

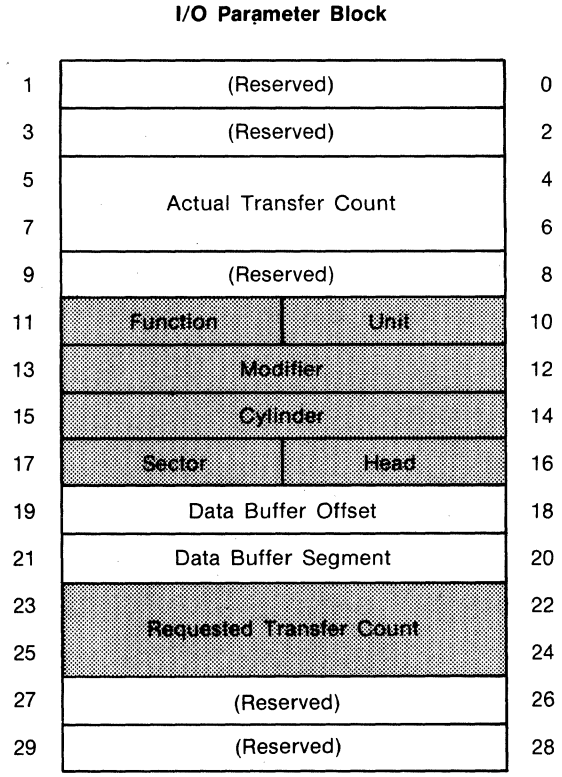


Figure 3-15. Write Data from Controller Buffer to Disk

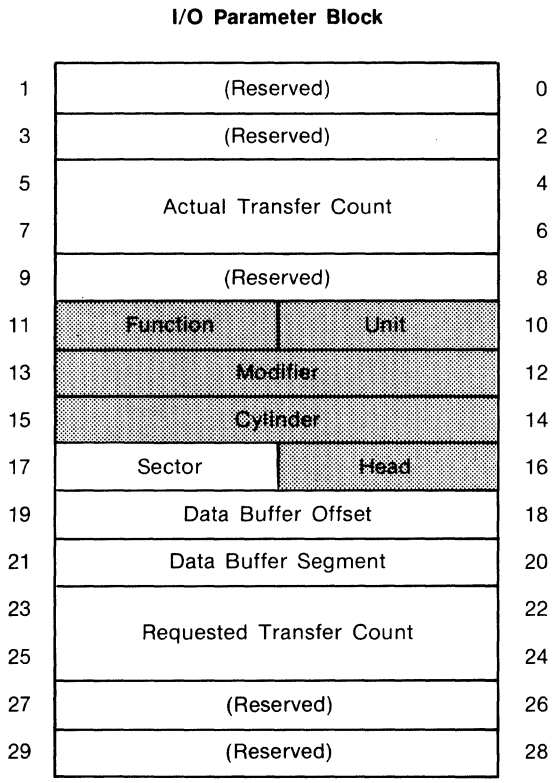


Figure 3-16. Initiate Track Seek

The beginning address in controller memory and the direction of data transfer are specified in the CYLINDER and HEAD fields, respectively:

Bytes 14 and 15 Starting controller memory address:
- Byte 15 — High Byte
- Byte 14 — Low Byte

Byte 16 Direction of data transfer:
- 00H — From controller to host
- FFH — From host to controller

The Buffer I/O function has two applications. Its primary purpose is for use with the diagnostic program. It also allows memory-to-memory transfers with a minimum of host overhead.

To perform this function, set up the shaded bytes in the I/O parameter block as shown in Figure 3-17.

3-23. DIAGNOSTIC

The Diagnostic function (FUNCTION = 0FH) causes the controller to perform a go/no-go self-diagnostic test that verifies internal data and status electronics and checks position and read/write electronics in the disk units. The diagnostic test program is contained in the controller's on-board PROM.

The diagnostic track is always located on a drive unit's last (highest number) track of head 0. When allocating memory space for the disk unit, this track must be dedicated to the diagnostic program. When initiating the diagnostic program, the head and cylinder are selected automatically, the user selects the drive unit. The diagnostic test is divided into three parts. The upper byte of the MODIFIER field (byte 13) determines the part of the diagnostic test that is executed:

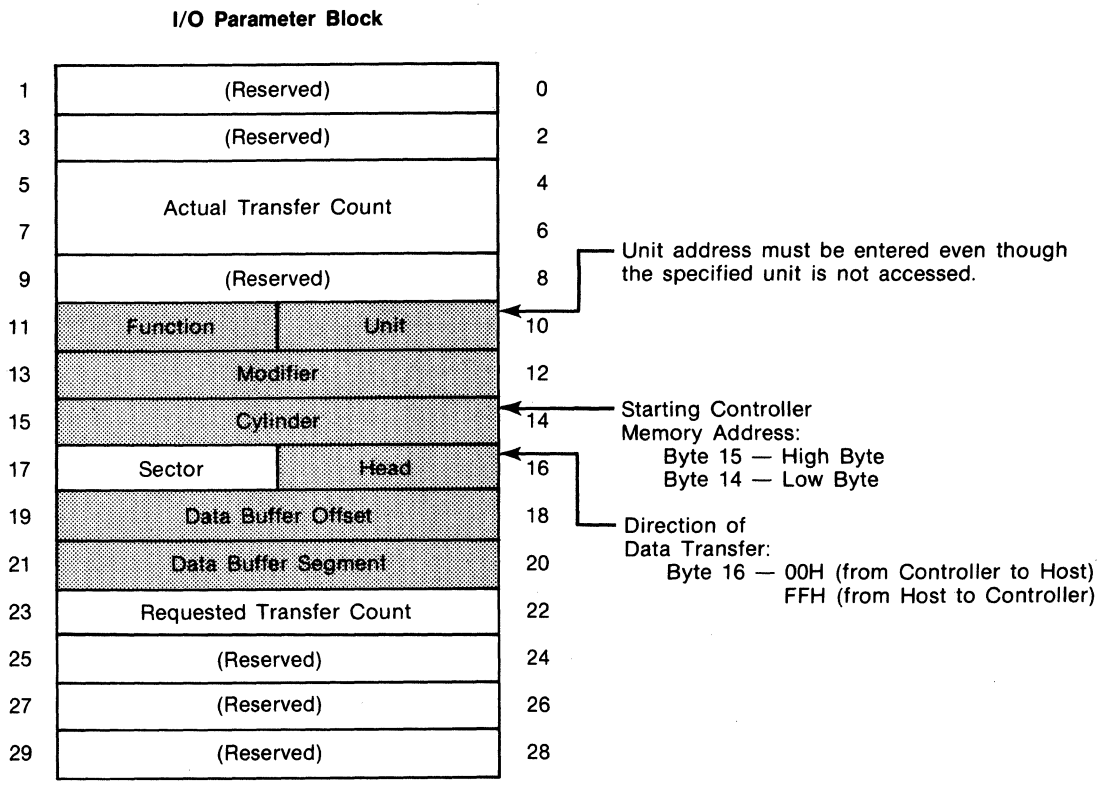


Figure 3-17. Buffer I/O

Byte 13	Function Executed
00H	Controller seeks the designated diagnostic track, performs a read ID and verifies the track position. It then writes and reads sector 0 with a 55AAH data pattern and verifies that the data read matches the data written.
01H	Controller performs a ROM checksum test to verify the contents of ROM.
02H or greater	Controller recalibrates the drive.

Any errors in the reading or writing are posted in the error status registers.

To perform this function, set up the shaded bytes in the I/O parameter block as shown in Figure 3-18.

3-24. POSTING STATUS

When the controller has completed an operation (read data, seek track, etc.), it posts the operation status in byte 1, the OPERATION STATUS byte, of the controller invocation block, using the following procedure:

1. The controller checks the STATUS SEMAPHORE byte (byte 3 of the controller invocation block) for 00H.
2. If the STATUS SEMAPHORE byte is non-zero, it indicates that the host CPU has not checked the OPERATION STATUS byte for the last status posted. When the host CPU does check the operation status, it sets the STATUS SEMAPHORE byte to 00H and clears the interrupt.
3. When the controller reads 00H in the STATUS SEMAPHORE byte, it posts the current status in the OPERATING STATUS byte, sets the STATUS SEMAPHORE byte back to non-zero and sets the interrupt.

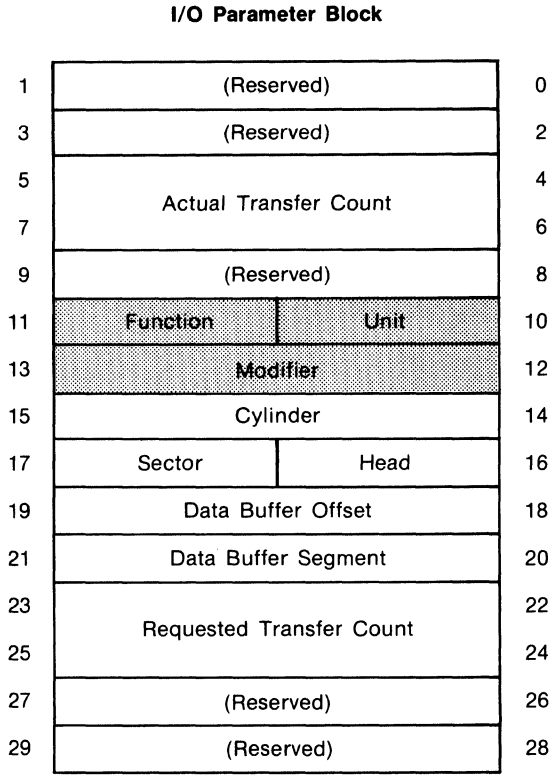


Figure 3-18. Diagnostic

4. The host CPU in turn, either polls the STATUS SEMAPHORE byte periodically for a non-zero or is interrupted, indicating that new status is present.

The status posted includes: operation complete, seek complete, media change detected and errors detected. If an error was detected, the unit on which the error occurred and an indication of whether the error was a hard error or a summary error is posted (see Figure 3-6). A more detailed description of the error is recorded in the error status buffer in the controller memory. To examine this error status the user transfers the information in the error status buffer from the controller to host system memory using the transfer error status function (FUNCTION = 01H) described in the following paragraph.

It should be noted that error status information is not cumulative. The error status buffers are cleared at the beginning of each new command operation, except the Transfer Error Status Command.

3-25. TRANSFER ERROR STATUS

The Transfer Error Status function (FUNCTION = 01H) transfers error status from the 12-byte error status buffer in the controller memory to a data buffer in the host system memory. The user can then examine the status bits to determine the cause of the error. Table 3-2 shows the information stored in each byte of the error status buffer. Table 3-3 describes which kind of errors are indicated by the setting of the hard (unretrievable) error and soft (retrievable) error bits in bytes 0 through 2. To perform the Transfer Error Status function, set up the shaded bytes in the I/O parameter block as shown in Figure 3-19.

Table 3-2. Error Status Buffer

	Error Status Buffer
Byte	**Function**
0 and 1	HARD ERROR STATUS — See Table 3-3
2	SOFT ERROR STATUS — See Table 3-3
3 and 4	DESIRED CYLINDER
5	DESIRED HEAD
6	DESIRED SECTOR
7 and 8	ACTUAL CYLINDER & FLAGS
9	ACTUAL HEAD & VOLUME
10	ACTUAL SECTOR
11	NUMBER OF RETRIES ATTEMPTED

3-26. INTERRUPTS

The controller normally posts interrupts to the host on three conditions:

1. Command complete
2. Seek complete
3. Media change (change disk pack)

The interrupt on command complete can be disabled by entering a one in bit 0 of the Modifier word in the I/O parameter block (bytes 12 and 13). The seek complete and media change interrupts can not be disabled. To clear an interrupt, the host writes a 00H to the Wake-Up I/O port.

Pins on the controller board allow the interrupt priority level of the controller to be set to from 0 to 7. Refer to the discussion of interrupt priority level selection in Chapter 2.

3-27. EXAMPLE CONTROLLER I/O PROGRAM

Appendix A provides an example of a host processor program to initiate data transfers between the host system memory and disk drives through the iSBC 220 controller.

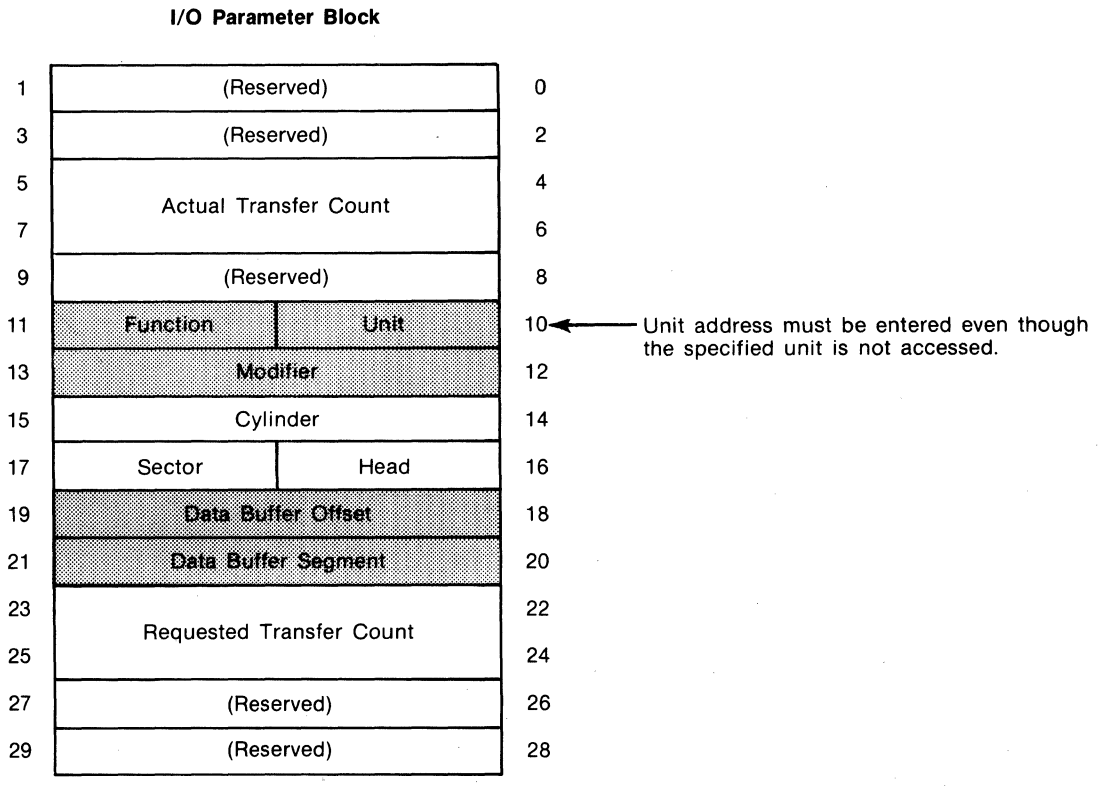


Figure 3-19. Transfer Error Status

Table 3-3. Bit Functions in Hard and Soft Error Bytes

Byte	Bit	Function
0	0 through 2	Reserved for future use.
	3	RAM ERROR — Controller RAM error was detected.
	4	ROM ERROR — Controller ROM error was detected.
	5	SEEK IN PROGRESS — Indicates a seek was already in progress for a unit when another seek was requested.
	6	ILLEGAL FORMAT TYPE — Both alternate track and defective alternate track flag set indicating an attempt to create an alternate track for a defective alternate track, which is not allowed, or an attempt to access an unassigned alternate track.
	7	END OF MEDIA — End of media was encountered before requested transfer count expired.
1	8	ILLEGAL SECTOR SIZE — Sector size read from the sector ID field conflicts with sector size information that controller specified in initialization command.
	9	DIAGNOSTIC FAULT — Micro-diagnostic fault detected.
	A	NO INDEX — Controller did not detect index pulse.
	B	INVALID COMMAND — Invalid function code detected.
	C	SECTOR NOT FOUND — Desired sector could not be located on selected track.
	D	INVALID ADDRESS — Invalid address was requested.
	E	SELECTED UNIT NOT READY — Selected unit is not ready, not connected, or not responding to unit connect request.
	F	WRITE PROTECTION FAULT — An attempt has been made to write to a write protected unit.
2	0 through 2	Reserved for future use.
	3	DATA FIELD ECC ERROR — Error has been detected in the data field of a sector. If bit 6 in Controller-Invocation status byte (byte 1) is set, error is hard and uncorrectable. If bit 6 is not set, error is soft and correctable.
	4	ID FIELD ECC ERROR — Error has been detected in the ID field of a sector. If bit 6 in Controller-Invocation status byte (byte 1) is set, error is hard and uncorrectable. If bit 6 is not set, error is soft and correctable.
	5	DRIVE FAULT — Hardware fault detected in selected drive unit. Fault characterized by: read/write fault, positioner fault, power fault or speed fault.
	6	CYLINDER ADDRESS MISCOMPARE — ID field contains a cylinder address different from the expected cylinder address.
	7	SEEK ERROR — Hardware seek error was detected.

CHAPTER 4
PRINCIPLES OF OPERATION

4-1. INTRODUCTION

This chapter provides a functional description of the iSBC 220 SMD Disk Controller circuit operation. The discussion assumes that the reader has a working knowledge of digital electronics and has access to the individual component description of each integrated circuit used on the board. As a prerequisite, the reader should be familiar with the programming conventions discussed in Chapter 3 of this manual, and the functional operation of the Intel 8089 I/O processor and the Multibus interface. Familiarity with the disk drive's operation and interface specifications will also prove beneficial in understanding the controller operation.

4-2. SCHEMATIC INTERPRETATION

A set of schematic diagrams for the controller board (figure 5-2) and a component location diagram (figure 5-1) are included in Chapter 5 of this manual.

The schematics are drawn to standard drafting conventions with input signals entering from the left and output signals exiting to the right. Input and output signals between individual sheets of a schematic include a location coordinate code immediately preceeding (input signals) or following (output signals) the signal name. This code defines the location of the origin or destination of the signal within the schematic diagrams. The first digit of the code is the schematic sheet number, and the last two characters specify the zone defined by the horizontal and vertical grid coordinates, which are printed around the perimeter of each schematic sheet. For example, the code "7B8" indicates that the origin or destination of the associated signal appears on sheet 7 of the schematic set within the zone defined by grid coordinates "B" and "8".

An "X" for one of the grid coordinates indicates an entire vertical column or horizontal row on the schematic sheet. For example, the code "7BX" indicates the entire "B" zone on sheet 7.

The logic symbols used in this manual are drawn as specified in ANSI Standards 14.15 and Y32.14. Standard definitions are used for symbols and active line levels on inputs and outputs (see figure 4-1). A small circle on the input of a logic element indicates that a relative low level is needed to activate the element. The absence of a circle indicates that a relative high level is needed to activate the element. Output levels are indicated in the same manner.

Logic gating symbols are drawn according to their circuit function rather than the manufacturer's definition. For example, the gates shown in figure 4-1 can be drawn in one of the two configurations shown, depending on their circuit application.

In addition to the inversion symbol convention, signal nomenclature also follows an active state convention. When a signal (or level) is active in its low state, the signal name is followed by a virgule or "slash" (e.g., XACK/); when a signal is active in its high state, the slash is omitted from the signal name, (e.g., XACK). This convention corresponds to putting a bar over a signal name to indicate it is active in its low state (e.g., XACK).

4-3. FUNCTIONAL OVERVIEW

General. The function of the iSBC 220 SMD Disk Controller board is to allow the host system to access any location on a specific disk of a selected disk drive and either:

1. Transfer data to that disk location from system (host) memory (write operation), or
2. Transfer data from that disk location to system memory (read operation).

To accomplish this task, the controller circuitry is divided into two sections (see figure 4-2):

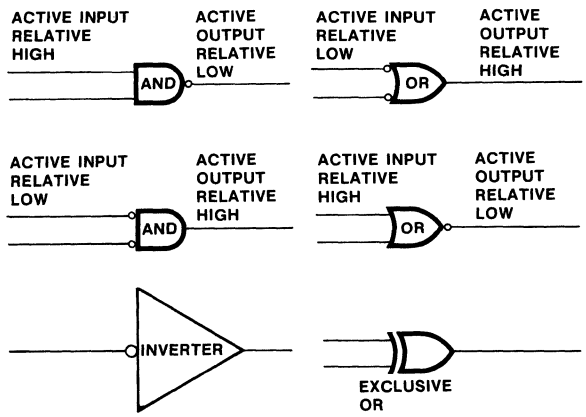


Figure 4-1. Logic conventions.

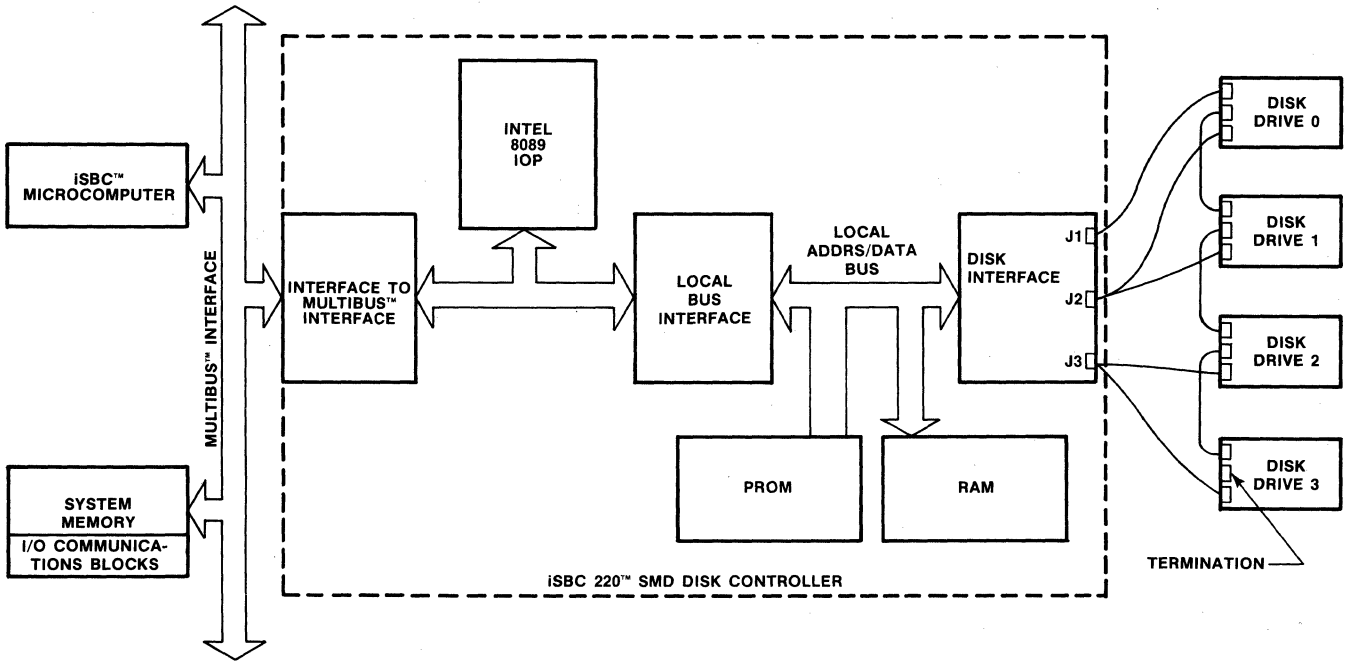


Figure 4-2. Simplified block diagram of iSBC 220 controller.

1. Logic that controls communications and data transfer between the host processor and the controller through the Multibus interface, and
2. Logic that controls data transfer between the controller and the disk drive(s) through the SMD interface.

The Intel 8089 I/O processor (IOP) controls the data transfer process, using a program stored in on-board ROM. It receives instructions from the host processor through four I/O communications blocks in system memory. Once the host instructs the controller to begin a data transfer, the 8089's internal processor makes a DMA transfer to or from system memory, independent of the host processor.

2K bytes of RAM are included on the board for intermediate storage of data and to allow on-board error checking. This data buffer allows DMA transfer to be made between the controller and host system memory, which minimizes Multibus overhead and eliminates disk drive overruns.

Communicating with the host. Figure 4-3 provides a detailed block diagram of the controller. The Bus Arbiter and the Bus Controller manage the transfer of data between system memory and controller through the Multibus interface. The Bus Arbiter negotiates with the current bus master for control of the Multibus interface. The Bus Controller generates control signals that gate data transfers between system memory and the on-board RAM. It also controls the transfer of data from RAM to the disk communication circuitry.

The Multibus interface Address Latches transmit 20-bit addresses to system memory via the Multibus interface. The Multibus interface Data Transceiver transmits data either to or from system memory via the Multibus Interface. The controller data bus is 16-bits. The Data Transceiver uses a byte-swap technique to allow data transfer with either an 8-bit or 16 bit system memory.

The Wake-Up Address Comparator is used to assign the controller a host system I/O port address and to set up a communications link between the 8089 IOP and the I/O communications blocks in system memory. (A detailed discussion of the controller initialization procedure is given in Chapter 3 and in paragraphs 4-12 through 4-15 in this section.)

Communicating with the disk. The 8089 IOP treats the ROM, RAM and disk communications side of the controller circuitry as local memory. The Local Address Latches transmit 16-bit addresses to local memory. The Local Data Transceiver transmits data either to or from local memory. Some of the addresses in local memory provide access to local I/O ports (see paragraph 4-20 for a detailed discussion of local I/O ports). The Address Decoder decodes these addresses and generates chip select or enable

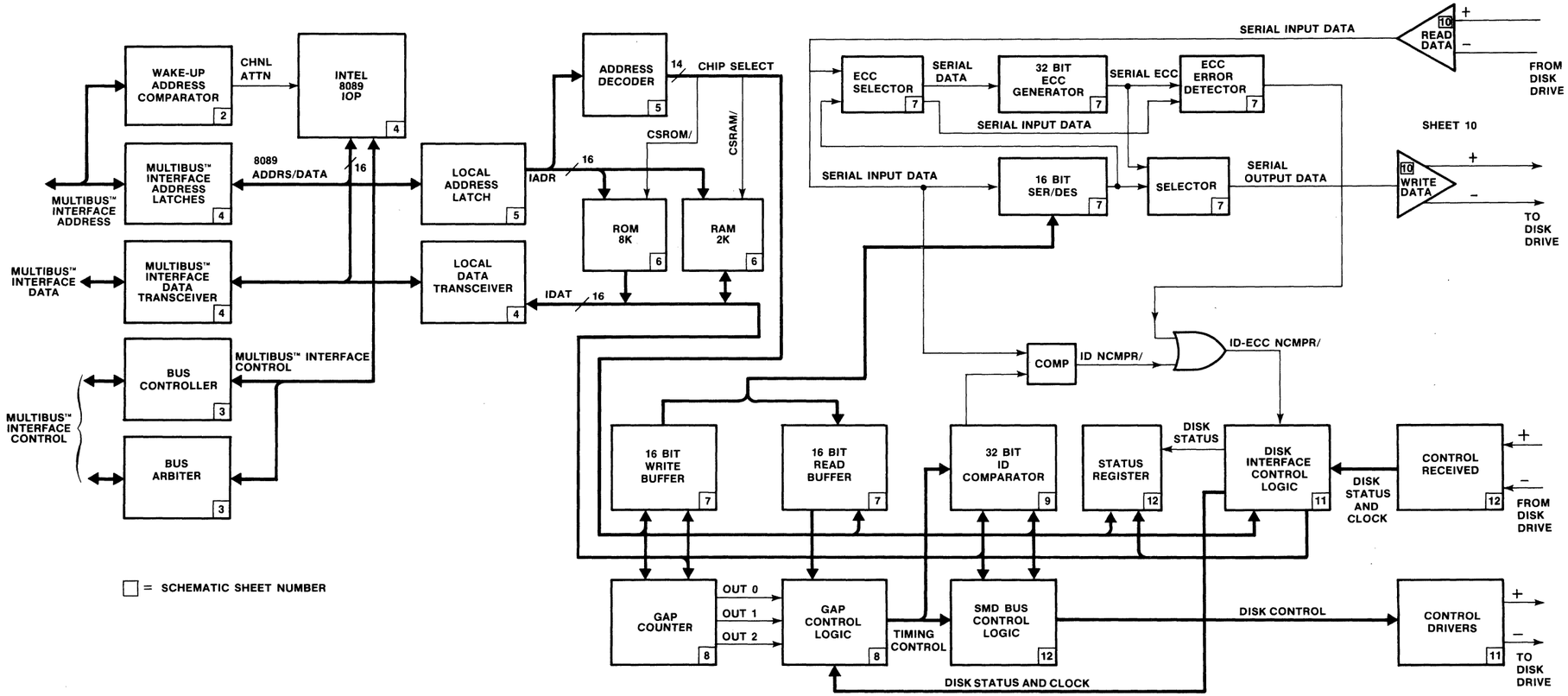


Figure 4-3. iSBC 220™ Controller Functional Block Diagram

signals that control the transfer of data to and from the disk. For example, the address 8028H enables the 16-Bit Write Buffer to receive a data word from the local memory. The ROM and RAM are also assigned specific ranges of addresses in local memory.

The 16-Bit SER/DES (Serializer/Deserializer) performs the serial-to-parallel and parallel-to-serial conversion required to transfer data between the disk and system memory. The 16-Bit Write Buffer and the 16-Bit Read Buffer provide intermediate storage for a single 16-bit parallel word between the RAM and the SER/DES. On a write operation, a 16-bit word is transferred from RAM to the write buffer. The SER/DES then converts the word from parallel to serial and transmits it to the disk through the write data driver. On a read operation, a 16-bit serial word is transmitted from the disk through the Read Data Receivers to the SER/DES. The SER/DES then performs a serial-to-parallel conversion and stores the resulting parallel word in the read buffer. The Write Data Driver and the Read Data Receivers are designed to generate and read SMD standard drive signals.

The 32-Bit ID Comparator determines when the selected sector on the disk is found during the search for sector ID operation that precedes a write or read function. When a write or read is initiated, the 32-bit sector identification (cylinder, head and sector number) is loaded in the 32-Bit ID Comparator. Sector IDs from the disk are then read and compared with the selected sector ID. When the selected sector is found, data transfer is initiated.

The 32-Bit ECC Generator creates an error checking code (ECC) that is appended to the end of each sector ID field and to each data field (see figure 3-1). This ECC is used for error checking and correction of data errors. It allows all the errors in a burst of up to 11 bits to be corrected.

The Gap Control Logic controls the spacing of data within a sector. Three programmable Counters, which count disk clock pulses, provide timing for the Gap Control Logic. The ability to program the Counters allows the disk(s) to be formatted for a number of different record sizes.

The SMD Bus Control Logic transmits disk control information to the disk drive units through the Control Line Drivers. The Input Control Logic receives status information from the disk drive units and controls the sequencing of the controller read and write operations.

A more detailed overview of the read and write operations is given in paragraph 4-29 through 4-33.

4-4. DETAILED FUNCTIONAL DESCRIPTION

The detailed functional description of the iSBC 220 SMD Disk Controller circuitry is divided into two major sections: Controller to Host Communications and Controller to Disk Communications. Within each of these sections, the following subjects are discussed:

Controller to Host Communications:

- Multibus Interface
- 8089 IOP
- Bus Arbiter
- Bus Controller
- Multibus Interface Data Transfer Logic
- Controller Initialization
- Wake-Up Address Comparator
- Controller Reset and Clear
- Establishing a Link with I/O Communications Blocks
- Interrupt Priority
- Memory Map
- ROM
- RAM
- I/O Port Decode Logic

Controller to Disk Communications

- Controller to Disk Drive Interface
- DMA Mode
- Disk Formatting
- Write Data Transfer
- Read Data Transfer
- SER/DES Logic
- Sync Byte Comparator Logic
- 32-Bit ID Comparator Logic
- ECC Generator Logic
- Status Register Logic
- Line Drivers and Receivers

4-5. CONTROLLER TO HOST COMMUNICATIONS

The following discussion provides a detailed functional description of the section of the iSBC 220 SMD Disk Controller that communicates with the host through the Multibus interface.

4-6. MULTIBUS INTERFACE

The 8089 IOP communicates with the host processor and the system memory through the Multibus interface. The Multibus interface signal description and pin configurations are explained in Chapter 2. A detailed description of the Multibus interface operation can be found in the *Intel Multibus Specification* Intel Order Number 9800683.

4-7. 8089 I/O PROCESSOR (IOP)

The 8089 IOP, U79 (4X4), is a microprocessor device that has been designed specifically to perform high speed I/O transfers of data between system memory and mass storage devices such as disk drives. Its ability to perform DMA data transfers independent of the host processor allows it to carry out most system memory-to-disk transfers of data simultaneously with other host processor operations. Refer to *The 8086 Family User's Manual,* Intel Order Number 9800727 for a detailed explanation of the 8089 and supporting IC devices.

A number of 8089 control lines have important functions in the controller design. The RESET line (4D4), when pulled high, resets the 8089 to the beginning of its internal firmware control program. Channel Attention line CA (4C4) allows the host to gain the attention of the 8089. On the first channel attention following a reset, the 8089 fetches the contents of address FFFF6H and begins an internal initialization procedure. On subsequent channel attentions, the 8089 looks to the I/O communications blocks in system memory for further instructions. Refer to paragraphs 4-12 through 4-15 for a detailed discussion of the controller initialization procedure and the use of the CA line.

The Bus Interface Unit (BIU) in the 8089 controls the controller local data bus cycles, transferring instructions and data between the 8089 IOP and external memory or the disk. Every bus access is associated with a register tag bit that indicates to the BIU whether the host system memory or local memory is to be addressed. The BIU outputs the type of bus cycle on status lines S0/, S1/ and S2/. The 8288 Bus Controller decodes these lines and provides signals that selectively enable one bus or the other.

The 8089 is a 16-bit processor, but it is capable of making both single-byte fetches (8-bit system memory) or two-byte fetches (16-bit system memory). The address zero line, IADR-0 (5C1), controls the byte swapping facility of the controller when communicating with an 8-bit system memory.

4-8. CLOCK CIRCUIT

The clock circuit consists of U59, an 8284A Clock/ Driver (4C5), and a 15 MHz crystal. The 8284A divides the crystal output by three to produce the 5 MHz CLK necessary to drive the 8089 IOP. The 8284A produces a reset signal (RST), which is used on power-up to reset the 8089, Interrupt Latch U60 (3B6) and the Read/Write Control logic. In addition to the reset signal, the 8284A also produces a synchronized ready input to the 8089. A high on the READY line received from the addressed device (XACK/ from external memory, RDY | TIME OUT from the on-board read/write port), indicates that the memory or read/write port has accepted data during a write operation or data is ready to be read during a read operation.

4-9. BUS ARBITER

The 8289 Bus Arbiter, U85 (3D6), controls the 8089 IOP's access to the Multibus interface (see Figure 4-4). The 8289 monitors the 8089's status lines (S01/, S1/ and S2/). When the lines indicate that the 8089 needs a Multibus interface cycle, and the 8089 does not presently control the bus, the 8289 activates a bus request (BREQ/). The low on BREQ/ is transmitted to the bus priority resolving circuitry in the host processor, which returns a low on Bus Priority In line BPRN/, giving the 8089 access to the Multibus interface. Having received access to the Multibus interface, the 8289 activates its busy signal (BUSY/), indicating to the other masters on the system that the Multibus interface is in use. The 8289 then activates the address enable signal (AEN/), which is transmitted to the 8288 Bus Controller, U86 (3C4), to enable its command outputs, to the 8284A Clock Generator, U59 (4C6), to enable its bus ready logic, and to the System Address Latches, U76-U78 (4X2), to allow an address to be gated on to the Multibus interface.

4-10. BUS CONTROLLER LOGIC

The 8288 Bus Controller, U86 (3C4), decodes the status line outputs (S0/, S1/ and S2/) from the 8089 IOP and generates the appropriate bus cycle signal. Table 4-1 shows the different signals generated for each configuration of the IOP's status lines.

These bus cycle signals can be divided into two groups: those which allow the 8089 to access system memory (MWTC/ and MRDC/) and those which allow the 8089 to access local memory (I-AIOWC/ and I-IORC/). The 8089 uses the I/O Read (I-IORC/) and I/O Write (I-AIOWC/) signals to read information from the local ROM, U82 and U83, (6X7), or to read from or write to the local RAM, U94 through

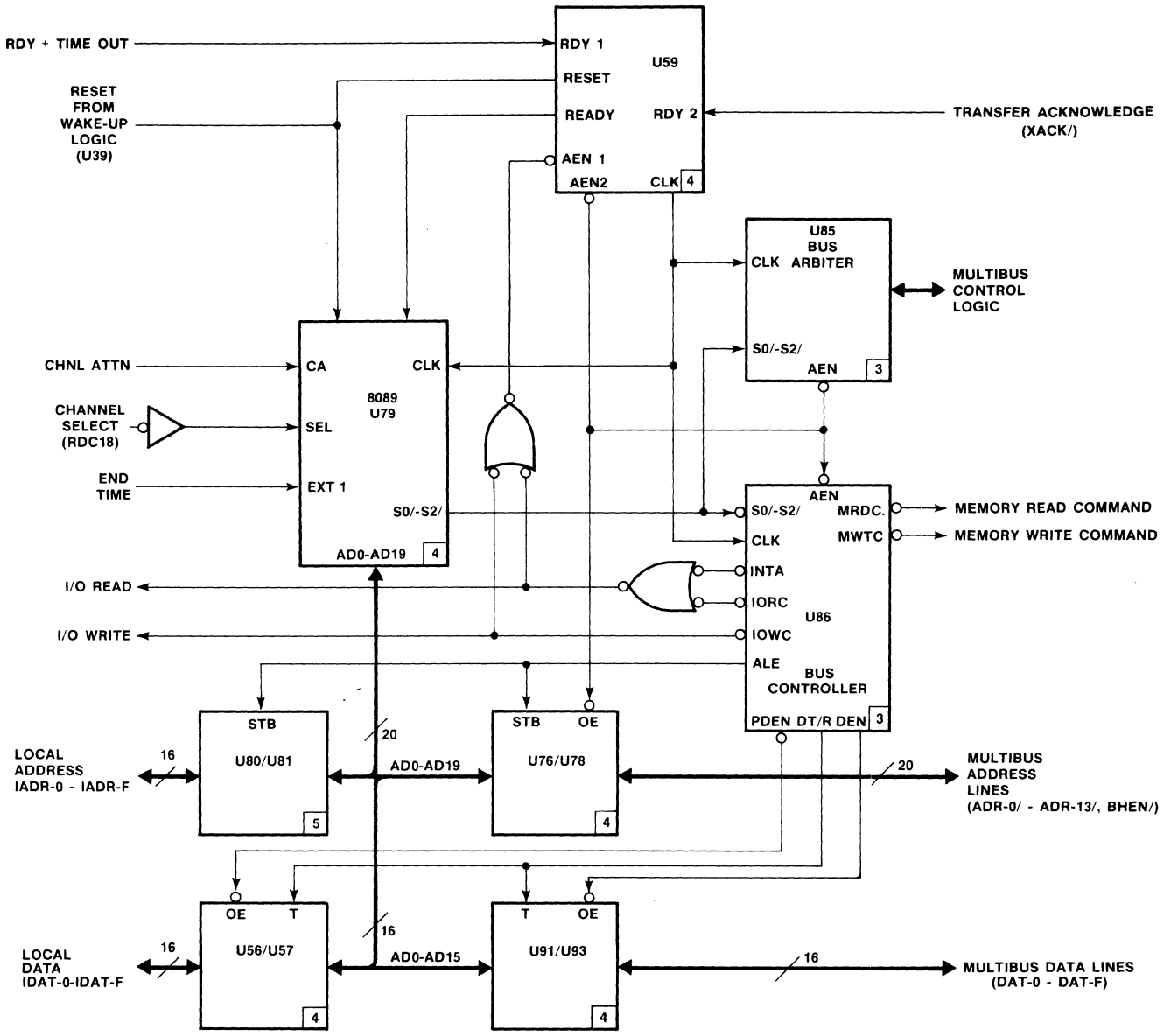


Figure 4-4. Bus Arbiter and Bus Controller Logic

Table 4-1. 8089 Status Line Decodes

Status Input			CPU Cycle	8288 Command
S2/	S1/	S0/		
0	0	0	Instruction Fetch, Local	INTA/
0	0	1	Read Memory, Local	IORC/
0	1	0	Write Memory, Local	IOWC/, AIOWC/
0	1	1	Halt	None
1	0	0	Instruction Fetch, System	MRDC/
1	0	1	Read Memory, System	MRDC/
1	1	0	Write Memory, System	MWTC/, AMWC/
1	1	1	Passive	None

U97, (6X4). The 8089 also uses I-IORC/ and I-IOWC/ to gate on the Read and Write Function Decoders, U33 and U32 (5B2 and 5A2). The function decoders are explained further in paragraph 4-20.

The 8288 Bus Controller also generates a group of signals that control address and data flow throughout the iSBC 220 controller. The Address Latch Enable line (ALE) is used to strobe addresses from the 8089 into both the system Address Latches, U76-U78 (4X2), and the Local Address Latches, U80-U81 (5X7).

Data Transmit/Receive (DT/R), Data Enable (DEN), and Peripheral Data Enable (PDEN/) control the data flow through the controller. DT/R controls the direction of data transmission through the Multibus interface and local transceivers. If DT/R is high, data is transmitted either on to the Multibus interface through transceivers U91, U92 and U93 (4X7) or on to the local bus through transceivers U56 and U57 (4X6). If DT/R is low, the data transfer is in the opposite direction, into the 8089 through one of the two sets of transceivers. DEN and PDEN controls the selection of the transceivers. If DEN is high the Multibus interface transceivers U91, U92 and U93 are enabled, and if PDEN/ is low (indicating a peripheral cycle) local transceivers U56 and U57 are enabled.

4-11. MULTIBUS INTERFACE DATA TRANSFER LOGIC

The controller has three sets of Multibus interface data transceivers: low-byte transceiver U92, which buffers DAT-0/ through DAT-7/, high-byte transceiver U91, which buffers DAT-8/ through DAT-F/, and swap-byte transceiver U93, which takes the data from DAT-0/ through DAT-7/ on the Multibus interface and switches it to high-byte data bus lines AD8 through AD15 on the controller board (see figure 4-5). This byte-swap is performed only when the controller is interfacing with an 8-bit system memory. In this case, every odd address read from system memory is transmitted to the high-byte data lines of the controller. The procedure is reversed when writing to the 8-bit system memory. Three signals control the transceiver: ENBL HI BYTE/ (5C1), which controls the high-byte transceiver; ENBL LO BYTE/ (5C1), which controls the low-byte transceiver (derived from ADRO/); and ENBL SWAP BYTE/ (5C1), which controls the swap byte transceiver. Figure 4-5 shows when each of the control signals is active.

4-12. CONTROLLER INITIALIZATION

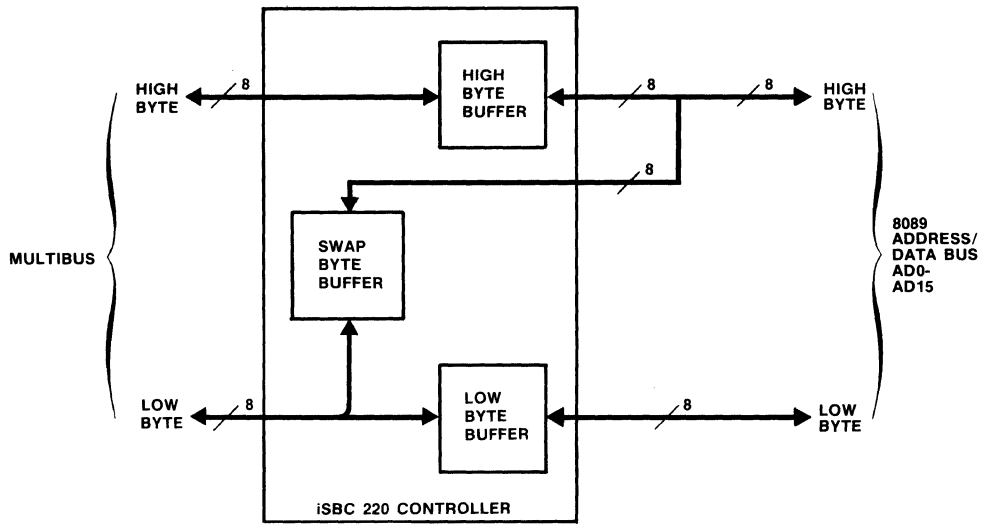
Before data can be transferred between system memory and the controller, the controller must be initialized. The initialization procedure, which is described in paragraph 3-12, involves:

1. Resetting the 8089 IOP.
2. Clearing the reset.
3. Establishing a communication link between the 8089 and the I/O communications blocks in system memory.
4. Reading the disk drive parameters from system memory to the controller on-board RAM.

The following paragraphs describe the hardware operations that take place during this initialization procedure.

4-13. WAKE-UP ADDRESS COMPARATOR

For the purpose of resetting the controller, clearing the reset or getting the attention of the 8089 IOP (raising CA), the host addresses the controller as an I/O port in its system I/O space. To perform one of these functions it writes a one byte command to the specified I/O port called the wake-up I/O port (see Figure 4-6). Table 4-2 shows the three possible commands. The user determines the address of the I/O port at which the controller is to reside (called the "Wake-Up Address") and sets the address on the Wake-Up Address switches S1-1 through S1-8 and S2-3 through S2-10 (2x6), on the controller board. When the host issues a write command (IOWC/) to the Wake-Up Address in system I/O space, U70 and U71 (2A5) on the controller compare the address with the switch settings. If they agree, WAKEUP/ is pulled low, enabling the controller to decode the command on the Multibus interface data lines and determine the action to be taken.



	8-BIT SYSTEM MEMORY		16-BIT SYSTEM MEMORY	
	I-ADR0/ = L	I-ADR0/ = H	I-ADR0/ = L	I-ADR0/ = H
ENBL LO BYTE/	L	H	*	L
ENBL SWAP BYTE/	H	L	*	H
ENBL HI BYTE/	H	H	*	L
*NOT APPLICABLE				

Figure 4-5. Data Transmission Between Multibus™ Interface and Controller Data Transceivers

The host may use 8-bit or 16-bit I/O port addressing. The user sets switch S2-2 (2A7) to indicate to the controller the type of addressing that is being used. When S2-2 is open (8-bit addressing), pin 9 of U70 is held high, creating a "don't care" situation for the outputs of High-Byte Wake-Up Address Comparators U72 and U73.

Table 4-2. Host Wake-Up Commands

Command	Description
00H	Clear Interrupt and Clear Reset
01H	Channel Attention (Start 8089 IOP)
02H	Reset 8089 IOP

As it is discussed in Chapter 3, the controller also uses the setting of the Wake-Up Address switches to calculate the address of the first byte of the Wake-Up Block, which is the first I/O communications block in system memory.

4-14. CONTROLLER RESET AND CLEAR

The first operation that must be performed during the initialization of the controller is a reset of the 8089 IOP. To reset the 8089, the host processor writes an 02H to the wake-up address. The WAKE-UP/ lines goes low and gates the 02H (DAT-0/ high and DAT-1/ low) into the Wake-Up Decoder, U39 (3B7), producing a low on the controller reset (CNTLR RST/) line. A low on CNTLR RST/ resets the 8089 (4X4), resets Read/Write Control Logic U36 (sheet 8) and clears Control Register U21 (11B7). Once the controller has been reset, the host processor writes a 00H (Clear Interrupt) to the wake-up address, which clears the reset. The Wake-Up Decoder U39 decodes the highs on DAT-0/ and DAT-1/ to raise CNTLR RST/.

4-15. ESTABLISHING A LINK WITH I/O COMMUNICATIONS BLOCKS

Following a power-up event or a software reset (02H written to the wake-up I/O port), the link between

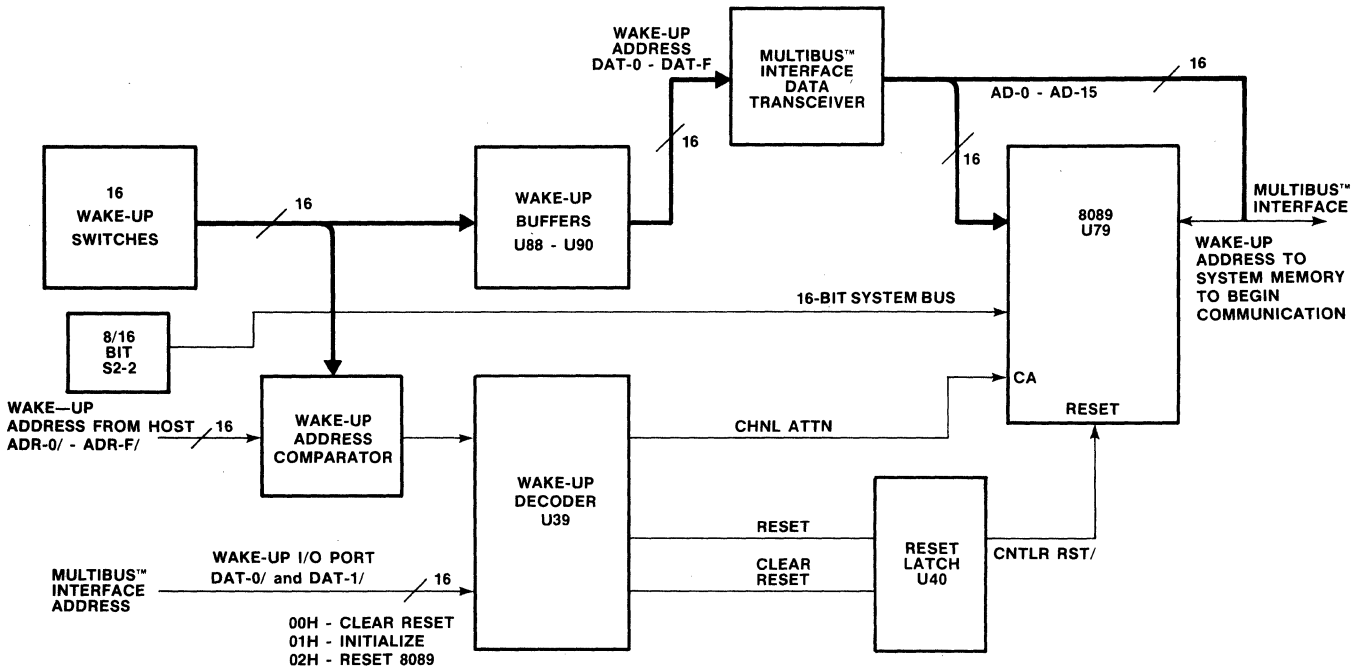


Figure 4-6. Wake-Up Address Logic

the controller and the I/O communications blocks in system memory must be established. To establish this link, a clear reset (00H) is written to the wake-up I/O port followed by a channel attention (01H). The 01H is gated into U39, producing a high on CHNL ATTN, which in turn raises the CA input to the 8089 IOP (4C4).

Being the first Channel Attention following reset, the 8089 begins an internal initialization process. The first step of this process is to do a fetch of address FFFF6H. The address is transmitted on the 8089 Address/Data lines (AD0-AD15) to latches U80 and U81 (5C7). Gates U61, U69, U71 and U47 (5D4) decode the output of these latches. The output of U71 enables U67, gating the status of the 16-bit SYS BUS switch (S2-1) through Data Bit 0 line (DAT-0/) to the 8089. Switch S2-1 on (16 Bit SYS BUS/ low) indicates that the host memory system supports 16-bit data transfers and S2-1 off indicates 8-bit data transfers. Inverter U67 also generates Transfer Acknowledge (XACK/), which is sent to the 8089 (through the 8284A) indicating that the operation has been completed.

After determining the width of the system bus (8-bit or 16-bit) the 8089 fetches the addresses shown in figure 4-7 as part of the initialization sequence.

Fetching addresses FFFF8/9H gates zeros into the 8089. Fetching addresses FFFFA/BH causes the GATE SWS/ line (5C1) to go low. GATE SWS/ gates the settings of the wake-up address switches, S1-1 through S1-8 and S2-3 through S2-10 through buffers U88, U89 and U90 (2X2) and into the 8089. The 8089 multiplies the settings of the wake-up switch by 2^4, to determine the 20-bit address of the wake-up block, the first I/O communications block in system memory. The 8089 then uses this address to fetch the wake-up block and establish a link with the I/O communications blocks. On subsequent channel attentions (host writes 01H to the wake-up I/O port), the 8089 skips the wake-up block and goes directly to the channel control block, the second I/O communications block. The 8089 uses the channel control

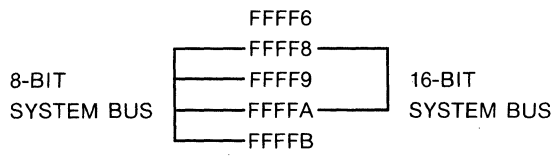


Figure 4-7. Address Fetches In Initialization Sequence.

block to obtain the starting address of the controller's ROM resident I/O transfer program (also called the channel control program). From this point on, this firmware program directs the controller activities. One of the first operations of the firmware is to again fetch the starting address of the wake-up block. It then links its way through the channel control block and the controller invocation block to the I/O parameter block where it obtains instructions and parameters for a specific I/O operation.

4-16. INTERRUPT PRIORITY LOGIC

Wire wrap pins W4-C and W4-0 through W4-7 (3B2) allow the user to select the interrupt priority of the controller with respect to other peripherals in the system. To issue an interrupt to the host, the 8089 IOP writes an 0100H to local I/O port 8010H. A high on data line BDAT-8 and a low on write decoder line WDC10/ is then generated, causing interrupt latch U60 (3B6) to pull its output high and pull the selected interrupt line to the Multibus interface low. A 00H written to the system I/O port wake-up address, clears the interrupt (refer to paragraph 4-14).

Jumper pins W2-C, 1 and 2 allow the user to select the Any Request option. A jumper installed between pins W2-C and 1 causes the controller to relinquish control of the Multibus interface following a request from a higher priority device only. A jumper installed between pins W2-C and 2 causes the controller to relinquish control of the Multibus interface following a request from any device, higher or lower priority.

4-17. LOCAL MEMORY MAP

As was discussed in the Functional Overview, the 8089 IOP addresses the ROM, RAM and the disk communications side of the controller circuitry as local memory. Figure 4-8 shows a map of this local memory. The following paragraphs discuss the ROM, RAM and I/O ports.

4-18. ROM

The controller ROM, which contains the 8089 IOP's disk control program, consists of two (4K x 8-bit) ROM devices, U82 and U83 (6C7). On any read from local memory in the range of 0000H to 1FFFH, chip select decoder U39 (5C2) decodes address lines IADR-E and IADR-F and pulls ROM chip-select line CSROM/ low, enabling the ROM devices.

4-19. RAM

The controller RAM consists of four (1K x 4-bit) RAM devices, U94 through U97 (6X4). On any read or write to local memory in the range of 4000H to 47FFH, chip select decoder U39 (5C3) pulls RAM chip-select line CSRAM/ low, enabling the RAM devices.

4-20. LOCAL MEMORY MAPPED I/O PORTS

The 8089 IOP views the controlling devices in the disk control circuitry (such as ID comparators, counters, write buffer, read buffer, etc.) as local I/O ports, each with an address in local memory space. To enable one of these devices, the 8089 executes a read or a write to the devices respective address. On any read or write to local memory in the range 8000H through 8038H, chip select decoder U39 (5C3) pulls its pin 10 low. When this low on pin 10 of U39 is accompanied by a low on I/O read line I-IORC/, read I/O port address decoder U33 (5B2) is enabled; when the low on pin 10 of U39 is accompanied by a low on I/O write line I-AIOWC/, write I/O port address decoder U32 (5A2) is enabled. When enabled, U32 or U33 decode local memory address lines IADR-3 through IADR-5 to select the desired disk control device. Table 4-3 shows the address of each local I/O port and its function.

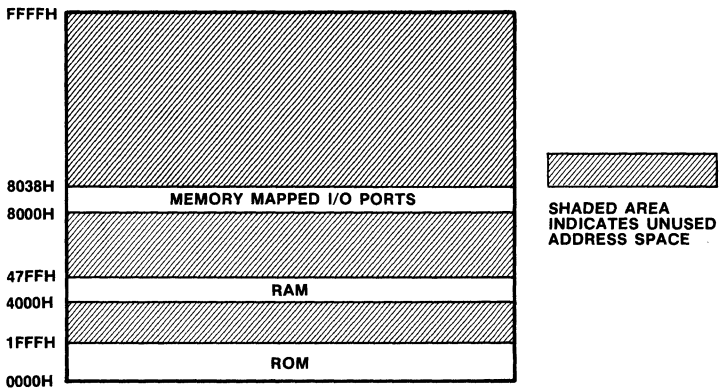


Figure 4-8. Local Memory Map

Table 4-3. Local I/O Ports

Address	Read (U33 Enabled)		Write (U32 Enabled)	
	Enable Line	Function	Enable Line	Function
8000H	RDC00/	Read Disk Status	WDC00/	Write to bits 0, 1, 4, 5, 6 and 9 of SMD bus.
8008H			WDC08/	Clear index and ID not compare latches
8010H			WDC10/	Write to control register.
8018H	RDC18/	Raise 8089 Ch 2 CA input.	WDC18/	Write to SMD bus Unit Select register
8020H	RDC20/	Read contents of counter 2	WDC20/	Load counter 0
8022H	RDC20/	Read contents of counter 1	WDC20/	Load counter 1
8024H	RDC20/	Read contents of counter 2	WDC20/	Load counter 2
8026H			WDC26/	Write mode word
8028H	RDC28/	Read contents of read buffer	WDC28/	Write data to write buffer
8030H			WDC30/	Write sector ID to high comparator, start track format operation.
8038H			WDC38/	Write sector ID to low comparator

4-21. CONTROLLER TO DISK DRIVE COMMUNICATIONS

The following discussion provides a detailed functional description of the section of the iSBC 220 SMD Disk Controller that communicates with the disk drive through the SMD interface. The discussion is broken into four areas: (1) description of the SMD interface signals; (2) explanation of how the controller formats a disk prior to the performing read and write functions; (3) explanation of how writes and reads are performed; and (4) descriptions of the various circuits that perform the data transfer.

4-22. CONTROLLER TO DISK DRIVE INTERFACE

All the signals that are transmitted between the controller and the disk drives are transmitted through either the Control Cable or the Read/Write Cable. The physical configuration of these cables is described and illustrated in Chapter 2. The SMD specification requires signals that are transmitted between the drives and the controller to be differential signals, and receivers to translate the differential input signals to levels compatible with internal logic.

The following functional description of the interface signals is based on a typical drive installation. Different drive manufacturers may use these signals for other functions. While reading this discussion, consult the drive manufacturer's user manual for the specific drive being employed.

4-23. CONTROL CABLE SIGNALS

Control and status information is exchanged between the controller and the drive through the Control Cable. Output signals are defined as those signals that the controller transmits and input signals as those the controller receives. The Control Cable is connected to J1 on the iSBC 220 board and goes to the first drive in a string of up to four. Subsequent Control Cables are connected from drive to drive in a daisy chain fashion. The eighteen output lines, which the controller transmits to the disk drives through the Control Cable, are:

a. Device Select Enable Line
b. Device Select Lines (4)
c. Function Tag Lines (3)
d. Bus Out Lines (10)

The eight input lines, which the controller receives from the disk drives through the Control Cable, are status lines.

4-24. SELECTION LINES

The controller transmits five disk drive selection signals to the drives:

a. **Device Select Enable.** (Unit Select Strobe). Enables the decode logic that the drive uses to decode the device select lines.
b. **Device Select 0 through Device Select 3.** Four binary coded lines select the desired disk drive. A module select identifier plug in each drive (which the operator can change) determines the address of each drive. See the disk drive manufactuer's OEM manual for instructions.

Table 4-4. Function Tag/Bus-Out Definitions

BUS OUT BITS	FUNCTION DECODE		
	SET CYLINDER-TAG 1	SET HEAD ADDRESS-TAG 2	CONTROL SELECT-TAG 3
0	Cylinder Address 1	Head Select 1	Write Select
1	Cylinder Address 2	Head Select 2	Read Select
2	Cylinder Address 4	Head Select 4	Offset Forward
3	Cylinder Address 8	Head Select 8	Offset Reverse
4	Cylinder Address 16	Head Select 16	Unsafe Reset
5	Cylinder Address 32		Address Mark
6	Cylinder Address 64		Rezero
7	Cylinder Address 128		Data Strobe Early
8	Cylinder Address 256		Data Strobe Late
9	Cylinder Address 512		Not Used

4-25. FUNCTION TAGS AND BUS-OUT LINES

The controller generates three different function tag signals. These lines are used in conjunction with the Bus-Out lines to control the disk drive operations. Only one of the tag lines may be active at a time. Table 4-4 shows how the Bus-Out lines and tag lines are decoded to perform tasks or transmit data to the disk drive.

a. SET CYLINDER (TAG 1). Transfers the cylinder address bits to the drive via the Bus-Out lines and initiates internal drive functions. Typically the Set Cylinder Tag line loads the next cylinder address into the drive logic at the leading edge of the set cylinder pulse (see Figure 4-9), allowing sufficient time for the drive logic to perform necessary internal operations (difference calculations and set up of the servo circuitry), and initiates the seek signal within the drive coincident with the trailing edge of the set cylinder pulse. The typical times required for this operation are provided in Figure 4-9.

b. SET HEAD ADDRESS (TAG 2). Gates the Bus-Out lines to the selected disk drive. These lines are decoded in the drive to select the fixed or removable volume and the head within the volume. Selection of a surface is normally referred to as selecting a head (the head that corresponds to a surface). The head address corresponds to one track in a given cylinder of a fixed or removable volume. The head number within a volume is always assumed to start with zero. Bus Out bit 4 selects the type of volume: fixed or removable.

c. CONTROL SELECT (TAG 3). Gates coded commands, as listed in Table 4-4, to a selected disk drive. A description of the functions of these commands is given in Table 4-5. A detailed functional description of each of the listed operations can be found in the disk drive manufacturer's OEM manual.

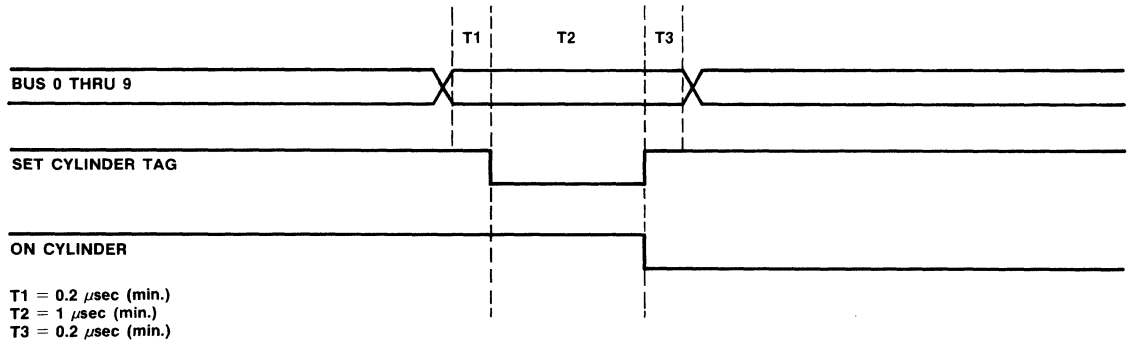


Figure 4-9. Set Cylinder Timing

Table 4-5
Control Tag and Bus Out Line Functions

CONTROL FUNCTIONS	DESCRIPTION
WRITE SELECT	Enable the write circuitry in the drive, permitting write data, sent to the drive over the Read/Write cable, to be written on the selected disk surface. If one of the following conditions occurs when the Write Select signal is sent to the drive, the drive fault logic activates and sends an Unsafe status to the controller: 1. no write data is sent, 2. the Write Protect Line in the drive is active, 3. offset is active, or 4. the drive Ready signal is not active,
READ SELECT	Causes the drive circuitry to read from a selected area of the disk, convert that raw data to Non Return To Zero format (NRZ) data, and transmits it over the Read/Write cable to the controller.
OFFSET FORWARD	Causes the head positioner in the selected disk drive to move the selected R/W head a fixed distance towards the center of the disk, from the nominal track position. If this signal is activated while Write Select is active, the drive will activate the Fault status line to the controller. This signal is activated in an attempt to recover hard to read data (suspected soft error).
OFFSET REVERSE	Causes the head positioner in the selected disk drive to move the selected R/W head a fixed distance towards the outer edge of the disk, from the nominal track position. If the signal is activated while Write Select is active, the drive will activate the Fault Status line to the controller. This signal is activated in an attempt to recover hard to read data (suspected soft errors).
FAULT CLEAR	Clears the Fault or Unsafe status line in a selected drive. This signal has no effect unless the fault condition has been corrected.
ADDRESS MARK ENABLE	Used in conjunction with either Write Select or Read Select. When it is used with Write Select, it allows the controller to write an AM. When used with the Read Select, it indicates the controller is looking for an address mark. Activating the address mark enable line in the drive enables the address mark detection logic to look for a 16 bit gap in read data pulses. See AM Found Status Line, Table 4-6.
REZERO	Causes the head positioner in the selected disk drive to position the Read/Write heads over cylinder 0. Activation of the Rezero signal in the drive causes the Seek Error and the Head Address Register (HAR) to be reset and set to zero respectively.
DATA STROBE EARLY	Used with the Read Select Signal to a selected disk drive to attempt to recover hard to read data (suspected soft error). Activation of this signal causes the clocked data separator in the disk drive to strobe the data at a fixed time earlier than nominal.
DATA STROBE LATE	Used with the Read Select Signal to a selected disk drive to attempt to recover hard to read data (suspected soft error). Activation of this signal causes the clocked data separator in the disk drive to strobe the data at a fixed time later than nominal.

4-26. STATUS LINES

The eight status lines sent to the controller form a selected disk drive over the Control Cable provide:

a. an indication of the drive status

b. data relating to the position of the R/W heads on the disk surface, which the controller uses to condition circuitry for timing, reading, or writing. A description of these status lines is provided in Table 4-6.

Table 4-6
Status Line Definitions

STATUS LINE	DESCRIPTION
INTERFACE ENABLE	Indicates to the drive that continuity exists between the controller and the drive via the Control Cable. Without this continuity, line receivers in the disk drive will not be enabled for communications with the controller. This signal may be called Open Cable Detect in some drive manuals.
INDEX MARK	A pulse received from a selected disk drive once every disk revolution (16.67 ms); nominally 2.4μs in duration. The index pulse indicates to the controller that a particular reference on the disk is passing under the R/W heads.
FAULT	Indicates to the controller that, within the selected disk drive, an unsafe condition has been detected, which would make the reliability of read/write operations questionable. Normally, logic in the drive will disable the read, write, and positioning circuitry until a Rezero operation, Fault Reset or operator intervention occurs. Depending on the particular disk drive, some conditions that would cause the Fault circuitry to activate are: a. Write Unsafe b. Write Transition Failure c. Write Current Failure d. Power Failure e. Multiple Head Select
SEEK ERROR	Indicates to the controller that the selected disk drive has failed to complete an initial head load, Seek operation, or Rezero operation within drive specified time limits. Seek Error is also set when an invalid cylinder address is sent to the drive. This signal may be reset by the controller issuing a Rezero command or by operator intervention.
ON CYLINDER	Indicates to the controller that the selected disk drive has successfully completed the initial head load, a Seek operation, or Rezero operation within drive specified time limits. On Cylinder goes inactive coincident with the issuance of Seek, Offset Forward, Offset Reverse or Rezero and is reactivated at the completion of the operation.
UNIT READY	Indicates to the controller the selected disk drive is: a. Selected b. The disk pack is up to its nominal speed c. The head load operation was completed successfully d. Fault (unsafe) conditions do not exist in the drive e. The drive is not in the CE mode or off-line (see drive manufacturer's OEM manual).

Table 4-6. Status Line Definitions (Continued)

STATUS LINE	DESCRIPTION
WRITE PROTECTED	Indicates to the controller that the selected disk drive's Write Protect switch is in the Protect position. The operator sets this switch to prevent the controller from writing over sensitive data. To deactivate this line, the switch must be moved to the other position. Receipt of this signal by the controller prevents writing on the selected disk drive.
ADDRESS MARK FOUND	Indicates to the controller that the selected disk drive has detected an area of a selected track that has an absence of data bits for at least 16 bit cell times. This signal is used to establish timing in the controller for the assertion of Read Enable, prior to reading an ID field and header.

4-27. READ/WRITE CABLE SIGNALS

Read Data, Write Data, Clocks, and some status lines constitute the information exchanged over the Read/Write cables. Output signals are defined as those signals that the controller transmits to the disk drives, and input signals those that the controller receives. The Read/Write cables are connected from the controller to the disk drive in radial fashion, that is one cable from the controller to each of the drives. The Read/Write cable that connects to J2 on the controller splits into two cables; one going to the drive at physical address 0; the other going to physical address 1. The Read/Write cable that connects to J3 on the controller splits into two cables; one going to the drive at physical address 2; the other going to physical address 3. The physical configuration of these cables is explained and illustrated in Chapter 2. Each of the discussed signals are received or output by all the drives. The controller multiplexes all received signals (except SELECTED-DEV 0-3)(9B8) to common lines (see Figure 5-2, sheet 10). All output signals from the controller are fanned out to all four cables simultaneously. Therefore, the descriptions provided apply only to signals of a selected Read/Write cable.

a. WRITE DATA. The Write Data differential line pairs from the controller transmit the NRZ data to the drive for recording on the disk surface. The write data transmitted over this differential line pair is synchronized with the Write Clock signal.

b. WRITE CLOCK. The Write Clock signal sent to the drive via differential line pairs synchronizes the write data. The write clock is derived from the Servo clock signal sent to the controller from the selected disk drive and thus ensures the proper bit rate transmission when writing as well as when reading the data back.

c. READ DATA. The Read Data signal is transmitted from the disk to the controller via differential line pairs. This data read from the disk pack has been separated from the clocks and put into the NRZ format. The data is transmitted to the controller in a serial fashion, bit-by-bit. The read data received at the controller and gated through the line receiver is strobed into the controller logic by the Read clock.

d. READ CLOCK. The Read Clock signal is transmitted to the controller via differential line pairs. It is derived in the disk drive from the data read from the disk. The controller uses Read Clock to strobe the read data onto the board at the proper time relationship, see U63 (10B2).

e. SERVO CLOCK. The 9.677 MHZ Servo Clock signal is transmitted to the controller via differential line pairs. This signal is transmitted back to the drive as Write Clock. Since the Servo Clock is derived from the servo signal that is recorded on the rotating disk, it reflects any speed variations.

4-28. SELECTED

The only status line sent to the controller via the Read/Write cable is the Selected signal. This signal is a result of the disk drive comparing the four device select lines to the module select plug and achieving a favorable comparison. When the Selected line for a particular drive activates, it, in turn, enables the line receivers for the selected drive's Read/Write cable on the controller.

4-29. CONTROLLER TO DISK DRIVE INTERFACE TIMING

The following paragraphs provide a detailed discussion of the inter-circuit timing that occurs when formatting a disk, writing to a disk or reading from a

disk. The discussion is provided to describe the interaction of the timing logic shown on Sheet 8 of the Schematic Diagram, with the disk drive interface receivers and drivers shown on sheets 9 through 12 and the other data transfer circuitry described in paragraphs 4-34 through 4-39.

4-30. DMA MODE

In general, when the controller is performing a read or a write function it locates the area of the disk where the read or write is to be performed, then enters its DMA mode to perform the actual transfer. (The process of locating the area to be read or written to is discussed in the following paragraphs.) In the DMA mode, the 8089 IOP (see figure 4-2) controls the transfer of data between the local RAM block and the write and read buffers (called the read/write port). The data transfer circuitry on the controller board controls the transfer of data between the read/write port and the disk.

The RDY (Ready) line (8D1) is used for hand shaking between the 8089 and the data transfer circuitry. When RDY is low, the 8089 is quiescent; when RDY is high, the 8089 performs a DMA transfer of data either from local RAM to the write buffer (block-to-port) or from the read buffer to local RAM (port-to-block). Gate U46 (8D3) controls the RDY line.

To perform a write or a read, the 8089 executes firmware to set up data (write only) and condition the hardware for the selected operation. It then enters the DMA mode and attempts to transfer data. At this time; the R/W GATE (8D1) is high (see Figure 4-10); U45-8 (8D3) is high, held so by the low on the ENBL XFER line (8D1); and the R/WDC 28 line, the output of U42-6 (8D7), is low. The low on R/WDC 28 is thus keeping RDY activated. On this first attempt to transfer data in the DMA mode, the 8089 activates either RDC 28/ or WDC 28/ (8D8), depending on whether a read or a write is being performed, respectively (refer to paragraph 4-34). When RDC 28/ or WDC 28/ is activated, the R/WDC 28 lines is activated, lowering RDY and putting the 8089 into its quiescent (wait) state. When the controller's data transfer circuitry has found the area on the disk where the read or write is to begin, it activates ENBL XFER (8D1). On the next occurance of a Bit Ring-0 pulse, BR-0 (8D1), following the activation of ENBL XFER, U45-8 (8D3) is activated, activating RDY. The 8089 then immediately performs the data transfer (writes a word into the write buffer or reads a word from the read buffer) and lowers R/WDC 28. On the next clock into U45-11, U45-8 is raised. On the 8089's next attempt to perform a data transfer, R/WDC 28 is also raised, lowering RDY. The data transfer does not occur and the 8089 goes into its wait state. During this time, the SER/DES either transfers the word from the write buffer to the disk or reads another word from

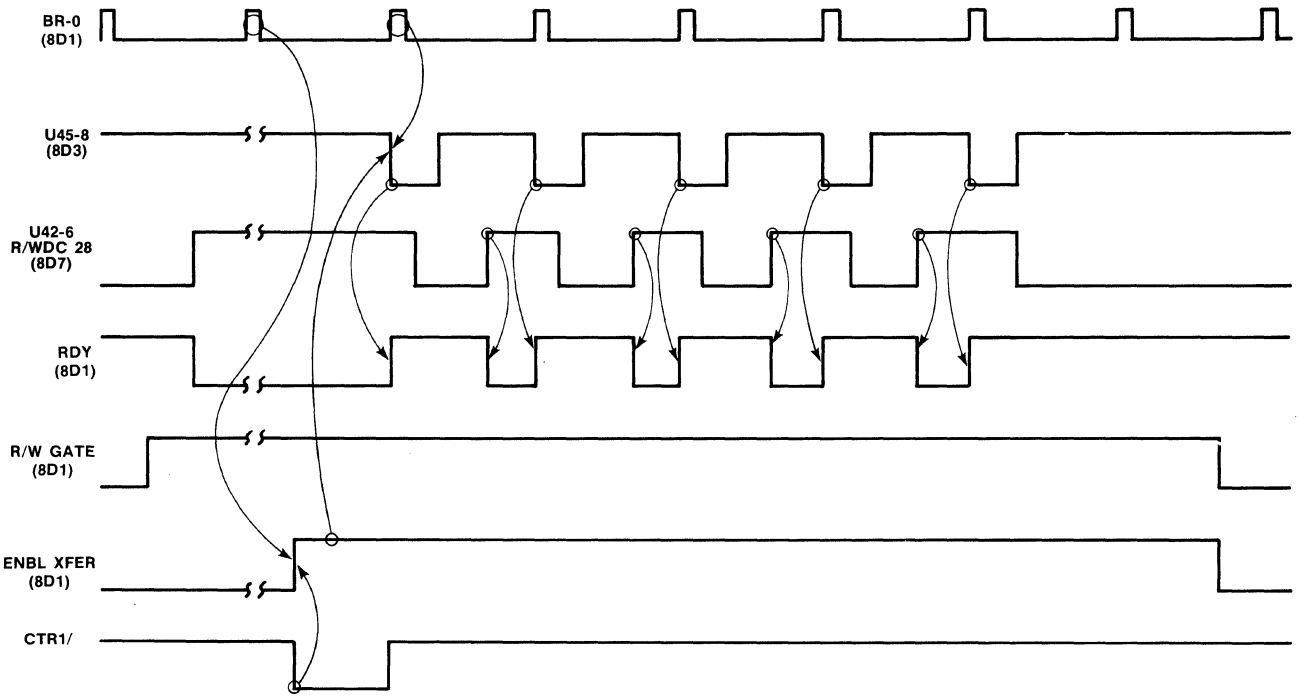


Figure 4-10. Timing Diagram for RDY Signal

the disk into the read buffer. Then on the next BR-0 pulse, RDY is again activated and the next DMA data transfer occurs. The 8089 continues in this DMA mode until the R/W GATE line is lowered.

4-31. DISK FORMATTING

Before the surfaces of a disk volume can be used for the writing and reading of data, the disk volume must be formatted. Formatting is the operation of writing all the address fields, gaps, ID headers, etc. for the complete disk volume. The controller performs this operation under software control. The software routine that controls this disk formatting operation allows only a single track to be formatted for each Format command. The host thus issues a new Format command to the controller board for each track to be formatted until the formatting of the entire disk volume is complete.

The implementation of the Format command is divided into two operations. During the first operation, address marks (AM), gaps and ID fields are written during a single disk revolution. During the second operation, data fields are written (using the write data sequence described in paragraph 4-32) with user supplied data. The second operation requires two disk revolutions, one to write the odd physical data fields (1, 3, 5, ...) and one to write the even physical data fields (0, 2, 4, ...). Three disk revolutions are thus required to format a single track. The hardware execution portion of the format operation is discussed in the following paragraphs.

When the Format command is issued to the controller, the 8089 IOP begins the format operation by performing a Seek to the desired track (cylinder). When the heads are positioned over the desired track, the 8089 writes a 0000H to I/O port 8010H (decoded as WDC 10/), which enables U21 (11B7) and activates the Write Gate-F (U21-12), Control Select-Tag 3 (U21-2) and FORMAT lines (11A1). See Figure 4-11.) Write Gate-F, which is transmitted through U49 (11B4) and BUS 0 (11B1), and Control Select-Tag 3 are transmitted to the selected drive where they enable the write circuitry. The controller then writes all zeros to the drive while the 8089 waits for the receipt of the first Index pulse (12D8).

The receipt of Index sets latch U41 (12D5), which in turn sets bit F of the Status Register, U26 (12D3). To monitor the Status Register, the 8089 reads I/O port 8000H (decoded as RDC 00). Upon detecting Index, the 8089 writes a XXXXH to I/O port 8030H (decoded as WDC 30), which enables the format logic U41 (8B6). Then the 8089 writes a 0001H to I/O port 8000H (decoded as WDC 00), which activates WRT GATE and allows the writing of address marks (AM) and Sector ID fields.

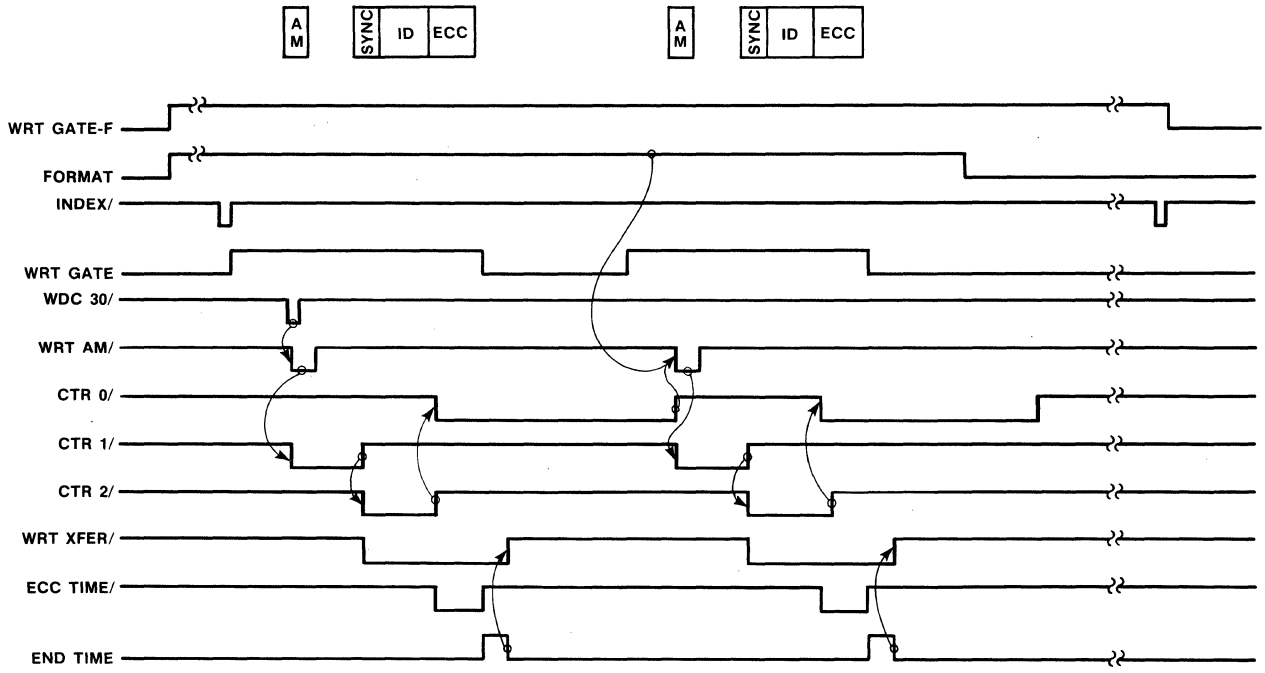


Figure 4-11. Timing Diagram for Disk Formatting

The time that the 8089 allows between the detecting of Index and the activating of U41 (12D5) is approximately 38 byte times, which is the predetermined time for the first gap of the track format, G1 (see figure 3-1 for a pictorial representation of the track format). Also during G1, the 8089 writes the sync byte (0019H) to the write buffer, U50 and U53 (7C7 and 7D7), by writing to I/O port 8028H (decoded as WDC 28/). It performs this operation in preparation for writing the ID field on the track.

When U41 (8B7) is activated, it activates WRT AM/ (8B1), which is transmitted through BUS 5 (11C1) to the drive, causing the AM to be written on the disk. WRT AM/ also starts counter 1, CTR 1 of U64 (8A7). (The 8089 preset the counters in U64 at the beginning of the format operation.) When CTR 1 times out at the end of 11 byte times, it activates the WRT XFER/ line through U41-9 (8C4) and CTR 2. The activation of WRT XFER/ initiates the 8089's DMA mode (as discussed in paragraph 4-30), during which time the sync byte and the sector ID are written onto the disk. CTR 2 times out at the end of the ID field, starting CTR 0 and activating the ECC TIME line (8B1). During the ECC TIME, the ECC code from the ECC generator is written following the ID field (refer to paragraph 4-37 for a description of the operation of the ECC generator). At the end of ECC TIME, the END TIME line is enabled, which lowers the WRT XFER/ line and takes the 8089 out of the DMA mode.

CTR 0 is set for a time equal to the ECC+G3+DATA +G4, which the 8089 sets according to the sector size selected for the drive. When CTR 0 times out, it activates WRT AM/ and CTR 2, which begins the formatting of the second sector. This procedure is repeated until the 8089 determines that the last ID field has been formatted. The 8089 then begins searching for the Index pulse. Upon receipt of Index the 8089 resets WRITE GATE-F and FORMAT, inhibiting the writing of the next AM. The 8089 then continues through the Format routine to the second operation, which is the writing of the data fields with user supplied data. The write data function, discussed in the following paragraphs, describes the write data operation.

4-32. WRITE DATA TRANSFER

The write operation is divided into two steps: (1) read sector ID and (2) write data. When a write is initiated, the 8089 IOP writes 01H to I/O port 8000H (decoded as WDC00/). Latch U2 (11C6) then sets BUS 5 (11C1) high, enabling the drives address mark (AM) search; and sets BUS 1 (11B1) high, enabling the drive's read circuitry and raising the read gate, RD GATE (11A1). (See figure 4-12.)

The 8089 has previously written to I/O port 8020H (decoded as WDC20/) to load counters 0, 1 and 2 of U64 (8A7). It also writes to I/O ports 8030H and 8038H (decoded as WDC30/ and WDC38/), loading the ID of the sector to be written to, into the 32-bit ID comparator logic.

When an address mark is found, the drive activates the AM FND/ line, which resets U41 (8C7) and activates the ID FIELD/ line. The enabling of the ID FIELD/ line lowers the AM ENABLE gate to the drive and initiates the search for the sync byte.

In searching for the sync byte, serial data from the disk is read into the SER/DES. Sync byte comparator U68 and U58 (7B5) monitors the outputs of the SER/DES and pulls the SYNC BYTE/ line (9C6) low when 19H — the sync byte — is detected. The enabling of SYNC BYTE/, enables the SYNC FND/ line, which in turn activates the ID comparator U19, U20, U37 and U38 (9DX) and word clock U35 (8D6). (See the discussion of the Sync Byte Comparator Logic in paragraph 4-35.)

SYNC FND/ also raises the ENBL XFER line, which enables the ECC Generator logic (7AX) and Ready Latch U45 (8D3), and gates on counter 0 of U64 (8A7).

The 32-bit comparator (see paragraph 4-36) compares the ID read from the disk with the ID of the selected sector. At the end of the ID time, counter 0 times out, pulling the ECC TIME/ line low and initiating the ECC compare (see paragraph 4-37). If the ID and the ECC are valid, bit 6 of the controller status register U27 (12C3) is reset. At the end of ECC time, U36-7 (8A2) pulls the END TIME line high, which resets RD GATE. The 8089 then checks bit 6 of control status register U27 (12C3). If the bit is inactive, the 8089 continues with the write operation. If the ID or ECC are not valid (bit 6 active), the AM ENABLE and RD GATE lines are then reasserted and the controller searches for the next address mark.

To begin the second step of the write operation, the 8089 writes to I/O port 8000H (decoded WDC00/) and enables the write gate (WRT GATE), which in turn activates BUS 0 (11B1), enabling the drive's write circuitry. When counter 0 times out, counter 1 is started. Counter 1 is set for a time interval equivalent to the ECC time plus GAP 2. When counter 1 times out, counter 2 is started and the U41-9 (8C4) is set, activating WRT XFER/. WRT XFER/ enables write buffers U50 and U53 (7C7) and the ECC comparator logic (7AX), and raises the RDY line high indicating to the 8089 that the write buffer is ready to receive data.

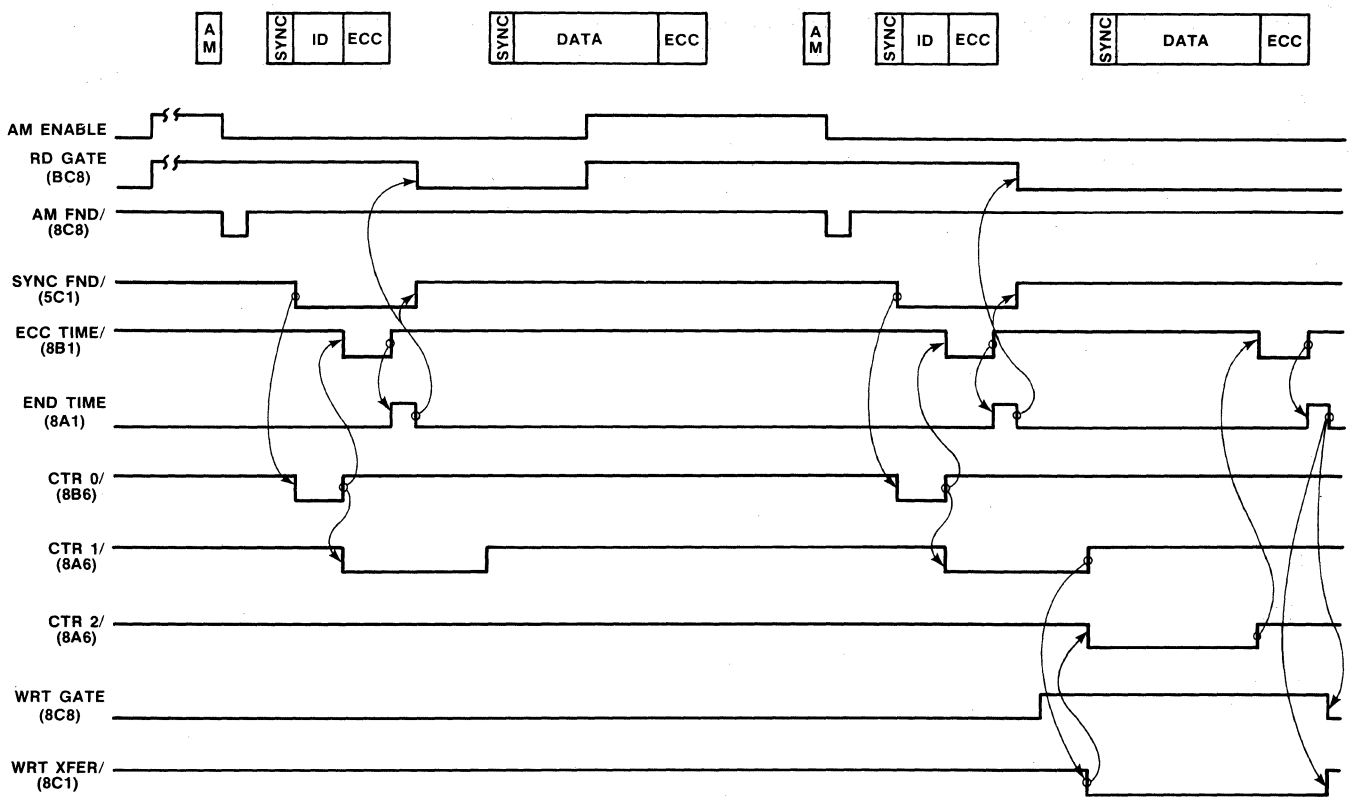


Figure 4-12. Timing Diagram for Write Data

The 8089 then enters its DMA mode to write data from local RAM to the disk (see the discussion of the DMA mode in paragraph 4-30). The controller continues transferring data to the disk in this manner until Counter 2 times out, indicating the end of the data field, and raises the ECC TIME line. With the ECC TIME line activated, the ECC generated during the data transfer is written to the disk. END TIME then terminates the write operation.

4-33. READ DATA TRANSFERS

The read operation is divided into two steps: (1) read sector ID and (2) read data. The reading of the sector ID is performed in the same manner as for the write operation (see figure 4-13).

When the desired sector is located, the RD GATE is again raised to search for the sync byte of the data field. When SYNC FND/ is activated, counter 2 is started through U42-11 (8D5) and U31-6 (8A5), the ECC generator is enabled and the RDY line is activated, initiating the DMA read data transfer mode. Data is then transferred from the disk to local RAM for the duration of counter 2.

When counter 2 times out, ECC TIME is activated. Following ECC TIME, END TIME is raised, terminating the read operation.

4-34. SER/DES LOGIC

The serial/deserialize logic performs two functions: (1) converts parallel data words into a serial string of bits to be sent to the disk drive during a write operation, and (2) converts a serial string of bits into 16-bit words during a read operation. The SER/DES logic is made up of Write Buffer U50 and U53 (7C7), SERializer/DESerializer U51 and U54 (7C5), Read Buffer U52 and U55 (7C4), and Selector U65 (7A7).

During a write operation (WRT XFER/ low), the 8089 IOP writes to I/O port address 8028H. Write I/O port address decoder U32 (5A2) decodes this address and pulls WDC28/ low, clocking the data to be written to the disk (BDAT-0 through BDAT-F) into write buffer U50 and U53 (7C7). A high on load serial register line LDSR (7C6), derived from word clock U35 (8C6) loads the contents of the write buffer (SR-0 through SR-F) into the SER/DES (7C5). Read/write clock R/W CLK-B (7B8) then clocks the

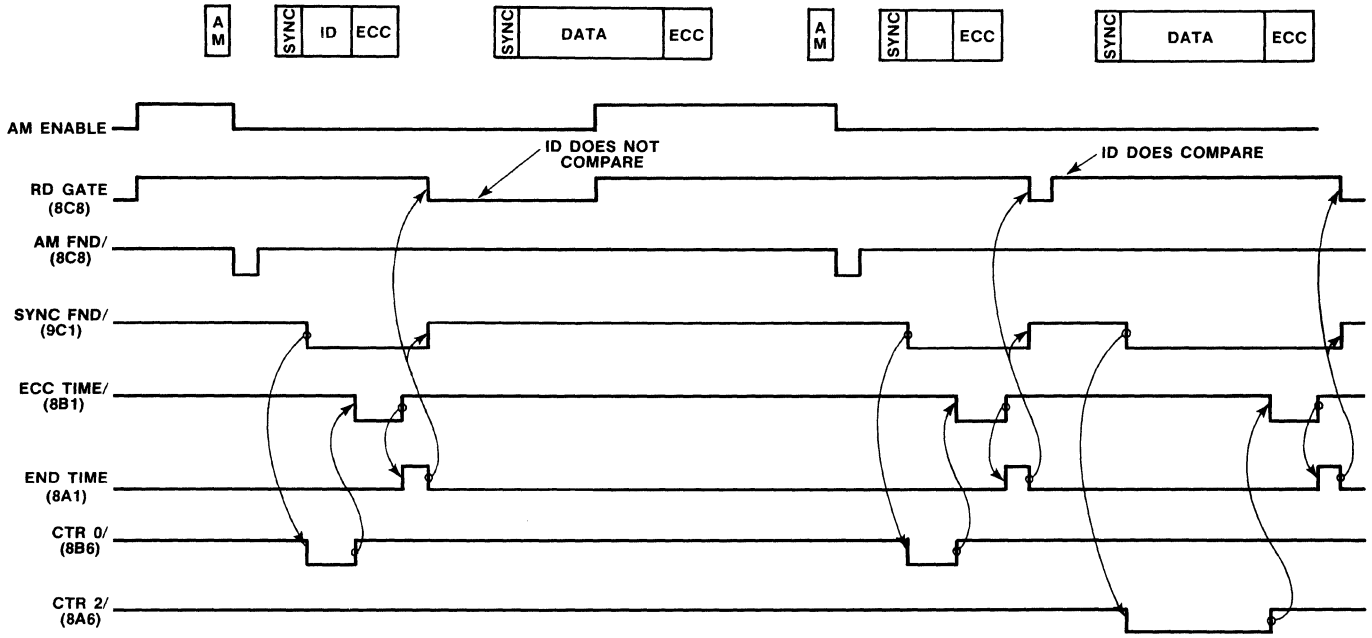


Figure 4-13. Timing Diagram for Read Data

data bit by bit through the QH output of U51 (7D5), and through selector U65 (7A7) to the WRT DATA line. R/W CLK-A clocks the serial data string on WRT DATA through U12 (10C3) to the selected drive.

During a read operation, the R/W CLK-B (10B2) gates the serial data string (RD DATA) from the disk drive through U63 (10B2) and selector U65 (7A7) and into the SI input of U54 (7C5), creating a 16-bit parallel word. Bit ring-0 line BR-0 (7B7) then clocks this word into read buffer U52 and U55 (7C4). BR-0 is derived from word clock U35 (8C6). With the read buffer loaded, the 8089 initiates a read to I/O port address 8028H. Read I/O port address decoder U33 (5B2) decodes this address and pulls RDC28/ low, which clocks the data word from the read buffer onto internal data bus IDAT-0 through IDAT-F.

4-35. SYNC BYTE COMPARATOR LOGIC

The sync byte comparator detects the presence of a sync byte during a read operation and synchronizes word clock U35 (8C6) with the data. A sync byte (always a 19H) is written preceding each sector ID and each data field to indicate to the controller that data to be read is forthcoming (see Figure 3-1).

During a read operation, sync byte decoder U58 and U68 (7B5) monitors the output of the SER/DES, U51 and U54 (7C5). When a 19H is detected, SYNC BYTE/ goes low indicating the presence of the sync byte. SYNC BYTE/ and the next output of R/W CLK-B set the SYNC FND flip-flop, U29 (9C6). SYNC FND activates word clock U35 (8D6), and activates the read/write logic (sheet 8). A further explanation of the sync byte logic can be found in paragraphs 4-31 through 4-33.

4-36. 32-BIT ID COMPARATOR LOGIC

The 32-bit ID comparator logic compares the sector ID of the record being searched for with the sector ID being read from the disk drive. The sector ID is made up of the flags, cylinder number, sector number and head address.

To load the sector ID of the record being searched for into 32-bit ID comparator U19, U37, U20 and U38 (9DX), the 8089 IOP writes to I/O ports 8030H, enabling the WDC30/ and WDC38/ lines, respectively. WDC30/ and WDC38/ initiate the loading of the sector ID into the ID comparator. This loading occurs prior to performing either a read or write data operation. The ID compare operation begins after the sync byte of an ID field has been detected (SYNC FND). R/W CLK-B clocks the ID information, which is stored in the ID comparator, out of U38 (pins 7 and 9) bit by bit. U28 (9D2) compares the serial string of bits with the sector ID from the disk drive (RD-DATA). If the two sector IDs differ, ID no-compare line ID NCMPR/ is activated; if they are the same,

ID NCMPR/ is raised. Selector U65 (7A7) ORs the ID NCMPR/ and the ECC NCMPR/ lines (see paragraph 4-37). The resulting ID-ECC NCMPR/ lines is latched into U29 (9B6). The Q/ output of U29, ID NCMPR-L, is transmitted to bit 6 of status register U27 (12C3). The 8089 IOP then reads the contents of the status register and checks the condition of bit 6. Bit 6 being set high indicates that the record read from the disk was either not the record being searched for or had an ECC error; conversely, bit 6 being set low indicates that the ID field compared and that there was not an ECC error. The 8089 IOP can then read or write the data portion of the record.

4-37. ECC GENERATOR LOGIC

The error checking code (ECC) logic performs two functions: (1) during a write operation, it generates a four byte ECC polynomial that is appended to the ID field (format operation only) and the data field (normal write) of a record (see figure 3-1), (2) during a read operation, it regenerates the ECC polynomial and compares it to the ECC field read from the disk record to ensure that the correct data was read from the drive.

During a write operation, serial data (either an ID field or a data field) is transmitted from the SER/DES (7C5) through selector U65 (7A7) and into the ECC generator through pins 1 and 2 of U98 (7A6), where the ECC polynomial is generated. At the same time a high on the WRT XFER DLYD line enables the serial data to be transmitted through gate U46 (7A7), U66 (7A3) and selector U65 (7A7) to the WRT DATA line, where it is transmitted to the disk. At ECC time (end of data field), WRT XFER DLYD goes low, inhibiting write data from being transferred through gate U46 (7A7). The ECC TIME/ line goes low, causing the ECC polynomial to be written onto the disk through U66 (7A3), U65 (7A7) and the WRT DATA line.

During a read operation, serial data (again either a sector ID or a data field) is read into the ECC generator through selector U65 (7A7) and into the SER/DES through U66 (7A3) and U65. At ECC time, U66 compares the ECC polynomial from the ECC generator bit by bit with the ECC polynomial from the disk and transmits the difference through U65 to the SER/DES for storage in RAM. If the difference is zero, the ID-ECC NCMPR/ line is pulled high indicating the sector ID was error free. (Refer to paragraph 4-36). If the result of the comparison is non-zero, the difference is called the error syndrome. The controller's firmware uses this syndrome to correct errors in a sector ID or data field (if correctable).

4-38. STATUS REGISTER LOGIC

Status register U26 and U27 (12X3) stores status information (see Table 4-7) transmitted to the controller from the selected disk drive and the status of the ID-ECC NCMPR/ line. When the 8089 IOP issues a Read Status command, Read Decode Output (RDC00/) goes low transferring the output of the status register U26 and U27 (12X3) onto the internal bus (IDAT-0 through IDAT-F). The status information is transmitted on IDAT-0 through IDAT-F to the 8089 through transceivers U56 and U57 (4A5). The 8089 analyzes the status information and communicates this status to the host processor, on request, through system memory. Refer to Chapter 3 for more detail on the status information.

4-39. LINE DRIVERS AND RECEIVERS

All the control, data and high speed clock signals transmitted between the controller and the disk drive use differential pair line drivers and receivers. The polarity on these lines is positive true logic i.e., when the + side of the line is more positive than the - side of line, a positive logic "1" is being transmitted.

Table 4-7. Status Register Bits

BITS	FUNCTION
F	INDEX
E	NOT DEFINED
D	NOT DEFINED
C	WRITE PROTECT
B	UNIT 3 SELECTED/
A	UNIT 2 SELECTED/
9	UNIT 1 SELECTED/
8	UNIT 0 SELECTED/
7	0
6	ID NOT COMPARE
5	0
4	DRIVE FAULT
3	SEEK ERROR
2	ON CYLINDER/
1	DRIVE READY/
0	0

CHAPTER 5
SERVICE INFORMATION

5-1. INTRODUCTION

This chapter provides service and repair assistance instructions, service diagrams, a complete electronic parts list for the printed circuit board assembly and a reference to the controller's self diagnostic.

5-2. SERVICE DIAGRAMS

The controller board component locations and schematic diagrams (figures 5-1 and 5-2, respectively) are included at the end of this chapter. Note that these diagrams are intended only for reference; they reflect the iSBC 220 controller design at the time this manual was printed. The schematics and component location diagrams packaged with the controller reflect the design version shipped and thus supercede the diagrams in this manual.

5-3. SERVICE AND REPAIR ASSISTANCE

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

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

a. Date you received the product.

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

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

d. Shipping and billing addresses.

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

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

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

Telephone

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

(800) 528-0595

All other locations: (602) 869-4600

TWX Number

910 - 951 - 1330

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

5-4. SELF DIAGNOSTIC

A self diagnostic is provided with the iSBC 220 controller, stored in the on-board PROM. It performs a go/no-go test of the controller hardware and firmware. If the controller passes the test, it indicates with a high degree of certainty that the controller is operating properly. See the discussion of the diagnostic in Chapter 3 for a description of the program and instructions for initiating the operation.

5-5. REPLACEABLE COMPONENTS

This section contains the information necessary to procure replacement components directly from commercial sources. Component manufacturers have been abbreviated in the parts list with a two to five character code. Table 5-1 cross-references the manufacturer's code with the name and location of the prime commercial source. Table 5-2 lists all the replaceable components on the controller board. Note that the components that are available commercially are listed in the "MFR CODE" column as "COML" and that they are ordered by description (OBD). Procure commercially-available components from a local distributor whenever possible.

Table 5-1. Code for Manufacturers

Mfr. Code	Manufacturer	Location
BECK	Beckman Instruments Inc.	Fullerton, CA
BOUR	Bourns, Inc.	Riverside, CA
CRYST	Crystek	Ft. Meyers, FL
CTSK	CTS Keene, Inc.	Paso Robles, CA
DALE	Dale Electronics	Columbus, NE
FAI	Fairchild Semiconductor	Mt. View, CA
INTEL	Intel	Santa Clara, CA
MOT	Motorola	Phoenix, AZ
SNGMO	Sangamo-Weston, Inc.	Pickens, SC
SPEC	Spectrol Electronics Corp.	City of Industry, CA
SPRG	Sprague Electronic Co.	Adams, MA
3M	3M Co.	St. Paul, MN
TI	Texas Instruments	Dallas, TX
COML	Any Commercial Source; Order By Description (OBD)	

Table 5-2. Controller Board Electrical Parts List

Reference Designation	Description	Mfr. Part No.	Mfr. Code	Qty.
C1, C2, C3, C5	Capacitor, 220μF, Tant, ±10%, 15V	150D226X9015E2	SPRG	4
C4	Capacitor, 0.33μF, Cer. Z5U Axial	OBD	COML	1
C6	Capacitor, 10μF, Tant, ±10%, 20V	150D106X9020B	SPRG	1
C7	Capacitor, 10pF, ±5%	D15-5C100J03	SNGMO	1
C8 through C35	Capacitor, 0.10μF, Cer. Z5U Axial	OBD	COML	28
CR1	Diode, Zener 7.5V ½W	1N5236B	COML	1
CR2	Diode, GP Switching 75V, 5W	1N4148	COML	1
J1	Connector, Header 60 Pin	3372-1302	3M	1
J2, J3	Connector, Header 40 Pin	3432-1302	3M	2
Q1	Transistor, NPN, GP 40	2N3904	TI	1
Q2	Transistor, PNP, GP 40V	2N3906	MOT	1
R1, R2, R7, R11, R13, R21, R29	Resistor, Carb, 1 kΩ, ¼W, ±5%	OBD	COML	7
R3	Resistor, Carb., 2.2 kΩ, ¼W, ±5%	OBD	COML	1
R4, R10, R24, R25	Resistor, Carb., 680 Ω, ¼W, ±5%	OBD	COML	4
R5, R8	Resistor, Carb 2.7 kΩ, ¼W, ±5%	OBD	COML	2
R6, R9, R12, R17, R19, R20, R27, R28	Resistor, Carb., 270 Ω, ¼W, ±5%	OBD	COML	8
R14, R15, R22, R23, R26	Resistor, Carb., 10 kΩ, ¼W, ±5%	OBD	COML	5
R18	Resistor, Carb, 100 kΩ, ¼W, ±5%	OBD	COML	1
R30	Resistor, Wirewound, 3 Ω, 5W, ±5%	CW5	DALE	1
RP1 through RP7	Resistor, Pack, 56 Ω, 8 Pin	4308R-101-56D	BOUR	7
RP8, RP10, RP11 RP12, RP14, RP15	Resistor, Pack, 68 Ω, 8 Pin	4308R-101-68D	BOUR	6
RP9, RP13, RP16 through RP23 RP25 through RP27	Resistor, Pack, 470 Ω, 8 Pin	764-3-R470	BECK	13
RP 24	Resistor, Pack 47 kΩ	764-1-R4.7K	BECK	1
RP28, RP29	Resistor, Carb., 10 kΩ, 8 Pin	764-1-R10K	BECK	2
S1	Switch, 8 Position, DIP	206-08LPST	CTSK	1
S2	Switch, 10 Position, DIP	206-10ST	CTSK	1

Table 5-2. Controller Board Electrical Parts List (Continued)

Reference Designation	Description	Mfr. Part No.	Mfr. Code	Qty.
U1, U2, U36	IC, Hex D Type Flip-Flop	SN74LS174N	TI	3
U3 through U6, U11, U16	IC, Quad Differential Driver	MC3453	MOT	6
U7	IC, Dual J-K Flip-Flop	SN74S112N	TI	1
U8 through U10, U14, U15, U24, U25	IC, Quad Differential Rec.	MC3450	MOT	7
U12	IC, Dual Pos. Edge. Trig. Flip-Flop	SN74S74N	TI	1
U13, U43, U49, U71, U87	IC Quad 2 Input OR	SN74LS32N	TI	5
U17, U29, U45, U60, U63	IC, Dual Pos. Edge Trig. Flip-Flop	SN74LS74N	TI	5
U18	IC, Quad 2 Input NOR	SN74S02N	TI	1
U19, U20, U37, U38	IC, 8 Bit Shift Reg	74LS165N	TI	4
U21	IC, Octal D Type Flip-Flop	SN74LS273N	TI	1
U22, U23, U88, U89, U90	IC, Octal Three State Buffer	SN74LS244N	TI	5
U56, U27, U50, U52, U53, U55	IC, Octal D Type Flip-Flop	SN74LS374N	TI	6
U28	IC, 2 Wide, 3 in, 2 in And-Or-Invert	SN74LS51N	TI	1
U30	IC, Hex Inverter	SN74LS04N	TI	1
U31, U47	IC, Quad 2 Input AND	SN74LS08N	TI	2
U32, U33	IC, 3 to 8 Decoder	SN74LS138	TI	2
U34	IC, Hex Inverter	SN74S04	TI	1
U35	IC, 4 Bit Binary Counter	SN74LS161N	TI	1
U39	IC, Dual 2 to 4 Line Decoder	SN74LS139N	TI	1
U40, U42, U44	IC, Quad 2 Input NAND	SN74LS00	T1	3
U41	IC, Quad R-S Type Latch	SN74LS279	TI	1
U46, U70	IC, TR1 3 Input NAND	SN74LS10N	TI	2
U48	IC, Hex Inverting Buf/Drvr	SN7406N	TI	1
U51, U54	IC, 8 Bit Shift/Storage Register	SN74LS299N	TI	2
U56, U57	IC, Octal Bus Transceiver	8286	INTEL	2
U58	IC, Dual 4 Input NAND	SN74LS20	TI	1
U59	IC, Clock Generator	8284A	INTEL	1
U61	IC, 13 input NAND	SN74S133N	TI	1
U62, U68	IC, Quad 2 Input NOR	SN74LS02N	TI	2
U64	IC, Programmable Counter/Timer	8253-5	INTEL	1
U65	IC, 257 Quad 2:1 MUX	SN74LS257N	TI	1
U66	IC, 9 Bit Parity Generator	SN74S280N	TI	1
U67	IC, Quad 3 State Buffer	SN74LS125N	TI	1
U69	IC, Hex Schmidt Trigger	74LS14	TI	1
U72, U73, U74, U75	IC, Quad 2 Input XNOR OC	SN74LS266	TI	4
U76, U77, U78	IC, Octal Latch, Inverting	8283	INTEL	3
U79	IC, Input/Output Processor	8089	INTEL	1
U80, U81	IC, Octal D Type Latch	SN74LS373N	TI	2
U85	IC, Bus Arbiter	8289	INTEL	1
U86	IC, Bus Controller	8288	INTEL	1
U91, U92, U93	IC, Octal Bus Transceiver, Invert.	8287	INTEL	3
U94, U95, U96, U97	IC, Static RAM	2114A-5	INTEL	4
U98, U99, U100, U101	IC, 8 Bit Shift Register	SN74LS164	TI	4
VR1	Voltage Regulator, -5V	MC7905CT	MOT	1
Y1	Crystal, 15.000 MHz	Type 44 Miniature HC45	CRYST	1

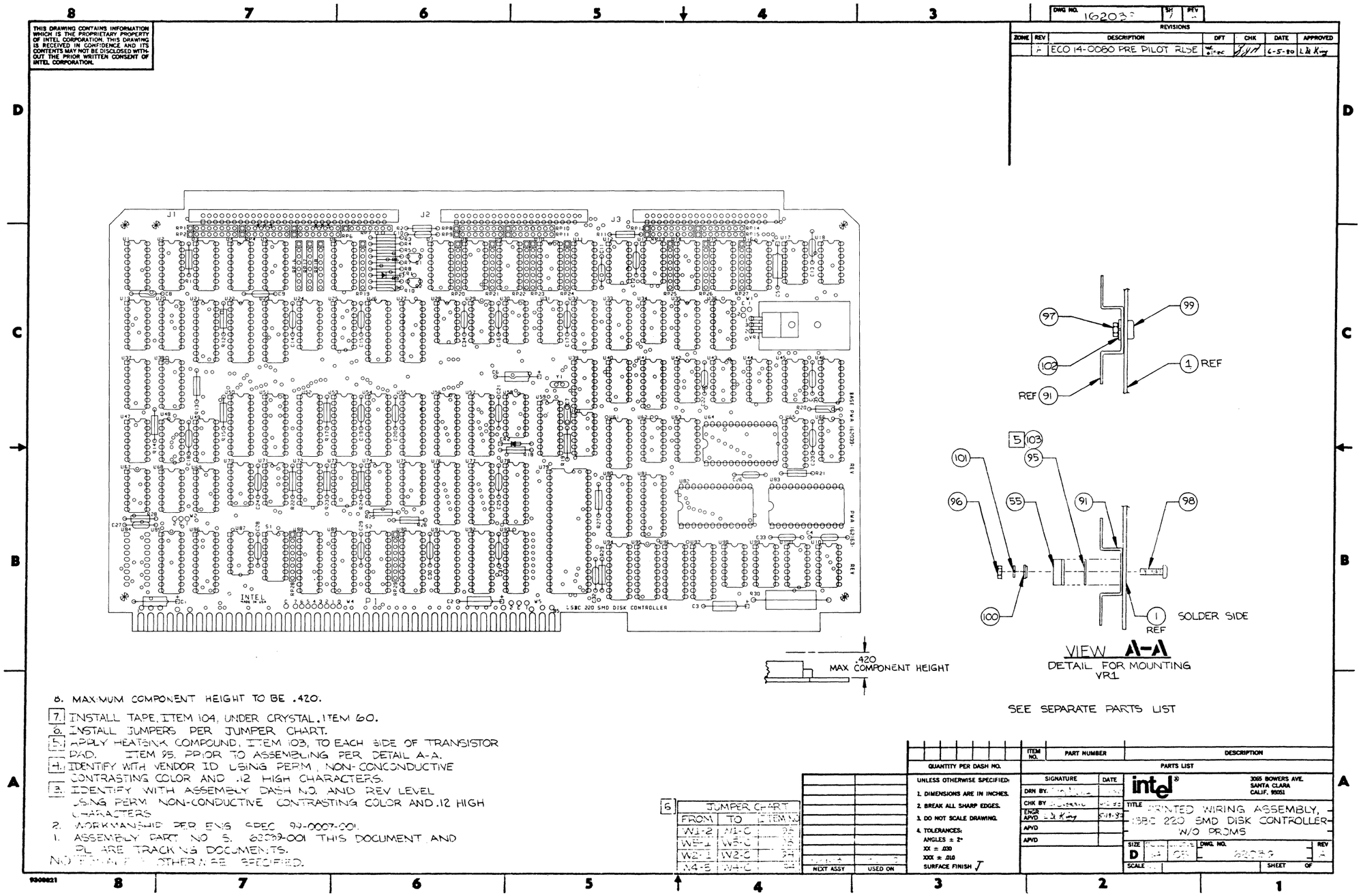


Figure 5-1. iSBC 220 SMD Disk Controller Parts Location Diagram

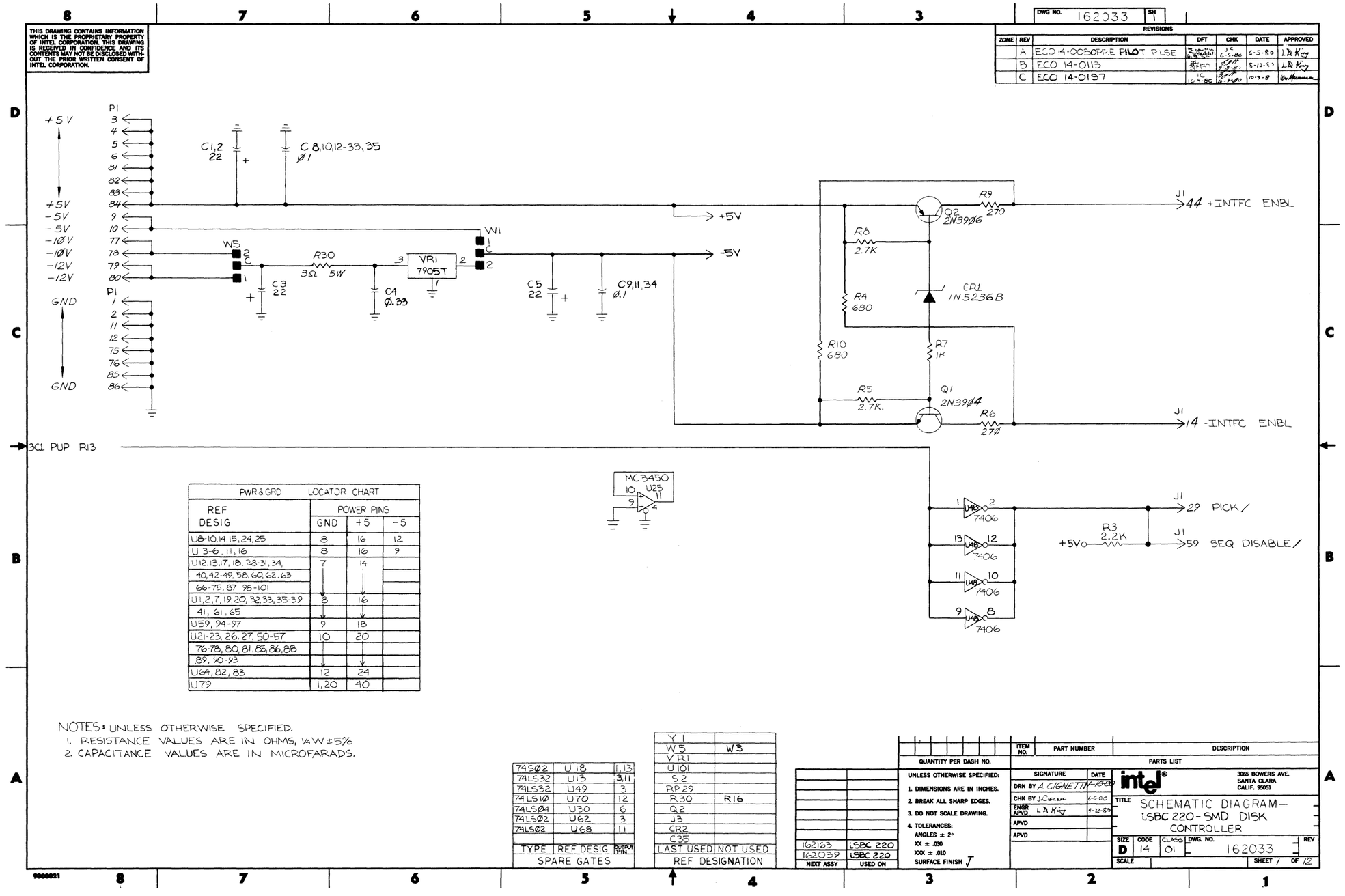


Figure 5-2. iSBC 220 SMD Disk Controller Schematic Diagram (Sheet 1 of 12)

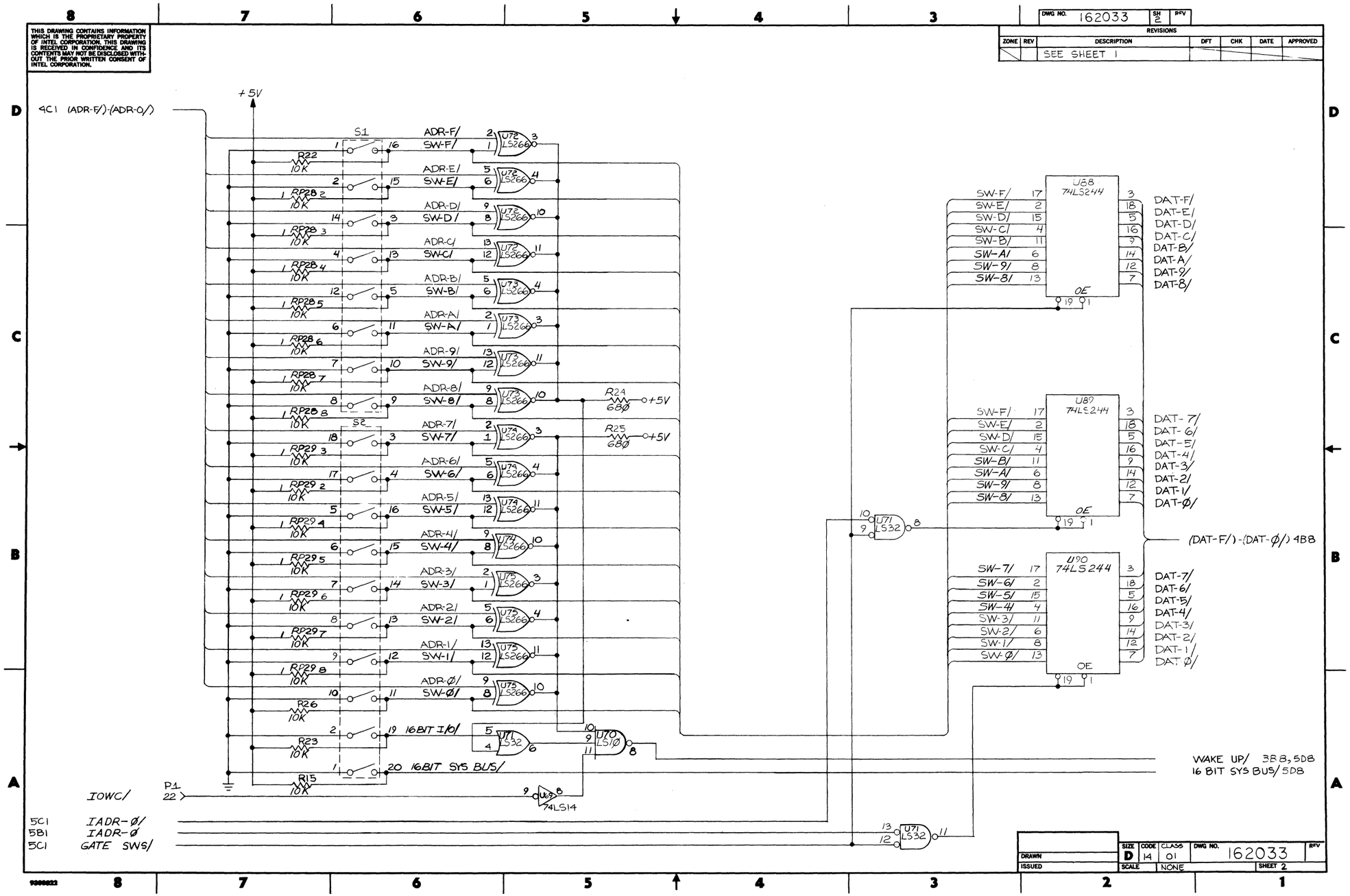


Figure 5-2. iSBC 220 SMD Disk Controller Schematic Diagram (Sheet 2 of 12)

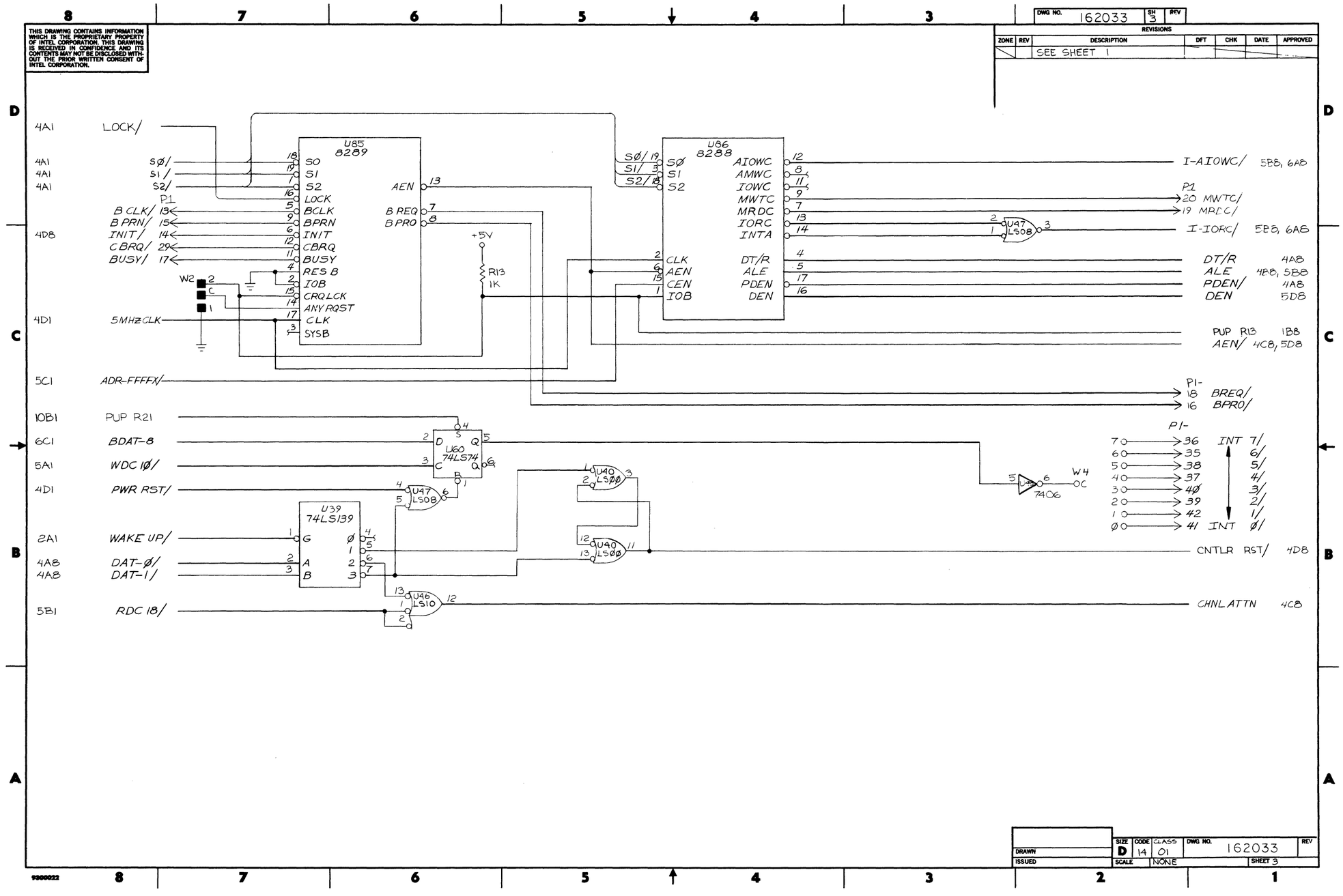


Figure 5-2. iSBC 220 SMD Disk Controller Schematic Diagram (Sheet 3 of 12)

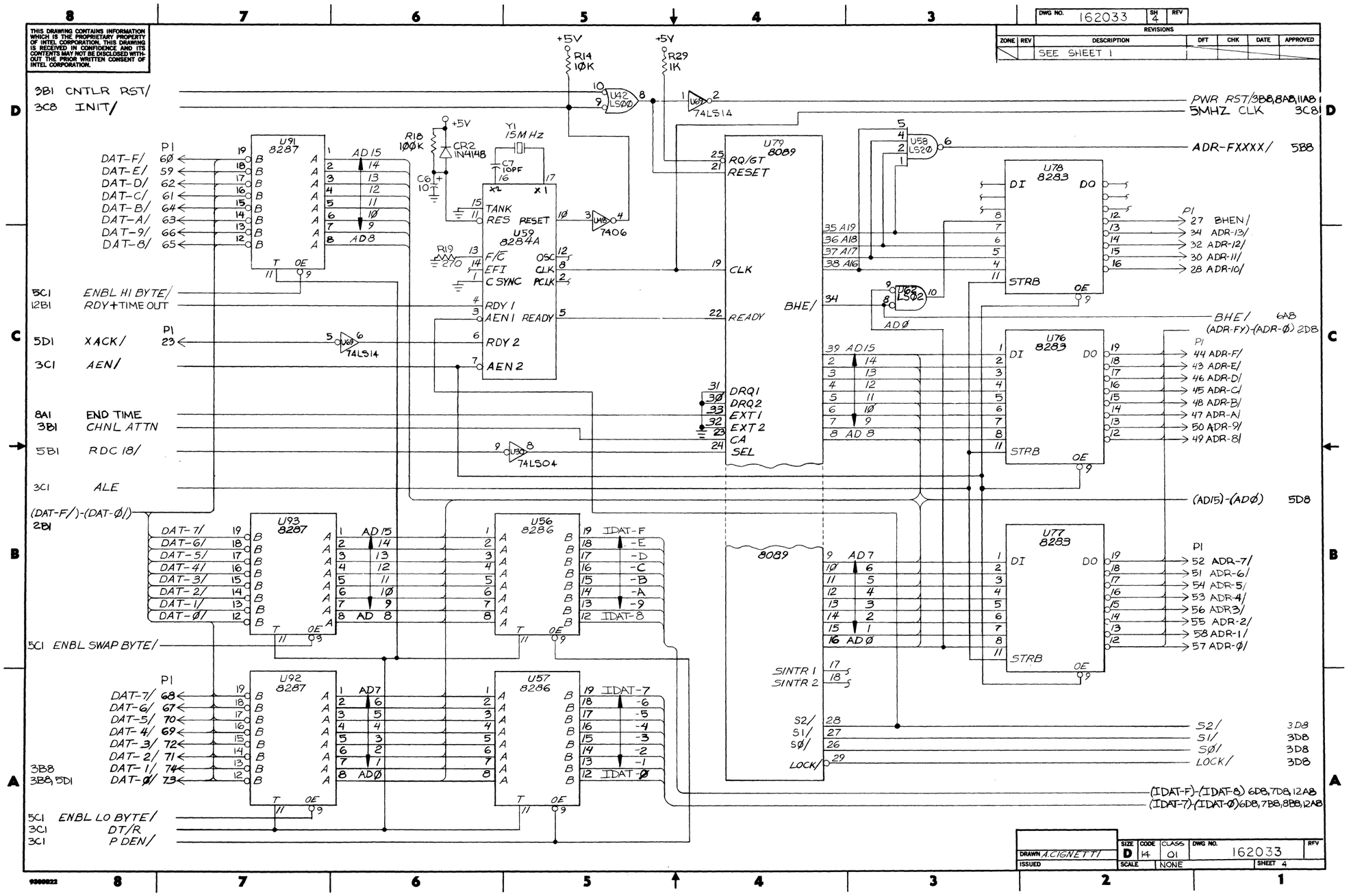


Figure 5-2. iSBC 220 SMD Disk Controller Schematic Diagram (Sheet 4 of 12)

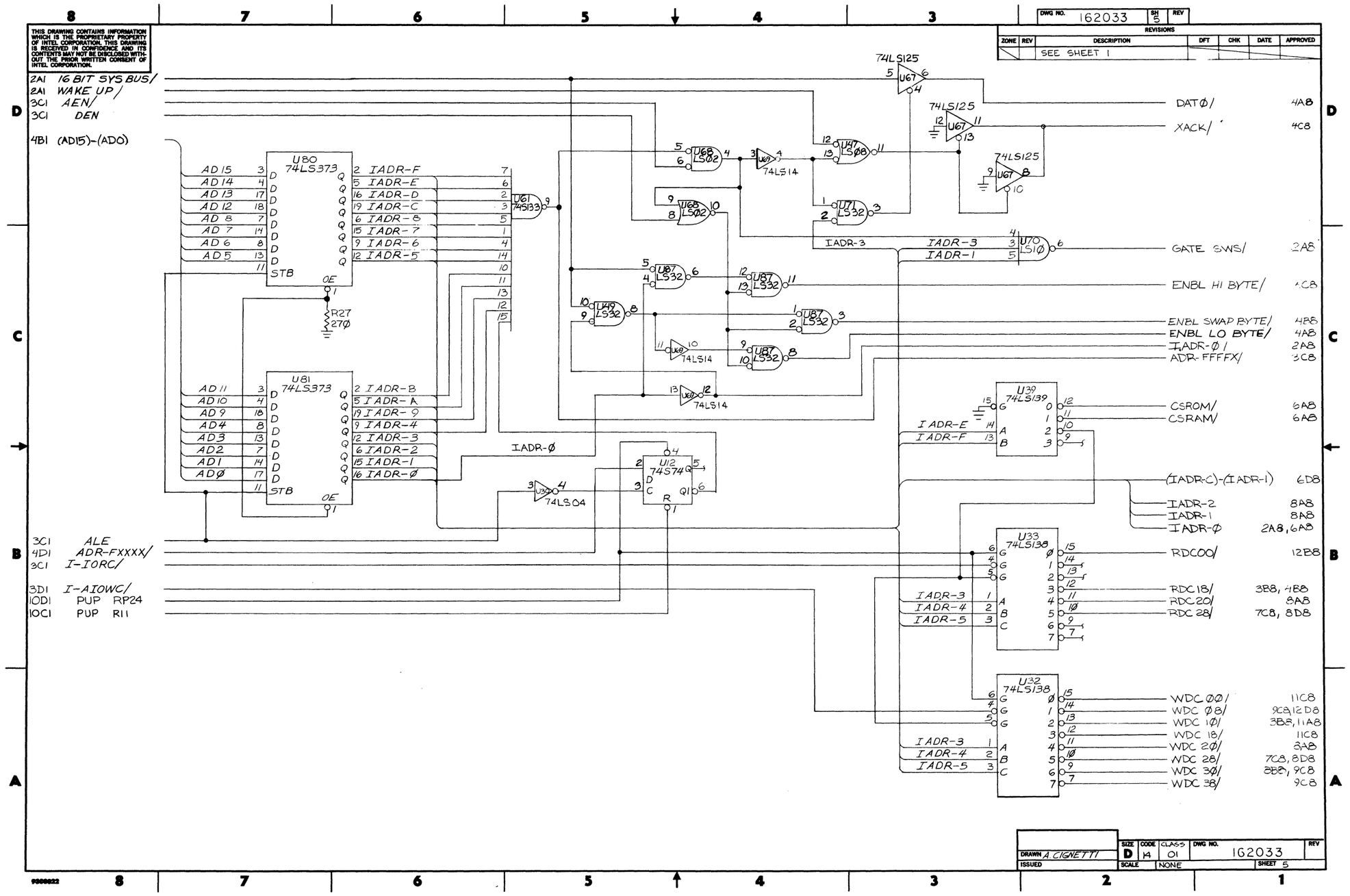


Figure 5-2. iSBC 220 SMD Disk Controller Schematic Diagram (Sheet 5 of 12)

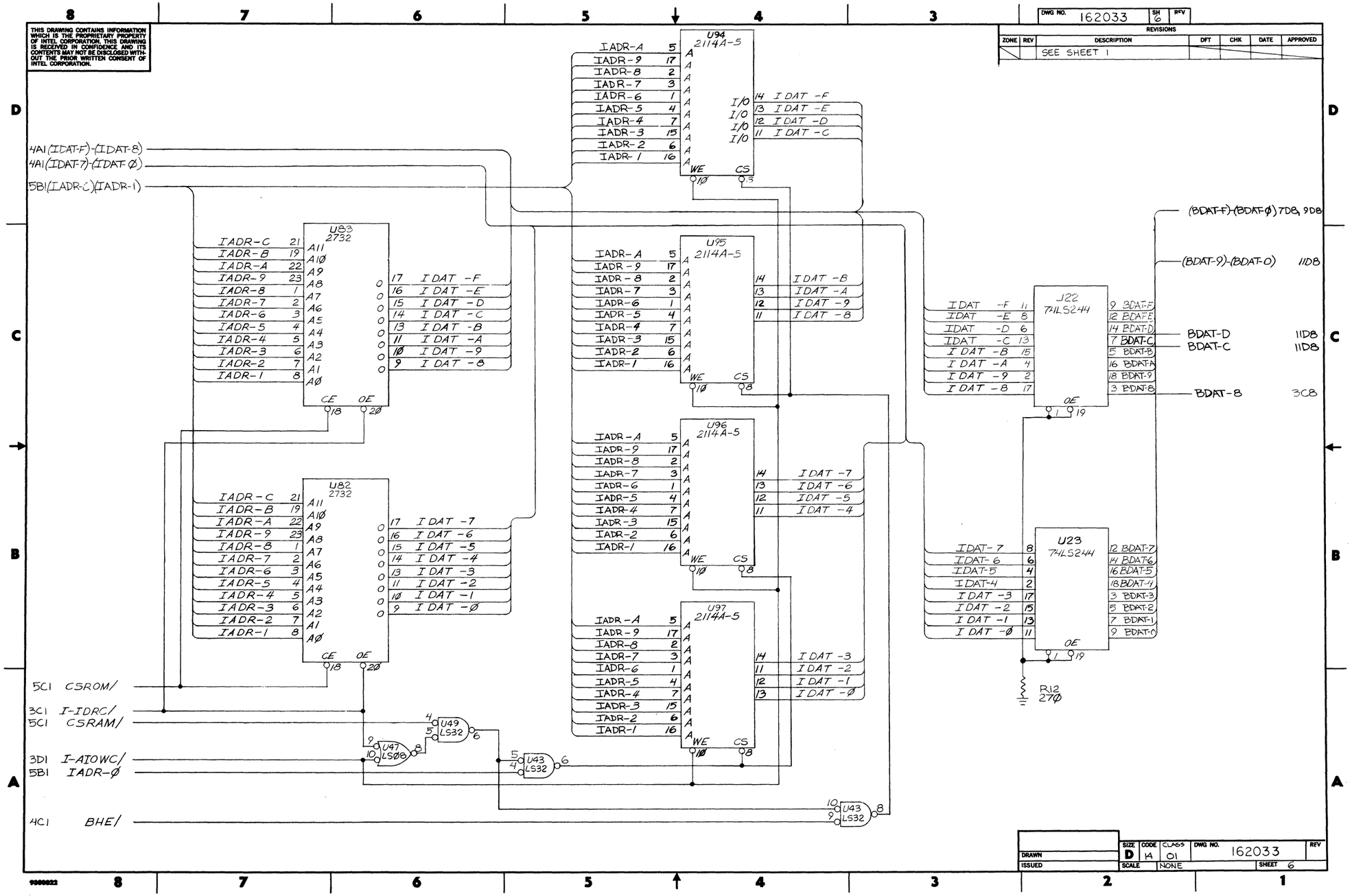


Figure 5-2. iSBC 220 SMD Disk Controller Schematic Diagram (Sheet 6 of 12)

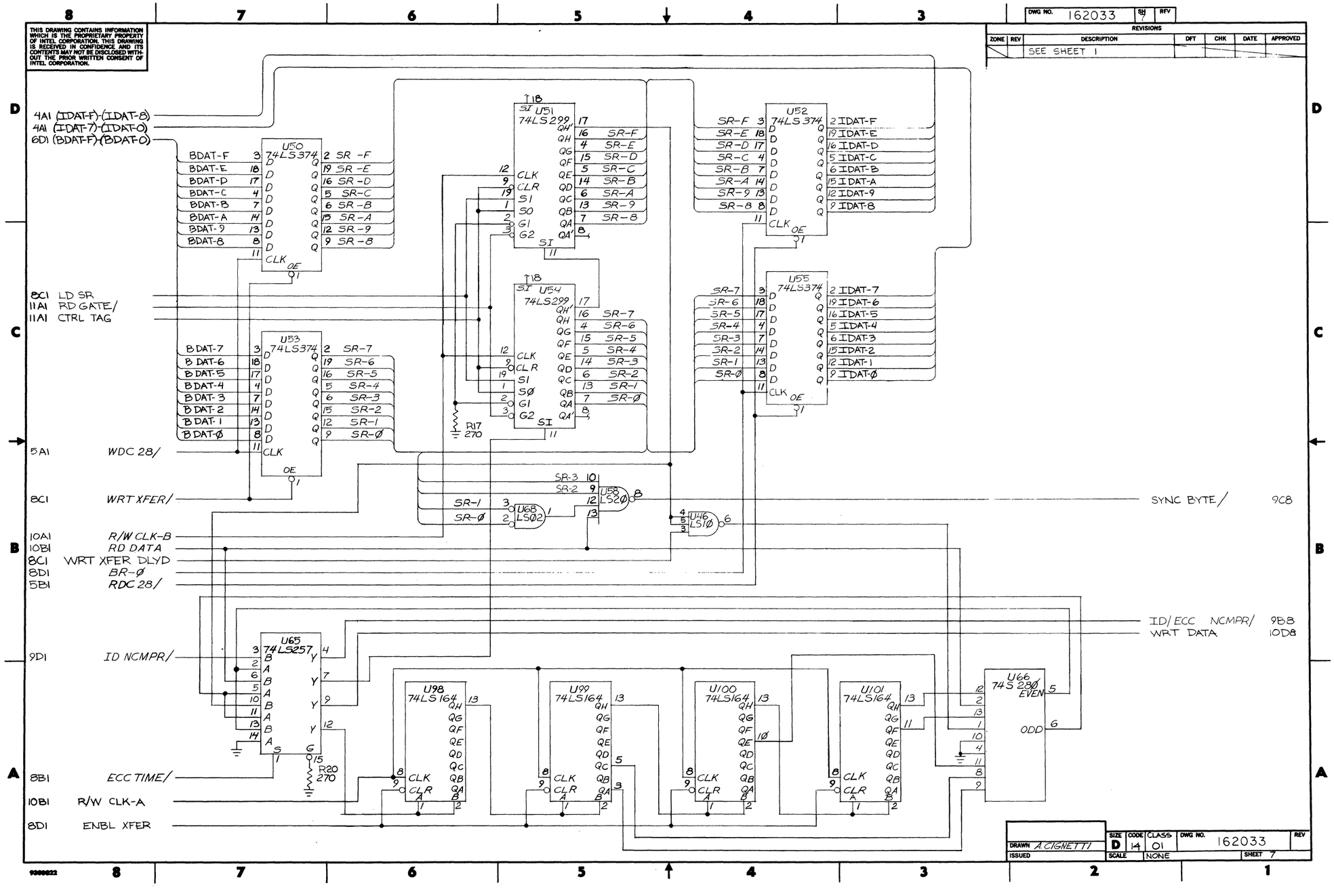


Figure 5-2. iSBC 220 SMD Disk Controller Schematic Diagram (Sheet 7 of 12)

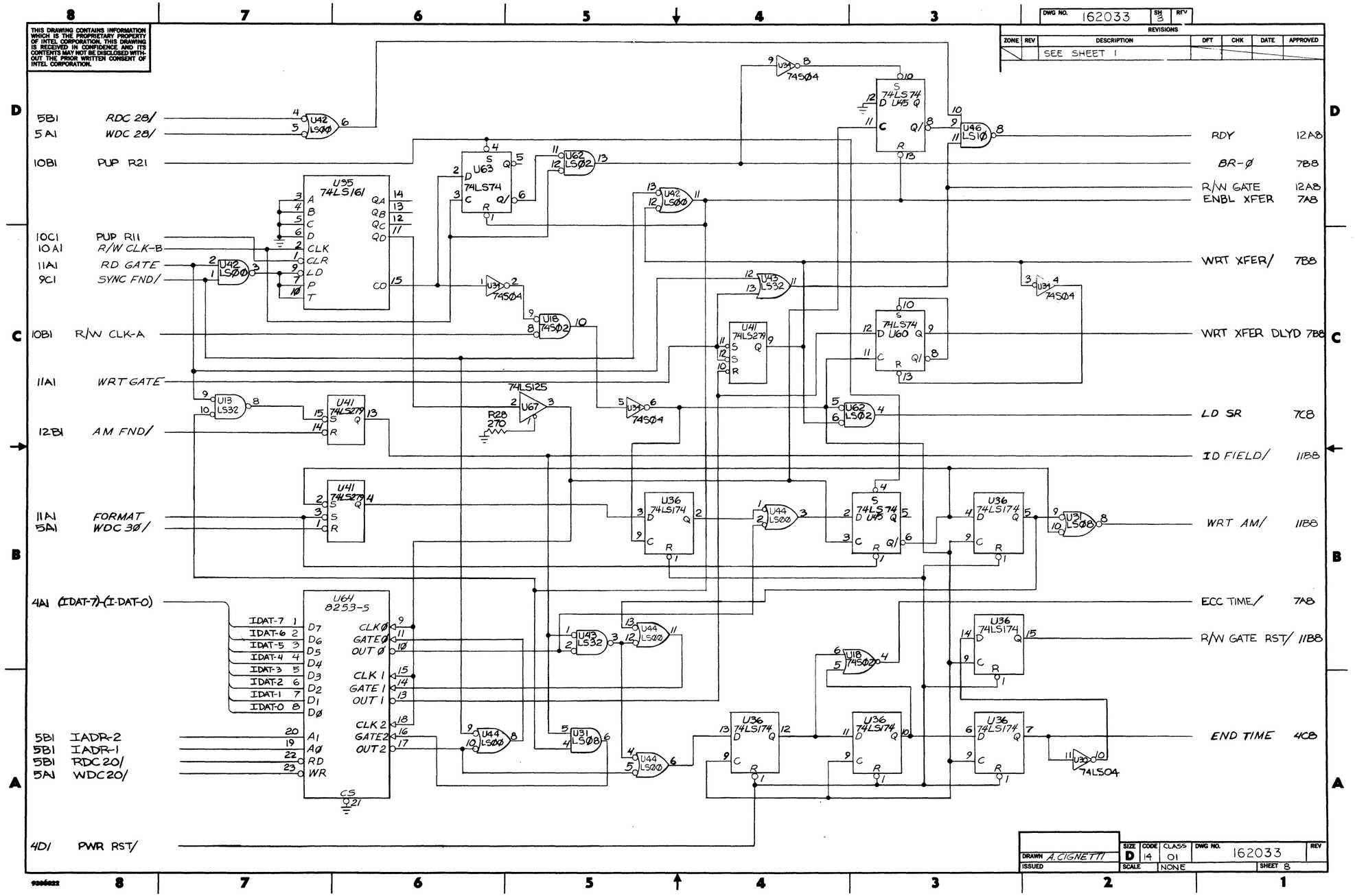


Figure 5-2. iSBC 220 SMD Disk Controller Schematic Diagram (Sheet 8 of 12)

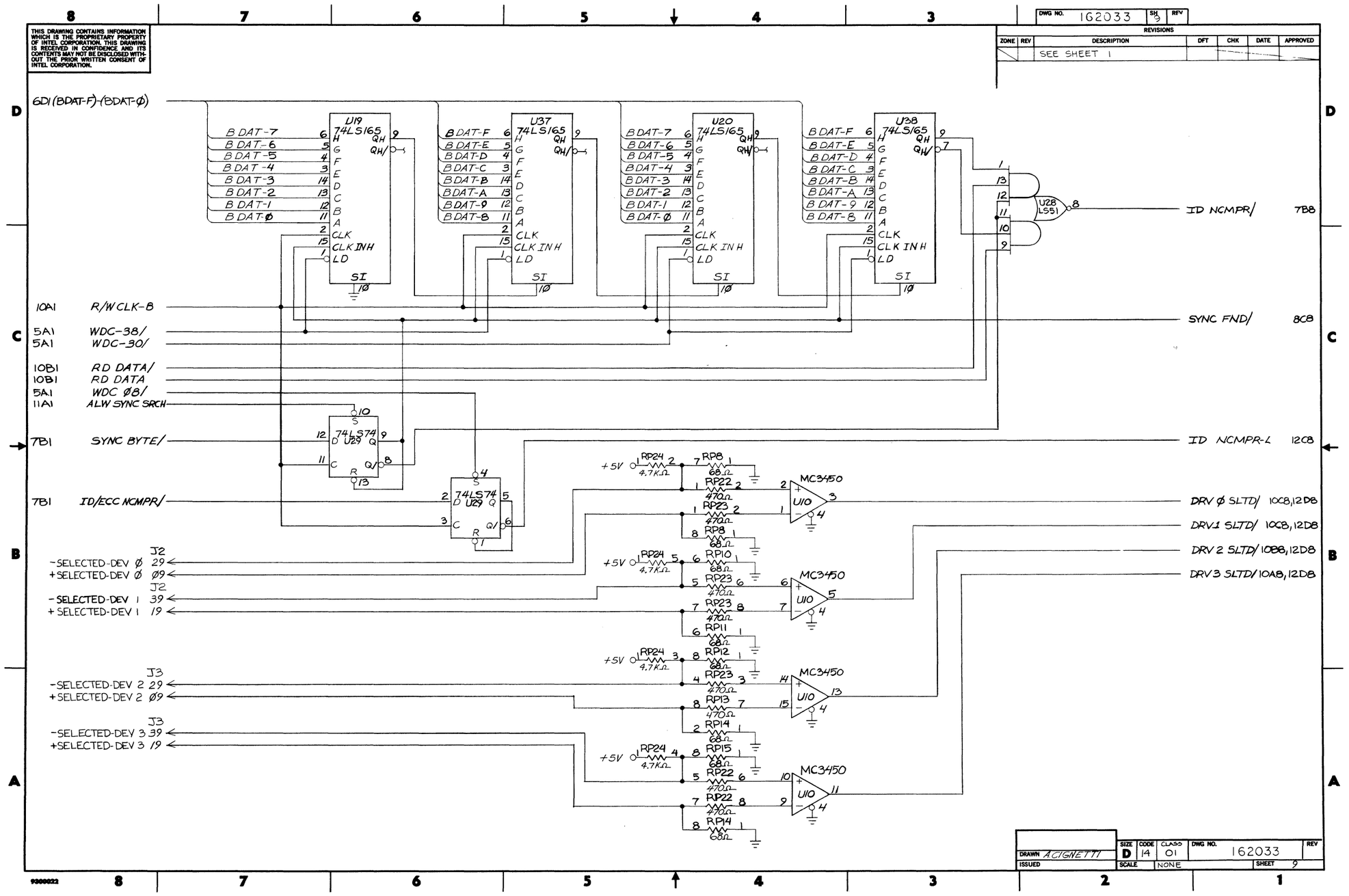


Figure 5-2. iSBC 220 SMD Disk Controller Schematic Diagram (Sheet 9 of 12)

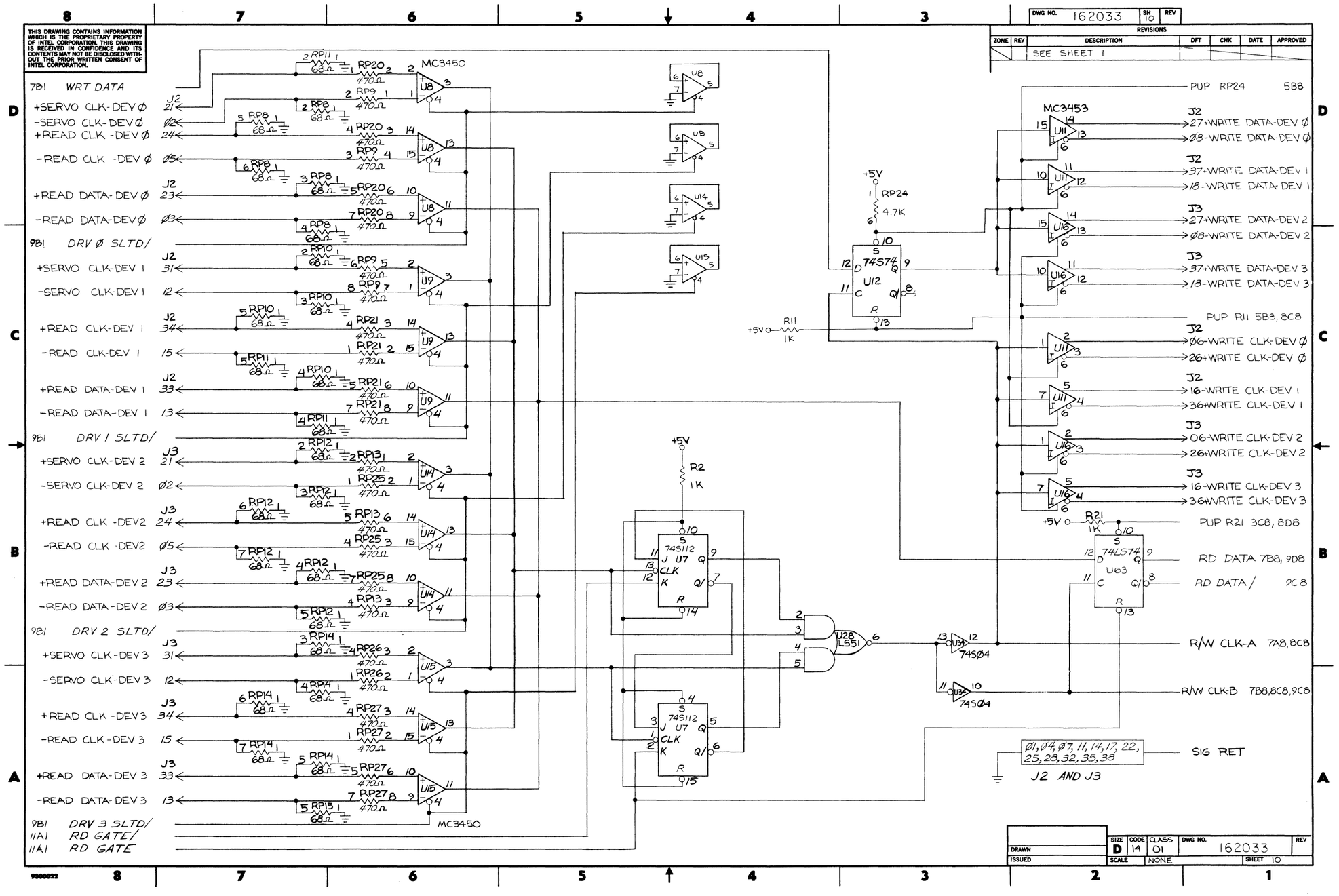


Figure 5-2. iSBC 220 SMD Disk Controller Schematic Diagram (Sheet 10 of 12)

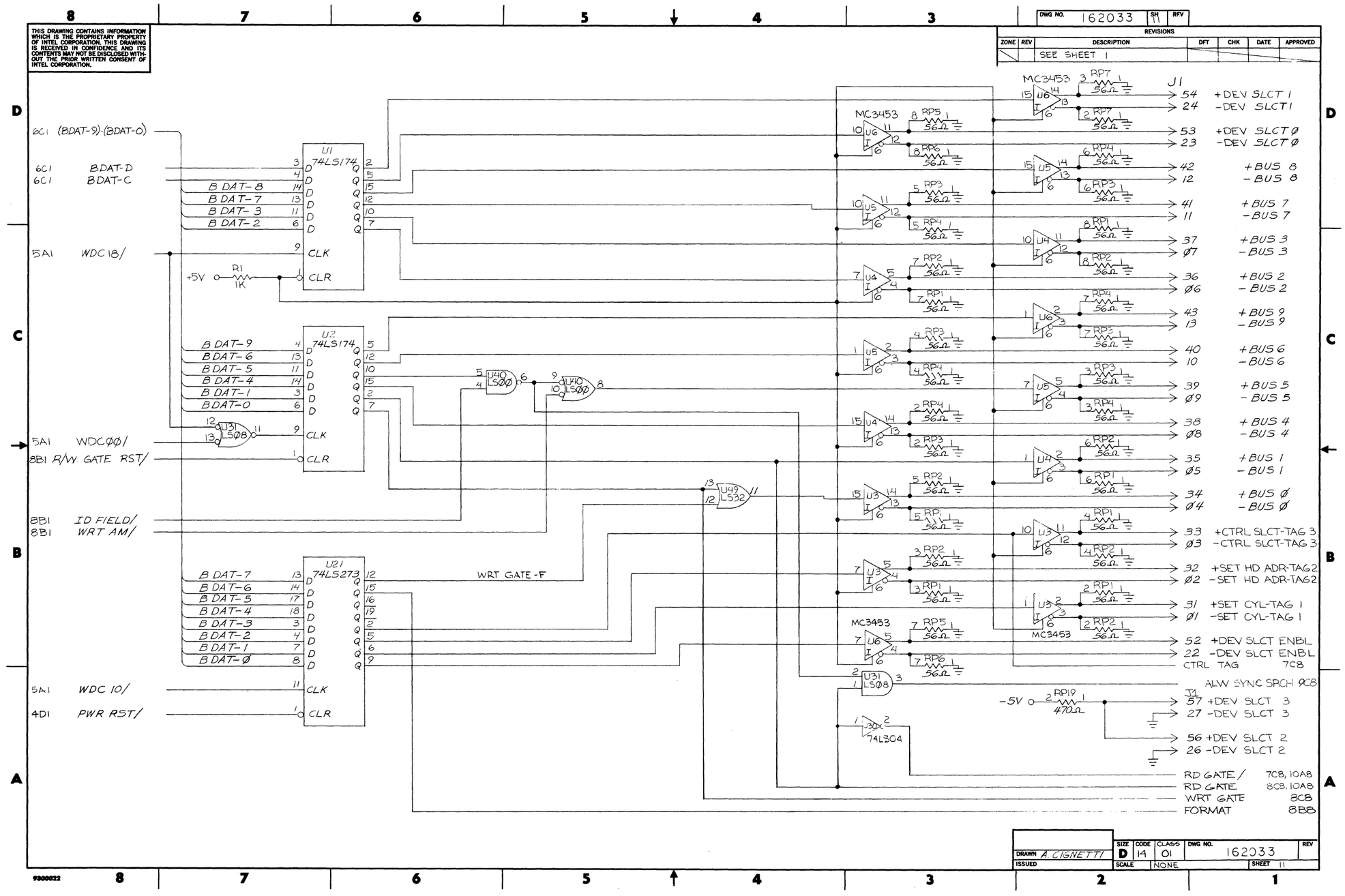


Figure 5-2. iSBC 220 SMD Disk Controller Schematic Diagram (Sheet 11 of 12)

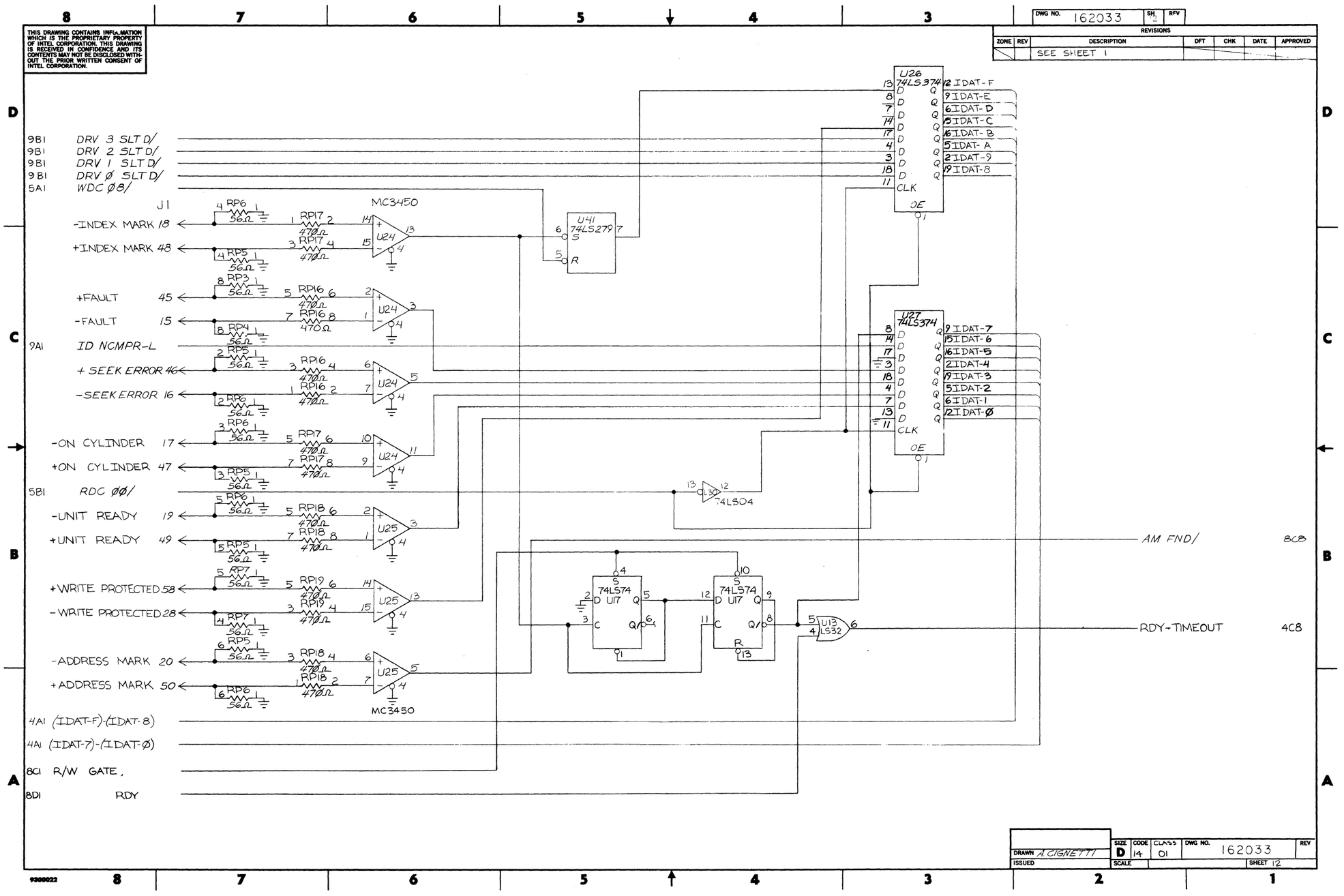


Figure 5-2. iSBC 220 SMD Disk Controller Schematic Diagram (Sheet 12 of 12)

APPENDIX A
EXAMPLE HOST PROCESSOR DISK CONTROL PROGRAM

INTRODUCTION

The following listing provides an example program that a host processor would run to direct data transfer between the host and the iSBC 220 controller. The program is written in MCS-86 Macro Assembler language. It illustrates the data structures that the iSBC 220 controller requires and shows a few simple disk operation drivers.

```
ISIS-II MCS-86 MACRO ASSEMBLER V2.1 ASSEMBLY OF MODULE PRGEXM
NO OBJECT MODULE REQUESTED
ASSEMBLER INVOKED BY:  ASM86 :F0:PRGEXM.SRC DATE(80100611) NOOBJECT

LOC  OBJ                  LINE     SOURCE

                             1     $PAGELENGTH(85) PAGEWIDTH(115) TITLE(iSBC 220 SMD DISK CONTROLLER PROGRAMMING EX
                                   AMPLE) XREF
                             2     ;
                             3     ; ##############################################################
                             4     ; ##                                                          ##
                             5     ; ##     iSBC 220 SMD DISK CONTROLLER PROGRAMMING EXAMPLE     ##
                             6     ; ##                                                          ##
                             7     ; ##############################################################
                             8     ;
                             9     ; THIS PROGRAM ILLUSTRATES THE DATA STRUCTURES REQUIRED BY THE iSBC 220 SMD
                            10     ; DISK CONTROLLER.  A FEW SIMPLE DISK OPERATION DRIVERS ARE ALSO SHOWN.
                            11     ;
                            12     ; THE HARDWARE CONFIGURATION SUPPORTED IS:
                            13     ;
                            14     ;       1.  iSBC 86/12A HOST CPU
                            15     ;       2.  20 BIT SYSTEM MEMORY ADDRESS WIDTH
                            16     ;       3.  16 BIT SYSTEM DATA BUS WIDTH
                            17     ;       4.  16 BIT SYSTEM I/O ADDRESS WIDTH
                            18     ;       5.  iSBC 220
                            19     ;               a.  WAKE UP ADDRESS ( WUA ) AT I/O PORT 0635H
                            20     ;               b.  INTERRUPT 5
                            21     ;               c.  -12 VOLTS INPUT
                            22     ;               d.  RELINQUISH BUS CONTROL ON ANY REQUEST
                            23     ;
                            24     ; FOR (2), PROGRAMMING OF DATA TRANSFERS MUST TAKE THIS INTO ACCOUNT,e.g. THERE
                            25     ;          IS NO WRAPAROUND IN SEGMENTS IF MORE THAN 64K BYTES ARE TRANSFERRED.
                            26     ;
                            27     ; iSBC 220 SWITCH AND JUMPER SETTINGS:
                            28     ;
                            29     ; FOR (3), SWITCH S2-1 IS CLOSED.
                            30     ; FOR (4), SWITCH S2-2 IS CLOSED.
                            31     ; FOR (5a), SWITCHES S1-6,S1-7,S2-5,S2-6,S2-8, AND S2-10 ARE CLOSED, THE
                            32     ;           REMAINING ADDRESS SELECT SWITCHES ARE OPEN.
                            33     ; FOR (5b), W4-C CONNECTS TO W4-5; INTERRUPT VECTORS MUST BE SET UP PROPERLY.
                            34     ; FOR (5c), W1-C CONNECTS TO W1-2, W5-C CONNECTS TO W5-1.
                            35     ; FOR (5d), W2-C CONNECTS TO W2-2.
                            36     ;
                            37 +1  $INCLUDE(COMBLK.SRC)
                        =1  38 +1  $EJECT TITLE(iSBC 220 COMMUNICATION BLOCKS)
```

```
LOC  OBJ                  LINE     SOURCE

                    =1      39     ; ---------------------------------
                    =1      40     ; |                               |
                    =1      41     ; |     COMMUNICATION BLOCKS      |
                    =1      42     ; |                               |
                    =1      43     ; ---------------------------------
                    =1      44     ;
                    =1      45     ; =============
                    =1      46     ;   I.  SCB
                    =1      47     ; =============
                    =1      48     ;
                    =1      49     ;       THE SCB TELLS THE 8089 ON THE iSBC 220 THE WIDTH OF THE 8089's LOCAL
                    =1      50     ;       BUS AND POINTS TO THE CCB.
                    =1      51     ;
                    =1      52     ;       ************************************************************************
                    =1      53     ;       *  THE MEMORY ADDRESS OF THE SCB IS EQUAL TO THE I/O WAKE-UP ADDRESS  *
                    =1      54     ;       *              ( WUA ) OF THE iSBC 220 MULTIPLIED BY 16.               *
                    =1      55     ;       ************************************************************************
                    =1      56     ;
  0635              =1      57             WUA     EQU     0635H           ; WAKE-UP ADDRESS I/O PORT NUMBER
                    =1      58     ;
----                =1      59     SCBSEG  SEGMENT AT WUA                  ; PUTS SCB AT ADDRESS 06350H
                    =1      60     ;
0000                =1      61     SCB     LABEL   FAR
0000 01             =1      62     SOC     DB      01H                     ; TELL 8089 IT IS ON A 16 BIT LOCAL BUS
0001 00             =1      63             DB      00H                     ; RESERVED
0002 0000----     R =1      64     CCBPTR  DD      CCB                     ; POINTER (SEGMENT + OFFSET) TO CCB
                    =1      65     ;
----                =1      66     SCBSEG  ENDS
                    =1      67     ;
                    =1      68     ; ==============
                    =1      69     ;   II.  CCB
                    =1      70     ; ==============
                    =1      71     ;
                    =1      72     ;       THIS BLOCK CONTAINS THE CONTROL BYTES, BUSY FLAGS, AND POINTERS TO THE
                    =1      73     ;       STARTING ADDRESSES OF THE CHANNEL PROGRAMS FOR THE 8089.
                    =1      74     ;
----                =1      75     CCBSEG  SEGMENT                         ; CCB MUST BE CONTIGUOUS
                    =1      76     ;
0000                =1      77     CCB     LABEL   FAR
0000 01             =1      78     CCW1    DB      01H                     ; START CH. 1 PROGRAM IN LOCAL MEMORY
0001 00             =1      79     BSYFLG1 DB      00H                     ; CH. 1 BUSY FLAG
0002 0400----     R =1      80     CH1PTR  DD      CH1PC                   ; POINTER TO FIFTH BYTE OF CIB, WHICH
                    =1      81                                             ;  CONTAINS STARTING ADDRESS OF CH. 1
                    =1      82                                             ;  FIRMWARE PROGRAM
0006 0000           =1      83             DW      0000H                   ; RESERVED
0008 01             =1      84     CCW2    DB      01H                     ; START CH. 2 PROGRAM IN LOCAL MEMORY
0009 00             =1      85     BSYFLG2 DB      00H                     ; CH. 2 BUSY FLAG
000A 0E00----     R =1      86     CH2PTR  DD      CH2PC                   ; POINTER TO LAST WORD OF CCB, WHICH
                    =1      87                                             ;  CONTAINS STARTING ADDRESS OF CH. 2
                    =1      88                                             ;  FIRMWARE PROGRAM
000E                =1      89     CH2PC   LABEL   FAR
000E 0400           =1      90             DW      0004H                   ; STARTING ADDRESS OF CH. 2 PROGRAM
                    =1      91     ;
----                =1      92     CCBSEG  ENDS
                    =1      93     ;
                    =1      94 +1  $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                     =1     95     ; ===============
                     =1     96     ;   III.  CIB
                     =1     97     ; ===============
                     =1     98     ;
                     =1     99     ;       THIS BLOCK CONTAINS GENERAL PURPOSE COMMAND AND STATUS BYTES, SEMA-
                     =1    100     ;       PHORES, AND POINTERS TO ALLOW THE USE OF THE iSBC 220 IN A MULTI-
                     =1    101     ;       PROCESSOR/MULTI-PROCESSING SYSTEM.
                     =1    102     ;
----                 =1    103     CIBSEG  SEGMENT                         ; CIB MUST BE CONTIGUOUS
                     =1    104     ;
0000                 =1    105     CIB     LABEL   FAR
0000 00              =1    106     CIBCMD  DB      00H                     ; CIB COMMAND BYTE NOT USED BY iSBC 220
0001 00              =1    107     OPSTS   DB      00H                     ; CIB STATUS BYTE IS USED BY iSBC 220
0002 00              =1    108     CMDSEM  DB      00H                     ; COMMAND BYTE SEMAPHORE
0003 00              =1    109     STSSEM  DB      00H                     ; STATUS BYTE SEMAPHORE
0004                 =1    110     CH1PC   LABEL   FAR
0004 00000000        =1    111             DD      0000H                   ; STARTING ADDRESS OF CH. 1 PROGRAM
0008 0000            =1    112     IOPBOFF DW      OFFSET IOPB             ; POINTER TO IOPB
000A ----          R =1    113     IOPBSG  DW      IOPBSEG
000C 00000000        =1    114             DD      0000H                   ; RESERVED
                     =1    115     ;
----                 =1    116     CIBSEG  ENDS
                     =1    117     ;
                     =1    118     ; ===============
                     =1    119     ;   IV.  IOPB
                     =1    120     ; ===============
                     =1    121     ;
                     =1    122     ;       THIS BLOCK CONTAINS THE DEVICE DEPENDENT CONTROL INFORMATION FOR THE
                     =1    123     ;       iSBC 220 CONTROLLER.
                     =1    124     ;
----                 =1    125     IOPBSEG SEGMENT                         ; IOPB MUST BE CONTIGUOUS
                     =1    126     ;
0000                 =1    127     IOPB    LABEL   FAR
0000 00000000        =1    128             DD      0000H                   ; RESERVED
0004 00000000        =1    129     ACTCNT  DD      0000H                   ; ACTUAL TRANSFER COUNT (32 BIT INTEGER)
0008 0200            =1    130     DEVCOD  DW      0002H                   ; DEVICE CODE (= 0002H FOR iSBC 220)
000A 00              =1    131     UNIT    DB      00H                     ; UNIT NUMBER (0 <= UNIT <= 3)
000B 00              =1    132     FUNC    DB      00H                     ; FUNCTION CODE (0 <= FUNCTION <= 0FH)
000C 0000            =1    133     MODIFY  DW      0000H                   ; MODIFIER WORD
000E 0000            =1    134     CYLNDR  DW      0000H                   ; CYLINDER NUMBER
0010 00              =1    135     HEAD    DB      00H                     ; HEAD NUMBER
0011 00              =1    136     SECTOR  DB      00H                     ; SECTOR NUMBER
0012 0000            =1    137     BUFOFF  DW      0000H                   ; POINTER TO DATA BUFFER
0014 0000            =1    138     BUFSEG  DW      0000H
0016 00000000        =1    139     REQCNT  DD      0000H                   ; REQUESTED TRANSFER COUNT (INTEGER)
001A 00000000        =1    140             DD      0000H                   ; RESERVED
                     =1    141     ;
----                 =1    142     IOPBSEG ENDS
                           143     ;
                           144 +1  $INCLUDE(INITBL.SRC)
                     =1    145 +1  $EJECT TITLE(DISK DRIVE INITIALIZATION TABLES)
```

```
LOC  OBJ                  LINE     SOURCE

                     =1    146      ; ------------------------------------------------------
                     =1    147      ; |                                                    |
                     =1    148      ; |     DISK DRIVE INITIALIZATION PARAMETER TABLES     |
                     =1    149      ; |                                                    |
                     =1    150      ; ------------------------------------------------------
                     =1    151      ;
                     =1    152      ;       THIS SEGMENT CONTAINS THE DRIVE CONFIGURATION DATA TABLES THAT ARE USED
                     =1    153      ;       BY THE INITIALIZATION ROUTINE.  THEY MUST BE MODIFIED TO REFLECT THE
                     =1    154      ;       PARTICULAR DRIVES BEING USED WITH THE iSBC 220 SHD DISK CONTROLLER.
                     =1    155      ;
                     =1    156      ; - IF A DRIVE IS NOT PRESENT, ITS INITIALIZATION TABLE MUST BE ALL ZEROES.
                     =1    157      ;
                     =1    158      ;    ..........................................................
                     =1    159      ;    |  BYTES PER SECTOR   |    MAXIMUM SECTORS PER TRACK    |
                     =1    160      ;    |---------------------+---------------------------------|
                     =1    161      ;    |        128          |            108                  |
                     =1    162      ;    |        256          |            64                   |
                     =1    163      ;    |        512          |            35                   |
                     =1    164      ;    |       1024          |            18                   |
                     =1    165      ;    ..........................................................
                     =1    166      ; ------------------------------------------------------------------------------
                     =1    167      ;
----                 =1    168      INITBLSEG       SEGMENT
                     =1    169      ;
                     =1    170      ; DRIVE #0 --- CARTRIDGE MODULE DRIVE (CMD) (32MB: 16MB FIXED, 16MB REMOVABLE)
                     =1    171      ;
0000 3703            =1    172              DW      823                     ; NUMBER OF CYLINDERS
0002 01              =1    173              DB      1                       ; NUMBER OF FIXED READ/WRITE SURFACES
0003 01              =1    174              DB      1                       ; NUMBER OF REMOVABLE R/W SURFACES
0004 23              =1    175              DB      35                      ; NUMBER OF SECTORS PER TRACK
0005 0002            =1    176              DW      512                     ; NUMBER OF BYTES PER SECTOR
0007 05              =1    177              DB      5                       ; NUMBER OF ALTERNATE CYLINDERS
                     =1    178      ;
                     =1    179      ; DRIVE #1 --- MINI-MODULE DRIVE (MMD) (80MB WINCHESTER: ALL FIXED)
                     =1    180      ;
0008 3703            =1    181              DW      823                     ; NUMBER OF CYLINDERS
000A 05              =1    182              DB      5                       ; NUMBER OF FIXED READ/WRITE SURFACES
000B 00              =1    183              DB      0                       ; NUMBER OF REMOVABLE R/W SURFACES
000C 12              =1    184              DB      18                      ; NUMBER OF SECTORS PER TRACK
000D 0004            =1    185              DW      1024                    ; NUMBER OF BYTES PER SECTOR
000F 06              =1    186              DB      6                       ; NUMBER OF ALTERNATE CYLINDERS
                     =1    187      ;
                     =1    188      ; DRIVE #2 --- NONEXISTENT
                     =1    189      ;
0010 0000            =1    190              DW      0000H                   ; NUMBER OF CYLINDERS
0012 00              =1    191              DB      00H                     ; NUMBER OF FIXED READ/WRITE SURFACES
0013 00              =1    192              DB      00H                     ; NUMBER OF REMOVABLE R/W SURFACES
0014 00              =1    193              DB      00H                     ; NUMBER OF SECTORS PER TRACK
0015 0000            =1    194              DW      0000H                   ; NUMBER OF BYTES PER SECTOR
0017 00              =1    195              DB      00H                     ; NUMBER OF ALTERNATE CYLINDERS
                     =1    196      ;
                     =1    197      ; DRIVE #3 --- NONEXISTENT
                     =1    198      ;
0018 0000            =1    199              DW      0000H                   ; NUMBER OF CYLINDERS
001A 00              =1    200              DB      00H                     ; NUMBER OF FIXED READ/WRITE SURFACES
001B 00              =1    201              DB      00H                     ; NUMBER OF REMOVABLE R/W SURFACES
001C 00              =1    202              DB      00H                     ; NUMBER OF SECTORS PER TRACK
001D 0000            =1    203              DW      0000H                   ; NUMBER OF BYTES PER SECTOR
001F 00              =1    204              DB      00H                     ; NUMBER OF ALTERNATE CYLINDERS
                     =1    205      ;
----                 =1    206      INITBLSEG       ENDS
                           207      ;
                           208 +1   $INCLUDE(DATSEG.SRC)
                     =1    209 +1   $EJECT TITLE(DATA SEGMENT)
```

```
LOC  OBJ                  LINE     SOURCE

                     =1    210      ; -------------------------
                     =1    211      ; |                       |
                     =1    212      ; |     DATA SEGMENT      |
                     =1    213      ; |                       |
                     =1    214      ; -------------------------
                     =1    215      ;
----                 =1    216      DATASEG SEGMENT
                     =1    217      ;
                     =1    218      ;       THIS SEGMENT CONTAINS VARIOUS DATA THAT ARE USED BY THE iSBC 220 DRIVER
                     =1    219      ;       SOFTWARE.
                     =1    220      ;
                     =1    221      ; - THE FLAGS ARE SET BY THE INTERRUPT SERVICE ROUTINE, AND ARE COPIES OF THE
                     =1    222      ;       CIB STATUS POSTED BY THE iSBC 220.  THE ROUTINES THAT USE THE FLAGS ARE
                     =1    223      ;       RESPONSIBLE FOR CLEARING THEM AFTER USE.
                     =1    224      ;-------------------------------------------------------------------------------
                     =1    225      ;
                     =1    226              PUBLIC  OPCMP,SKCMP,PKCHG,ERRSTS
                     =1    227      ;
                     =1    228      ;       OPERATION COMPLETE FLAGS
                     =1    229      ;
0000                 =1    230      OPCMP   LABEL   BYTE
0000 00              =1    231      OPCMP0  DB      00H                     ; OPERATION COMPLETE ON UNIT 0
0001 00              =1    232      OPCMP1  DB      00H                     ; OPERATION COMPLETE ON UNIT 1
0002 00              =1    233      OPCMP2  DB      00H                     ; OPERATION COMPLETE ON UNIT 2
0003 00              =1    234      OPCMP3  DB      00H                     ; OPERATION COMPLETE ON UNIT 3
                     =1    235      ;
                     =1    236      ;       SEEK COMPLETE FLAGS
                     =1    237      ;
0004                 =1    238      SKCMP   LABEL   BYTE
0004 00              =1    239      SKCMP0  DB      00H                     ; SEEK COMPLETE ON UNIT 0
0005 00              =1    240      SKCMP1  DB      00H                     ; SEEK COMPLETE ON UNIT 1
0006 00              =1    241      SKCMP2  DB      00H                     ; SEEK COMPLETE ON UNIT 2
0007 00              =1    242      SKCMP3  DB      00H                     ; SEEK COMPLETE ON UNIT 3
                     =1    243      ;
                     =1    244      ;       PACK CHANGE FLAGS
                     =1    245      ;
0008                 =1    246      PKCHG   LABEL   BYTE
0008 00              =1    247      PKCHG0  DB      00H                     ; PACK CHANGE ON UNIT 0
0009 00              =1    248      PKCHG1  DB      00H                     ; PACK CHANGE ON UNIT 1
000A 00              =1    249      PKCHG2  DB      00H                     ; PACK CHANGE ON UNIT 2
000B 00              =1    250      PKCHG3  DB      00H                     ; PACK CHANGE ON UNIT 3
                     =1    251      ;
                     =1    252      ;       ERROR STATUS BLOCK
                     =1    253      ;       (LOADED FROM CONTROLLER BY ERROR HANDLER)
                     =1    254      ;
000C 0000            =1    255      ERRSTS  DW      0000H                   ; ERROR STATUS WORD
000E 00              =1    256      SFERST  DB      00H                     ; SOFT ERROR STATUS BYTE
000F 0000            =1    257      DESCYL  DW      0000H                   ; DESIRED CYLINDER
0011 00              =1    258      DESHD   DB      00H                     ; DESIRED HEAD
0012 00              =1    259      DESSEC  DB      00H                     ; DESIRED SECTOR
0013 0000            =1    260      ACTCYL  DW      0000H                   ; ACTUAL CYLINDER + FLAG BITS
0015 00              =1    261      ACTHD   DB      00H                     ; ACTUAL HEAD
0016 00              =1    262      ACTSEC  DB      00H                     ; ACTUAL SECTOR
0017 00              =1    263      NMRTRY  DB      00H                     ; NUMBER OF RETRIES MADE
                     =1    264      ;
                     =1    265      ;       LAST OPERATION COMPLETE BYTE
                     =1    266      ;       (COPIED FROM CIB BY WAIT220)
                     =1    267      ;
0018 00              =1    268      LSTSTS  DB      00H
                     =1    269      ;
0019 90              =1    270              EVEN
                     =1    271      ;
001A                 =1    272      ENDDAT  LABEL   FAR                     ; END OF DATA SEGMENT
                     =1    273      ;
----                 =1    274      DATASEG ENDS
                           275      ;
                           276 +1   $INCLUDE(USER.SRC)
                     =1    277 +1   $EJECT TITLE(SYSTEM DEPENDENT INITIALIZATION)
```

```
LOC  OBJ                  LINE     SOURCE

                     =1    278      ; ----------------------------------------
                     =1    279      ; |                                      |
                     =1    280      ; |     SYSTEM DEPENDENT INITIALIZATION  |
                     =1    281      ; |                                      |
                     =1    282      ; ----------------------------------------
                     =1    283      ;
                     =1    284      ;       THIS ROUTINE SETS UP THE INTERRUPT VECTOR FOR AN iSBC 86/12A CPU
                     =1    285      ;       RUNNING UNDER THE iSBC 957A INTERFACE/EXECUTION PACKAGE.
                     =1    286      ;
                     =1    287      ; - THE 8259 INTERRUPT CONTROLLER AND OTHER INITIALZATIONS ARE PERFORMED
                     =1    288      ;       BY THE iSBC 957A FIRMWARE.
                     =1    289      ; ------------------------------------------------------------------------------
                     =1    290      ;
                     =1    291      ;       ===========================
                     =1    292      ;       INTERRUPT VECTOR DEFINITION
                     =1    293      ;       ===========================
                     =1    294      ;
  0005               =1    295              INTRPT  EQU     5               ; iSBC 220 INTERRUPT NUMBER
                     =1    296      ;
----                 =1    297      SEG0000 SEGMENT AT 0000H                ; INTERRUPT VECTORS ARE FROM ABSOLUTE
                     =1    298                                              ;     ADDRESSES 00000H TO 00FF0H
                     =1    299      ;
0094                 =1    300              ORG     80H + 4*INTRPT          ; LOCATION OF INTERRUPT VECTOR WITH
                     =1    301                                              ;     iSBC 957A FIRMWARE
0094 0000            =1    302      INTRIP  DW      0000H                   ; - INSTRUCTION POINTER
0096 0000            =1    303      INTRCS  DW      0000H                   ; - CODE SEGMENT
                     =1    304      ;
----                 =1    305      SEG0000 ENDS
                     =1    306      ;
                     =1    307      ;       ================
                     =1    308      ;       STACK ALLOCATION
                     =1    309      ;       ================
                     =1    310      ;
----                 =1    311      STACK   SEGMENT                         ; STACK SEGMENT
                     =1    312      ;
0000 (64             =1    313              DB      64 DUP(00H)             ; ALLOW 64 BYTES FOR STACK
     00
     )
                     =1    314      ;
0040                 =1    315      ENDSTK  LABEL   FAR
                     =1    316      ;
----                 =1    317      STACK   ENDS
                     =1    318      ;
                     =1    319      ;       ========================================
                     =1    320      ;       STACK AND INTERRUPT CONFIGURATION ROUTINE
                     =1    321      ;       ========================================
                     =1    322      ;
----                 =1    323      USERSEG SEGMENT
                     =1    324      ;
                     =1    325              PUBLIC  CONFIG
                     =1    326              ASSUME  DS:SEG0000
                     =1    327      ;
0000                 =1    328      CONFIG  PROC FAR
                     =1    329      ;
0000 FA              =1    330              CLI                             ; DISABLE INTERRUPTS WHILE SETTING UP
0001 B8----        R =1    331              MOV     AX,STACK                ;;; SET UP STACK
0004 8ED0            =1    332              MOV     SS,AX                   ;;;
0006 BC4000          =1    333              MOV     SP,OFFSET ENDSTK        ;;;
0009 B80000          =1    334              MOV     AX,0000H                ;;; GET POINTER TO SEGMENT 0000H
000C 8ED8            =1    335              MOV     DS,AX                   ;;;
000E C706940012 02   =1    336              MOV     INTRIP,OFFSET INT220    ;;; SET UP INTERRUPT VECTOR
0014 C7069600----  R =1    337              MOV     INTRCS,SEG INT220       ;;;
001A E4C2            =1    338              IN      AL,0C2H                 ;;; INPUT INTERRUPT MASK FROM 8259
001C 24DF            =1    339              AND     AL,11011111B            ;;; ENABLE INTERRUPT 5
001E E6C2            =1    340              OUT     0C2H,AL                 ;;; WRITE NEW MASK OUT TO 8259
0020 FB              =1    341              STI                             ;;; ENABLE INTERRUPTS
0021 CC              =1    342              INT     3                       ;;; GO TO MONITOR
                     =1    343      ;
                     =1    344      CONFIG  ENDP
                     =1    345      ;
----                 =1    346      USERSEG ENDS
                           347      ;
----                       348      SBC220DRIVER    SEGMENT
                           349      ;
                           350              ASSUME CS:SBC220DRIVER
                           351      ;
                           352 +1   $INCLUDE(RESET.SRC)
                     =1    353 +1   $EJECT TITLE(CONTROLLER RESET ROUTINE)
```

```
LOC  OBJ                  LINE     SOURCE

                     =1    354      ; ------------------------------------
                     =1    355      ; |                                  |
                     =1    356      ; |     CONTROLLER RESET ROUTINE     |
                     =1    357      ; |                                  |
                     =1    358      ; ------------------------------------
                     =1    359      ;
                     =1    360      ;       RES220 SETS UP THE COMMUNICATION BLOCKS FOR THE iSBC 220, LINKS THEM
                     =1    361      ;       TOGETHER AND GIVES A RESET, CLEAR RESET, CHANNEL ATTENTION SEQUENCE TO
                     =1    362      ;       THE CONTROLLER.  THIS CAUSES THE 8089 ON THE CONTROLLER TO SET UP ITS
                     =1    363      ;       INTERNAL POINTER TO THE CCB BY THREADING DOWN THE LINKS STARTING WITH
                     =1    364      ;       THE SWITCHES ON THE CONTROLLER.  SUBSEQUENT CA's WILL CAUSE THE 8089 TO
                     =1    365      ;       FETCH ITS POINTERS STARTING AT THE CCB.
                     =1    366      ;
                     =1    367      ; - IF THE CH. 1 BUSY FLAG IS NOT CLEARED WITHIN A "REASONABLE" AMOUNT OF TIME,
                     =1    368      ;       THEN THE iSBC 220 IS PROBABLY NOT RESPONDING TO THE CHANNEL ATTENTION.
                     =1    369      ;       ON THE CONTROLLER: CHECK SWITCH SETTINGS; VOLTAGES; RESET, CLEAR RESET,
                     =1    370      ;       CHANNEL ATTENTION SIGNALS; READY INPUT TO 8089; 8089 STATUS LINES; R/W
                     =1    371      ;       STROBES.
                     =1    372      ;
                     =1    373      ; - THE SYSTEM INTERRUPT LOGIC AND VECTORS FOR THE CONTROLLER ARE ASSUMED TO BE
                     =1    374      ;       CONFIGURED BY AN EXTERNAL PROGRAM.
                     =1    375      ;
                     =1    376      ; INPUT DATA:
                     =1    377      ;       NONE
                     =1    378      ;
                     =1    379      ; OUTPUT DATA:
                     =1    380      ;       CARRY FLAG:     = 0 IF RESET OKAY
                     =1    381      ;                       = 1 IF CH. 1 BUSY FLAG NOT RESET (NOT RESPONDING)
                     =1    382      ; ------------------------------------------------------------------------------
                     =1    383      ;
                     =1    384              PUBLIC  RES220
                     =1    385      ;
0000                 =1    386      RES220  PROC    FAR
                     =1    387      ;
0000 50              =1    388              PUSH    AX                      ; SAVE REGISTERS
0001 53              =1    389              PUSH    BX
0002 51              =1    390              PUSH    CX
0003 52              =1    391              PUSH    DX
0004 1E              =1    392              PUSH    DS
                     =1    393      ;
                     =1    394      ;       SET UP LINKS BETWEEN COMMUNICATION BLOCKS
                     =1    395      ;
                     =1    396      ; SCB
                     =1    397      ;
                     =1    398              ASSUME  DS:SCBSEG
0005 B83506          =1    399              MOV     AX,SCBSEG               ; GET POINTER TO SCB
0008 8ED8            =1    400              MOV     DS,AX
000A C706000000100   =1    401              MOV WORD PTR SOC,0001H          ; SET SOC BYTE AND CLEAR RESERVED BYTE
0010 C706020000000   =1    402              MOV WORD PTR CCBPTR,OFFSET CCB  ; SET POINTER TO CCB
0016 C7060400----  R =1    403              MOV WORD PTR CCBPTR+2,SEG CCB
                     =1    404      ;
                     =1    405      ; CCB
                     =1    406      ;       LDS     AX,CCBPTR               ; GET POINTER TO CCB
                     =1    407              ASSUME  DS:CCBSEG
001C C7060000001FF   =1    408              MOV WORD PTR CCW1,0FF01H        ; SET CCW1 AND CH. 1 BUSY FLAG
0022 C706020000400   =1    409              MOV WORD PTR CH1PTR,OFFSET CH1PC; SET POINTER TO FIFTH BYTE OF CIB
0028 C7060400----  R =1    410              MOV WORD PTR CH1PTR+2,SEG CH1PC ;    (HAS STARTING ADDRESS FOR CH. 1)
002E C706080000100   =1    411              MOV WORD PTR CCW2,0001H         ; SET CCW2 AND CLEAR CH. 2 BUSY FLAG
0034 C7060A000E00    =1    412              MOV WORD PTR CH2PTR,OFFSET CH2PC; SET POINTER TO CH. 2 STARTING ADDRESS
003A C7060C00----  R =1    413              MOV WORD PTR CH2PTR+2,SEG CH2PC
0040 C7060E000400    =1    414              MOV WORD PTR CH2PC,0004H        ; SET CH. 2 STARTING ADDRESS
                     =1    415      ;
                     =1    416      ; CIB
                     =1    417      ;
                     =1    418              ASSUME  DS:CIBSEG
0046 B8----        R =1    419              MOV     AX,CIBSEG               ; GET POINTER TO CIB
0049 8ED8            =1    420              MOV     DS,AX
004B C706000000000   =1    421              MOV WORD PTR CIBCMD,0000H       ; CLEAR CIB COMMAND AND CIB STATUS BYTES
0051 C706020000000   =1    422              MOV WORD PTR CMDSEM,0000H       ;    ...AND SEMAPHORES
0057 C706040000000   =1    423              MOV WORD PTR CH1PC,0000H        ; SET CH. 1 STARTING ADDRESS
005D C706080000000   =1    424              MOV     IOPBOFF,OFFSET IOPB     ; SET IOPB POINTER
0063 C7060A00----  R =1    425              MOV     IOPBSG,SEG IOPB
                     =1    426      ;
                     =1    427 +1   $EJECT
```

```
LOC  OBJ                  LINE     SOURCE

                    =1    428      ;       CLEAR OUT DATA SEGMENT
                    =1    429      ;
                    =1    430              ASSUME  DS:DATASEG
0069 B8----       R =1    431              MOV     AX,DATASEG              ; GET POINTER TO DATA SEGMENT
006C 8ED8           =1    432              MOV     DS,AX
006E B90D00         =1    433              MOV     CX,(OFFSET ENDDAT)/2    ; GET COUNT (# WORDS IN DATA SEGMENT)
0071 BB0000         =1    434              MOV     BX,0000H                ; CLEAR INDEX REGISTER
0074 C7070000       =1    435      CLRLP:  MOV WORD PTR [BX],0000H         ; CLEAR NEXT WORD IN DATA SEGMENT
0078 43             =1    436              INC     BX                      ; POINT TO NEXT WORD
0079 43             =1    437              INC     BX
007A E0F8           =1    438              LOOPNE  CLRLP                   ; DONE?
                    =1    439                                              ; NO--CLEAR ANOTHER WORD
                    =1    440                                              ; YES--INITIALIZE COMMUNICATION LINKS
                    =1    441      ;
                    =1    442      ;       OUTPUT RESET/CLEAR RESET/CHANNEL ATTENTION TO CONTROLLER
                    =1    443      ;
007C BA3506         =1    444              MOV     DX,WUA                  ; GET WAKE-UP I/O PORT ADDRESS
007F B002           =1    445              MOV     AL,02H                  ; GET RESET COMMAND BYTE
0081 EE             =1    446              OUT     DX,AL                   ; OUTPUT TO WAKE-UP I/O PORT
0082 B000           =1    447              MOV     AL,00H                  ; GET CLEAR RESET COMMAND BYTE
0084 EE             =1    448              OUT     DX,AL                   ; OUTPUT TO WAKE-UP I/O PORT
0085 B001           =1    449              MOV     AL,01H                  ; GET CHANNEL ATTENTION COMMAND BYTE
0087 EE             =1    450              OUT     DX,AL                   ; OUTPUT  TO WAKE-UP I/O PORT
                    =1    451              ASSUME  DS:CCBSEG
0088 B8----       R =1    452              MOV     AX,CCBSEG               ; GET POINTER TO CCB
008B 8ED8           =1    453              MOV     DS,AX
                    =1    454                                              ; (OTHER IMPLEMENTATIONS OF RES220 COULD
                    =1    455                                              ; INITIALIZE OTHER DEVICES WHILE THE
                    =1    456                                              ; iSBC 220 DOES ITS RESET SEQUENCE HERE)
008D B90010         =1    457              MOV     CX,1000H                ; SET TIME-OUT COUNTER
0090 F8             =1    458              CLC                             ; CLEAR CARRY FLAG
0091 F6060100FF     =1    459      RESLP:  TEST    BSYFLG1,00FFH           ; CHECK CH. 1 BUSY FLAG:
                    =1    460                                              ;    ZERO FLAG = BSYFLG1 & FFH
0096 7403           =1    461              JZ      RESDN                   ; BUSY FLAG CLEARED?
                    =1    462                                              ; YES--RETURN CARRY CLEAR
0098 E0F7           =1    463              LOOPNE  RESLP                   ; NO--DECREMENT COUNTER
                    =1    464                                              ; IF CX = 0, THEN BSYFLG1 NEVER GOT
009A F9             =1    465              STC                             ;    CLEARED, SO SET CARRY FLAG
009B 1F             =1    466      RESDN:  POP     DS                      ; RESTORE REGISTERS
009C 5A             =1    467              POP     DX
009D 59             =1    468              POP     CX
009E 5B             =1    469              POP     BX
009F 58             =1    470              POP     AX
00A0 CB             =1    471              RET                             ; RETURN
                    =1    472      ;
                    =1    473      RES220  ENDP
                          474      ;
                          475 +1   $INCLUDE(INIT.SRC)
                    =1    476 +1   $EJECT TITLE(INITIALIZATION ROUTINE)
```

```
LOC  OBJ                  LINE     SOURCE

                     =1    477      ; ------------------------------------
                     =1    478      ; |                                  |
                     =1    479      ; |      INITIALIZATION ROUTINE      |
                     =1    480      ; |                                  |
                     =1    481      ; ------------------------------------
                     =1    482      ;
                     =1    483      ;       INIT220 INITIALIZES THE iSBC 220 CONTROLLER BY LOADING PERTINENT INFOR-
                     =1    484      ;       MATION ABOUT THE DISK DRIVE(S) ATTACHED.
                     =1    485      ;
                     =1    486      ; - IF A DRIVE THAT IS SPECIFIED AS PRESENT WILL NOT RESPOND, INIT220 RETURNS
                     =1    487      ;       IMMEDIATELY WITH THE CARRY FLAG SET.
                     =1    488      ;
                     =1    489      ; INPUT DATA:
                     =1    490      ;       DISK DRIVE INITIALIZATION TABLES, IN SEGMENT "INITBLSEG".
                     =1    491      ;
                     =1    492      ; OUTPUT DATA:
                     =1    493      ;       CARRY FLAG              = 0 IF CONTROLLER INITIALIZED SUCCESSFULLY
                     =1    494      ;                               = 1 IF INITIALIZATION ERROR
                     =1    495      ; ------------------------------------------------------------------------------
                     =1    496      ;
                     =1    497              PUBLIC  INIT220
                     =1    498              ASSUME  DS:IOPBSEG
                     =1    499      ;
00A1                 =1    500      INIT220 PROC    FAR
                     =1    501      ;
00A1 50              =1    502              PUSH    AX                      ; SAVE REGISTERS
00A2 1E              =1    503              PUSH    DS
00A3 B8----        R =1    504              MOV     AX,IOPBSEG              ; GET POINTER TO IOPB
00A6 8ED8            =1    505              MOV     DS,AX                   ; PUT IN DS REGISTER
00A8 C6060B0000      =1    506              MOV     FUNC,00H                ; SET IOPB FUNCTION BYTE = INITIALIZE
00AD C7060C000000    =1    507              MOV     MODIFY,0000H            ; CLEAR MODIFIER (ENABLE RETRIES AND
                     =1    508                                              ;    INTERRUPT ON COMPLETION)
00B3 C7061400----  R =1    509              MOV     BUFSEG,INITBLSEG        ; PUT INITIALIZATION TABLES' SEGMENT IN
                     =1    510                                              ;   IOPB DATA BUFFER POINTER
00B9 C7061200F8FF    =1    511              MOV     BUFOFF,-8               ; START INITIALIZE WITH UNIT 0
00BF B000            =1    512              MOV     AL,00H                  ; CLEAR UNIT COUNTER
00C1 8306120008      =1    513      INITLP: ADD     BUFOFF,8                ; POINT TO NEXT DRIVE's INITIALIZE TABLE
00C6 A20A00          =1    514              MOV     UNIT,AL                 ; PUT UNIT INTO IOPB
00C9 E8CB00          =1    515              CALL    GO220                   ; DO INITIALIZE
                     =1    516                                              ;    (RETURNS CARRY FLAG SET OR CLEAR)
00CC 7205            =1    517              JC      INITDN                  ; UNIT INITIALIZED?
                     =1    518                                              ; NO--TERMINATE WITH CARRY BIT SET
00CE 40              =1    519              INC     AX                      ; YES--INCREMENT UNIT COUNTER
00CF 3C04            =1    520              CMP     AL,4                    ; CHECK UNIT COUNTER (CLEARS CARRY)
00D1 75EE            =1    521              JNZ     INITLP                  ; LAST DRIVE INITIALIZED?
                     =1    522                                              ; NO--INITIALIZE NEXT DRIVE
00D3 1F              =1    523      INITDN: POP     DS                      ; YES--RESTORE REGISTERS
00D4 58              =1    524              POP     AX
00D5 CB              =1    525              RET                             ; RETURN
                     =1    526      ;
                     =1    527      INIT220 ENDP
                           528      ;
                           529 +1   $INCLUDE(FORMAT.SRC)
                     =1    530 +1   $EJECT TITLE(FORMAT TRACK ROUTINE)
```

```
LOC  OBJ                  LINE     SOURCE

                     =1    531      ; --------------------------------
                     =1    532      ; |                              |
                     =1    533      ; |     FORMAT TRACK ROUTINE     |
                     =1    534      ; |                              |
                     =1    535      ; --------------------------------
                     =1    536      ;
                     =1    537      ;       FMTTRK SETS UP THE IOPB FOR A FORMAT TRACK FUNCTION, AND
                     =1    538      ;       INVOKES THE iSBC 220 CONTROLLER TO PERFORM THE OPERATION.
                     =1    539      ;
                     =1    540      ; INPUT DATA:
                     =1    541      ;       BP + 9  =>  INTERLEAVE FACTOR
                     =1    542      ;       BP + 8  =>  USER DATA BYTE 3
                     =1    543      ;       BP + 7  =>  USER DATA BYTE 2
                     =1    544      ;       BP + 6  =>  USER DATA BYTE 1
                     =1    545      ;       BP + 5  =>  USER DATA BYTE 0
                     =1    546      ;       BP + 4  =>  TYPE OF FORMAT
                     =1    547      ;       BP + 3  =>  HEAD
                     =1    548      ;       BP + 1  =>  CYLINDER
                     =1    549      ;       BP      =>  UNIT
                     =1    550      ;
                     =1    551      ; OUTPUT DATA:
                     =1    552      ;       CARRY FLAG      = 0 IF TRACK FORMATTED SUCCESSFULLY
                     =1    553      ;                       = 1 IF NON-RECOVERABLE ERROR OCCURRED
                     =1    554      ;
                     =1    555      ; - INTERLEAVE FACTOR OF 1 IMPLIES SEQUENTIAL SECTOR NUMBERING.
                     =1    556      ; - USER DATA BYTES 0 - 3 ARE REPLICATED THROUGHOUT THE DATA FIELD.
                     =1    557      ; - INTERLEAVE TYPES:
                     =1    558      ;       00 = NORMAL TRACK
                     =1    559      ;       40 = ALTERNATE TRACK (POINTED TO BY EXACTLY ONE DEFECTIVE TRACK,
                     =1    560      ;               CANNOT SUBSEQUENTLY BE FORMATTED DEFECTIVE)
                     =1    561      ;       80 = DEFECTIVE TRACK (DATA FIELD POINTS TO ALTERNATE TRACK)
                     =1    562      ; - TO SET UP A POINTER TO AN ALTERNATE TRACK, SET:
                     =1    563      ;       USER DATA BYTE 0 = ALTERNATE CYLINDER LOW BYTE
                     =1    564      ;       USER DATA BYTE 1 = ALTERNATE CYLINDER HIGH BYTE
                     =1    565      ;       USER DATA BYTE 2 = ALTERNATE HEAD
                     =1    566      ;       USER DATA BYTE 3 = 00H
                     =1    567      ; ------------------------------------------------------------------------------
                     =1    568      ;
                     =1    569              PUBLIC  FMT220
                     =1    570              ASSUME  DS:IOPBSEG
                     =1    571      ;
00D6                 =1    572      FMT220  PROC    FAR
                     =1    573      ;
00D6 50              =1    574              PUSH    AX                      ; SAVE REGISTERS
00D7 1E              =1    575              PUSH    DS
00D8 B8----        R =1    576              MOV     AX,IOPBSEG              ; GET POINTER TO IOPB
00DB 8ED8            =1    577              MOV     DS,AX
00DD 8A4600          =1    578              MOV     AL,[BP]                 ; GET UNIT NUMBER INTO IOPB
00E0 A20A00          =1    579              MOV     UNIT,AL
00E3 8B4601          =1    580              MOV     AX,[BP+1]               ; GET CYLINDER NUMBER INTO IOPB
00E6 A30E00          =1    581              MOV     CYLNDR,AX
00E9 8A4603          =1    582              MOV     AL,[BP+3]               ; GET HEAD INTO IOPB
00EC A21000          =1    583              MOV     HEAD,AL
00EF 892E1200        =1    584              MOV     BUFOFF,BP               ; GET POINTER TO FORMAT ARGUMENT LIST
00F3 8306120004      =1    585              ADD     BUFOFF,4                ;    INTO DATA BUFFER POINTER
00F8 8C161400        =1    586              MOV     BUFSEG,SS
00FC C6060B0002      =1    587              MOV     FUNC,02H                ; SET FUNCTION = FORMAT
0101 C7060C000000    =1    588              MOV     MODIFY,0000H            ; CLEAR MODIFIER (ALLOW ERROR RECOVERY
                     =1    589                                              ;    AND INTERRUPT ON COMPLETION)
0107 E88D00          =1    590              CALL    GO220                   ; START iSBC 220 AND WAIT FOR DONE
                     =1    591                                              ;    (RETURNS CARRY FLAG SET OR CLEAR)
010A 1F              =1    592      FMTDN:  POP     DS                      ; RESTORE REGISTERS
010B 58              =1    593              POP     AX
010C CA0A00          =1    594              RET     10                      ; RETURN (AND POP INPUT DATA OFF STACK)
                     =1    595      ;
                     =1    596      FMT220  ENDP
                           597      ;
                           598 +1   $INCLUDE(RDWRT.SRC)
                     =1    599 +1   $EJECT TITLE(READ DATA ROUTINE)
```

```
LOC  OBJ                  LINE     SOURCE

                     =1    600      ; ---------------------
                     =1    601      ; |                     |
                     =1    602      ; |     READ DATA       |
                     =1    603      ; |                     |
                     =1    604      ; ---------------------
                     =1    605      ;
                     =1    606      ;       RD220 SETS UP THE IOPB FOR A READ OPERATION, AND
                     =1    607      ;       INVOKES THE iSBC 220 TO PERFORM THE OPERATION.
                     =1    608      ;
                     =1    609      ; INPUT DATA:
                     =1    610      ;       BP + 11 =>  BYTE COUNT HIGH WORD
                     =1    611      ;       BP +  9 =>  BYTE COUNT LOW WORD
                     =1    612      ;       BP +  7 =>  DATA BUFFER SEGMENT
                     =1    613      ;       BP +  5 =>  DATA BUFFER OFFSET
                     =1    614      ;       BP +  4 =>  SECTOR
                     =1    615      ;       BP +  3 =>  HEAD
                     =1    616      ;       BP +  1 =>  CYLINDER
                     =1    617      ;       BP      =>  UNIT
                     =1    618      ;
                     =1    619      ; OUTPUT DATA:
                     =1    620      ;       CARRY FLAG      = 0 IF TRANSFER OCCURRED WITH NO OR RECOVERABLE ERROR
                     =1    621      ;                       = 1 IF UNRECOVERABLE ERROR OCCURRED
                     =1    622      ;       DATA BUFFER     FILLED WITH DATA FROM DISK IF NO UNRECOVERABLE ERROR
                     =1    623      ; ------------------------------------------------------------------------------
                     =1    624      ;
                     =1    625              PUBLIC  RD220
                     =1    626              ASSUME  DS:IOPBSEG
                     =1    627      ;
010F                 =1    628      RD220   PROC    FAR
                     =1    629      ;
010F 50              =1    630              PUSH    AX                      ; SAVE REGISTERS
0110 1E              =1    631              PUSH    DS
0111 B8----        R =1    632              MOV     AX,IOPBSEG              ; GET POINTER TO IOPB
0114 8ED8            =1    633              MOV     DS,AX
0116 8A4600          =1    634              MOV     AL,[BP]                 ; GET UNIT INTO IOPB
0119 A20A00          =1    635              MOV     UNIT,AL
011C 8B4601          =1    636              MOV     AX,[BP+1]               ; GET CYLINDER INTO IOPB
011F A30E00          =1    637              MOV     CYLNDR,AX
0122 8B4603          =1    638              MOV     AX,[BP+3]               ; GET HEAD AND SECTOR INTO IOPB
0125 A31000          =1    639              MOV WORD PTR HEAD,AX
0128 8B4605          =1    640              MOV     AX,[BP+5]               ; GET DATA BUFFER POINTER INTO IOPB
012B A31200          =1    641              MOV     BUFOFF,AX
012E 8B4607          =1    642              MOV     AX,[BP+7]
0131 A31400          =1    643              MOV     BUFSEG,AX
0134 8B4609          =1    644              MOV     AX,[BP+9]               ; GET BYTE COUNT INTO IOPB
0137 A31600          =1    645              MOV WORD PTR REQCNT,AX
013A 8B460B          =1    646              MOV     AX,[BP+11]
013D A31800          =1    647              MOV WORD PTR REQCNT+2,AX
0140 C7060C000000    =1    648              MOV     MODIFY,0000H            ; CLEAR MODIFIER (ENABLE INTERRUPT ON
                     =1    649                                              ;     COMPLETION AND RETRIES)
0146 C6060B0004      =1    650              MOV     FUNC,04H                ; SET FUNCTION = READ DATA
014B E84900          =1    651              CALL    GO220                   ; START FUNCTION AND WAIT FOR COMPLETION
                     =1    652                                              ;    (RETURNS CARRY FLAG SET OR CLEAR)
014E 1F              =1    653              POP     DS                      ; RESTORE REGISTERS
014F 58              =1    654              POP     AX
0150 CA0D00          =1    655              RET     13                      ; POP PARAMETERS OFF STACK AND RETURN
                     =1    656      ;
                     =1    657      RD220   ENDP
                     =1    658      ;
                     =1    659 +1   $EJECT TITLE(WRITE DATA ROUTINE)
```

```
LOC  OBJ                  LINE     SOURCE

                    =1    660      ; ---------------------
                    =1    661      ; |                   |
                    =1    662      ; |     WRITE DATA    |
                    =1    663      ; |                   |
                    =1    664      ; ---------------------
                    =1    665      ;
                    =1    666      ;       WRT220 SETS UP THE IOPB FOR A WRITE OPERATION, AND
                    =1    667      ;       INVOKES THE iSBC 220 TO PERFORM THE OPERATION.
                    =1    668      ;
                    =1    669      ; INPUT DATA:
                    =1    670      ;       BP + 11 =>  BYTE COUNT HIGH WORD
                    =1    671      ;       BP +  9 =>  BYTE COUNT LOW WORD
                    =1    672      ;       BP +  7 =>  DATA BUFFER SEGMENT
                    =1    673      ;       BP +  5 =>  DATA BUFFER OFFSET
                    =1    674      ;       BP +  4 =>  SECTOR
                    =1    675      ;       BP +  3 =>  HEAD
                    =1    676      ;       BP +  1 =>  CYLINDER
                    =1    677      ;       BP      =>  UNIT
                    =1    678      ;
                    =1    679      ;       DATA BUFFER CONTAINS INFORMATION TO BE WRITTEN TO DISK
                    =1    680      ;
                    =1    681      ; OUTPUT DATA:
                    =1    682      ;       CARRY FLAG      = 0 IF TRANSFER OCCURRED WITH NO OR RECOVERABLE ERROR
                    =1    683      ;                       = 1 IF UNRECOVERABLE ERROR OCCURRED
                    =1    684      ; ------------------------------------------------------------------------------
                    =1    685      ;
                    =1    686              PUBLIC  WRT220
                    =1    687              ASSUME  DS:IOPBSEG
                    =1    688      ;
0153                =1    689      WRT220  PROC    FAR
                    =1    690      ;
0153 50             =1    691              PUSH    AX                      ; SAVE REGISTERS
0154 1E             =1    692              PUSH    DS
0155 B8----       R =1    693              MOV     AX,IOPBSEG              ; GET POINTER TO IOPB
0158 8ED8           =1    694              MOV     DS,AX
015A 8A4600         =1    695              MOV     AL,[BP]                 ; GET UNIT INTO IOPB
015D A20A00         =1    696              MOV     UNIT,AL
0160 8B4601         =1    697              MOV     AX,[BP+1]               ; GET CYLINDER INTO IOPB
0163 A30E00         =1    698              MOV     CYLNDR,AX
0166 8B4603         =1    699              MOV     AX,[BP+3]               ; GET HEAD AND SECTOR INTO IOPB
0169 A31000         =1    700              MOV WORD PTR HEAD,AX
016C 8B4605         =1    701              MOV     AX,[BP+5]               ; GET DATA BUFFER POINTER INTO IOPB
016F A31200         =1    702              MOV     BUFOFF,AX
0172 8B4607         =1    703              MOV     AX,[BP+7]
0175 A31400         =1    704              MOV     BUFSEG,AX
0178 8B4609         =1    705              MOV     AX,[BP+9]               ; GET BYTE COUNT INTO IOPB
017B A31600         =1    706              MOV WORD PTR REQCNT,AX
017E 8B460B         =1    707              MOV     AX,[BP+11]
0181 A31800         =1    708              MOV WORD PTR REQCNT+2,AX
0184 C7060C000000   =1    709              MOV     MODIFY,0000H            ; CLEAR MODIFIER (ENABLE INTERRUPT ON
                    =1    710                                              ;     COMPLETION AND RETRIES)
018A C6060B0006     =1    711              MOV     FUNC,06H                ; SET FUNCTION = WRITE DATA
018F E80500         =1    712              CALL    GO220                   ; START iSBC 220 AND WAIT FOR DONE
                    =1    713                                              ;    (RETURNS WITH CARRY SET OR CLEAR)
0192 1F             =1    714              POP     DS                      ; RESTORE REGISTERS
0193 58             =1    715              POP     AX
0194 CA0D00         =1    716              RET     13                      ; POP PARAMETERS OFF STACK AND RETURN
                    =1    717      ;
                    =1    718      WRT220  ENDP
                          719      ;
                          720 +1   $INCLUDE(CORE.SRC)
                    =1    721 +1   $EJECT TITLE(START FUNCTION AND WAIT FOR COMPLETION)
```

```
LOC  OBJ                  LINE     SOURCE

                     =1    722     ; ------------------------------------------------
                     =1    723     ; |                                              |
                     =1    724     ; |     START FUNCTION AND WAIT FOR COMPLETION   |
                     =1    725     ; |                                              |
                     =1    726     ; ------------------------------------------------
                     =1    727     ;
                     =1    728     ;       THIS ROUTINE GIVES A CHANNEL ATTENTION (WAKE-UP) TO THE iSBC 220 AND
                     =1    729     ;       WAITS FOR THE FUNCTION SPECIFIED (BY THE CALLING PROCEDURE) TO FINISH.
                     =1    730     ;       IF AN ERROR OCCURRED, THE ERROR HANDLER IS INVOKED.
                     =1    731     ;
                     =1    732     ; INPUTS:
                     =1    733     ;       NONE
                     =1    734     ;
                     =1    735     ; OUTPUTS:
                     =1    736     ;       CARRY FLAG:     = 0 IF NO ERROR OR A RECOVERABLE ERROR OCCURRED
                     =1    737     ;                       = 1 IF UNRECOVERABLE ERROR OCCURRED.
                     =1    738     ; ------------------------------------------------------------------------------
                     =1    739     ;
0197                 =1    740     GO220   PROC    NEAR
                     =1    741     ;
0197 50              =1    742             PUSH    AX                      ; SAVE REGISTERS
0198 52              =1    743             PUSH    DX
0199 BA3506          =1    744             MOV     DX,WUA                  ; GET ADDRESS OF WAKE-UP I/O PORT
019C B001            =1    745             MOV     AL,01H                  ; GET WAKE-UP COMMAND BYTE
019E EE              =1    746             OUT     DX,AL                   ; GIVE WAKE-UP TO iSBC 220
019F E80800          =1    747             CALL    WAIT220                 ; WAIT FOR FUNCTION COMPLETE
01A2 7303            =1    748             JNC     DONE                    ; ERROR?
                     =1    749                                             ; NO--RETURN
01A4 E82900          =1    750             CALL    ERROR                   ; YES--CALL ERROR HANDLER (RETURNS WITH
                     =1    751                                             ;    CARRY FLAG SET OR CLEAR)
01A7 5A              =1    752     DONE:   POP     DX                      ; RESTORE REGISTERS
01A8 58              =1    753             POP     AX
01A9 C3              =1    754             RET                             ; RETURN
                     =1    755     ;
                     =1    756     GO220   ENDP
                     =1    757     ;
                     =1    758 +1  $EJECT TITLE(WAIT FOR FUNCTION COMPLETE ROUTINE)
```

```
LOC  OBJ                  LINE     SOURCE

                     =1    759      ; ------------------------------------
                     =1    760      ; |                                  |
                     =1    761      ; |     WAIT FOR FUNCTION COMPLETE   |
                     =1    762      ; |                                  |
                     =1    763      ; ------------------------------------
                     =1    764      ;
                     =1    765      ;       NORMALLY, THIS WAIT ROUTINE WOULD TRAP TO THE SYSTEM DISPATCHER/
                     =1    766      ;       SCHEDULER TO ALLOW ANOTHER TASK TO EXECUTE WHILE THE iSBC 220 COMPLETED
                     =1    767      ;       ITS FUNCTION.  HOWEVER, FOR THIS EXAMPLE, THE ROUTINE SIMPLY WAITS FOR
                     =1    768      ;       THE INTERRUPT SERVICE ROUTINE TO LOAD THE OPERATION COMPLETE STATUS
                     =1    769      ;       FROM THE CIB OPERATION STATUS INTO THE DATA SEGMENT.  IF AN ERROR
                     =1    770      ;       OCCURRED, THE STATUS IS AVAILABLE THERE FOR SUBSEQUENT PROCESSING BY
                     =1    771      ;       AN ERROR HANDLER.
                     =1    772      ;
                     =1    773      ; INPUT DATA:
                     =1    774      ;       OPERATION COMPLETE STATUS FROM THE CIB, COPIED INTO THE DATA SEGMENT
                     =1    775      ;               BY THE INTERRUPT ROUTINE
                     =1    776      ;
                     =1    777      ; OUTPUT DATA:
                     =1    778      ;       OPERATION COMPLETE BYTE         CLEARED
                     =1    779      ;       CARRY FLAG                      = 0 IF NO ERROR
                     =1    780      ;                                       = 1 IF ERROR OCCURRED
                     =1    781      ;       COPY OF CIB OPERATION STATUS    IN "LSTSTS" IF ERROR OCCURRED
                     =1    782      ;
                     =1    783      ;       ( OPERATION COMPLETE BYTE AND "LSTSTS" ARE IN SEGMENT "DATASEG" )
                     =1    784      ;-------------------------------------------------------------------------------
                     =1    785      ;
                     =1    786              ASSUME  DS:DATASEG
                     =1    787      ;
01AA                 =1    788      WAIT220 PROC    NEAR
                     =1    789      ;
01AA 50              =1    790              PUSH    AX                      ; SAVE REGISTERS
01AB 53              =1    791              PUSH    BX
01AC 1E              =1    792              PUSH    DS
01AD BB----        R =1    793              MOV     BX,DATASEG              ; GET POINTER TO DATA SEGMENT
01B0 8EDB            =1    794              MOV     DS,BX
01B2 BBFFFF          =1    795              MOV     BX,-1                   ; INITIALIZE INDEX REGISTER
01B5 FB              =1    796              STI                             ; MAKE SURE INTERRUPT CAN GET THROUGH
01B6 F4              =1    797              HLT                             ; *****  WAIT FOR INTERRUPT  *****
01B7 43              =1    798      WAITLP: INC     BX                      ; GET INDEX FOR NEXT UNIT
01B8 81E30300        =1    799              AND     BX,0003H                ; MASK UPPER BITS
01BC F607FF          =1    800              TEST    BYTE PTR [BX],0FFH      ; OPERATION COMPLETE STATUS = 00H?
                     =1    801                                              ;    (SIGN FLAG = BIT 7 OF OP. STATUS,
                     =1    802                                              ;     TEST INSTR. CLEARS CARRY FLAG)
01BF 74F6            =1    803              JZ      WAITLP                  ; STATUS <> 00H (OPERATION COMPLETE)?
                     =1    804                                              ; NO--CHECK NEXT UNIT
01C1 7906            =1    805              JNS     WAITDN                  ; YES--ERROR OCCURRED DURING FUNCTION?
                     =1    806                                              ; NO--RETURN WITH CARRY FLAG CLEAR
01C3 8A07            =1    807              MOV     AL,[BX]                 ; YES--SAVE CIB OP. STATUS IN "LSTSTS"
01C5 A21800          =1    808              MOV     LSTSTS,AL
01C8 F9              =1    809              STC                             ; SET CARRY FLAG TO INDICATE ERROR
01C9 C60700          =1    810      WAITDN: MOV BYTE PTR [BX],00H           ; CLEAR OPERATION COMPLETE BYTE
01CC 1F              =1    811              POP     DS                      ; RESTORE REGISTERS
01CD 5B              =1    812              POP     BX
01CE 58              =1    813              POP     AX
01CF C3              =1    814              RET                             ; RETURN
                     =1    815      ;
                     =1    816      WAIT220 ENDP
                           817      ;
                           818 +1   $INCLUDE(ERROR.SRC)
                     =1    819 +1   $EJECT TITLE(ERROR HANDLER)
```

```
LOC  OBJ                  LINE     SOURCE

                     =1    820      ; -------------------------
                     =1    821      ; |                       |
                     =1    822      ; |     ERROR HANDLER     |
                     =1    823      ; |                       |
                     =1    824      ; -------------------------
                     =1    825      ;
                     =1    826      ;       THIS ROUTINE IS SYSTEM DEPENDENT.  IN THIS EXAMPLE, THE ERROR INFOR-
                     =1    827      ;       MATION FROM THE CONTROLLER IS READ INTO SOFTWARE REGISTERS IN DATASEG,
                     =1    828      ;       WHERE IT CAN BE EXAMINED.  MORE SOPHISTICATED SYSTEMS MIGHT LOG THE
                     =1    829      ;       ERRORS TO DETERMINE WHEN A TRACK IS GOING BAD, FOR EXAMPLE.
                     =1    830      ;
                     =1    831      ; - THE TRANSFER STATUS FUNCTION WILL NOT RETURN AN ERROR.
                     =1    832      ; - THE UNIT NUMBER IN THE IOPB IS NOT CHANGED, SO THAT THE OPERATION COMPLETE
                     =1    833      ;       STATUS FOR THE TRANSFER STATUS FUNCTION WILL BE POSTED AGAINST THE SAME
                     =1    834      ;       UNIT AS CAUSED THE ERROR.
                     =1    835      ;
                     =1    836      ; INPUT DATA:
                     =1    837      ;       CIB OPERATION STATUS    IN "LSTSTS" IN DATA SEGMENT
                     =1    838      ;
                     =1    839      ; OUTPUT DATA:
                     =1    840      ;       ERROR STATUS FROM CONTROLLER    IN DATA SEGMENT
                     =1    841      ;       CIB OPERATION STATUS            IN "LSTSTS" IN DATA SEGMENT
                     =1    842      ;       CARRY FLAG                      = 0 IF SOFT (RECOVERABLE) ERROR
                     =1    843      ;                                       = 1 IF HARD (UNRECOVERABLE) ERROR
                     =1    844      ; ---------------------------------------------------------------------------
                     =1    845      ;
                     =1    846              ASSUME  DS:IOPBSEG
                     =1    847      ;
01D0                 =1    848      ERROR   PROC    NEAR
                     =1    849      ;
01D0 50              =1    850              PUSH    AX                      ; SAVE REGISTERS
01D1 1E              =1    851              PUSH    DS
01D2 B8----        R =1    852              MOV     AX,IOPBSEG              ; GET POINTER TO IOPB
01D5 8ED8            =1    853              MOV     DS,AX
01D7 A11200          =1    854              MOV     AX,BUFOFF               ; SAVE IOPB DATA BUFFER POINTER
01DA 50              =1    855              PUSH    AX
01DB A11400          =1    856              MOV     AX,BUFSEG
01DE 50              =1    857              PUSH    AX
01DF C70612000C00    =1    858              MOV     BUFOFF,OFFSET ERRSTS    ; GET POINTER TO DATA SEGMENT ERROR
01E5 C7061400----  R =1    859              MOV     BUFSEG,DATASEG          ;    STATUS REGISTERS
01EB C6060B0001      =1    860              MOV     FUNC,01H                ; SET FUNCTION = TRANSFER STATUS
01F0 C7060C000000    =1    861              MOV     MODIFY,0000H            ; CLEAR MODIFIER (ENABLE INTERRUPT ON
                     =1    862                                              ;    COMPLETION AND RETRIES)
01F6 E89EFF          =1    863              CALL    GO220                   ; START FUNCTION AND WAIT FOR COMPLETE
01F9 58              =1    864              POP     AX                      ; RESTORE IOPB DATA BUFFER POINTER
01FA A31400          =1    865              MOV     BUFSEG,AX
01FD 58              =1    866              POP     AX
01FE A31200          =1    867              MOV     BUFOFF,AX
0201 B8----        R =1    868              MOV     AX,DATASEG              ; GET POINTER TO DATA SEGMENT
0204 8ED8            =1    869              MOV     DS,AX
0206 F8              =1    870              CLC                             ; CLEAR CARRY FLAG
0207 A01800          =1    871              MOV     AL,DS:LSTSTS            ; GET OLD (ERROR) CIB OPERATION STATUS
020A 2440            =1    872              AND     AL,40H                  ; CHECK HARD ERROR BIT
020C 7401            =1    873              JZ      SFTERR                  ; HARD ERROR BIT SET?
                     =1    874                                              ; NO--LEAVE CARRY FLAG CLEAR
020E F9              =1    875              STC                             ; YES--SET CARRY FLAG
020F 1F              =1    876      SFTERR: POP     DS                      ; RESTORE REGISTERS
0210 58              =1    877              POP     AX
0211 C3              =1    878              RET
                     =1    879      ;
                     =1    880      ERROR   ENDP
                           881      ;
                           882 +1   $INCLUDE(INTRPT.SRC)
                     =1    883 +1   $EJECT TITLE(INTERRUPT SERVICE ROUTINE)
```

```
LOC  OBJ                  LINE     SOURCE

                     =1    884      ; ------------------------------------
                     =1    885      ; |                                  |
                     =1    886      ; |     INTERRUPT SERVICE ROUTINE    |
                     =1    887      ; |                                  |
                     =1    888      ; ------------------------------------
                     =1    889      ;
                     =1    890      ;       THIS ROUTINE SERVICES THE INTERRUPT GENERATED BY THE iSBC 220 UPON
                     =1    891      ;       OPERATION COMPLETE, SEEK COMPLETE, OR DISK PACK CHANGE.  IT COPIES THE
                     =1    892      ;       CIB OPERATION STATUS INTO ONE OF FOUR BYTES ASSOCIATED WITH EACH OF
                     =1    893      ;       THESE EVENTS.  IT IS ASSUMED THAT SYSTEM PROGRAMS MAKE USE OF THE
                     =1    894      ;       INFORMATION TO RESUME TASKS, HANDLE ERROR LOGGING/RECOVERY, AND KEEP
                     =1    895      ;       TRACK OF DIRECTORY INFORMATION.  FOR THIS PROGRAMMING EXAMPLE, ONLY
                     =1    896      ;       THE OPERATION COMPLETE BYTES ARE USED.
                     =1    897      ;
                     =1    898      ; - THE SYSTEM INTERRUPTS ARE CONFIGURED BY EXTERNAL PROGRAMS.
                     =1    899      ; ------------------------------------------------------------------------------
                     =1    900      ;
                     =1    901              PUBLIC  INT220
                     =1    902      ;
0212                 =1    903      INT220  PROC    FAR
                     =1    904      ;
0212 FB              =1    905              STI                             ;;; ENABLE HIGHER PRIORITY INTERRUPTS
0213 50              =1    906              PUSH    AX                      ;;; SAVE REGISTERS
0214 53              =1    907              PUSH    BX
0215 52              =1    908              PUSH    DX
0216 1E              =1    909              PUSH    DS
                     =1    910              ASSUME  DS:CIBSEG
0217 B8----        R =1    911              MOV     AX,CIBSEG               ; GET POINTER TO CIB
021A 8ED8            =1    912              MOV     DS,AX
021C A00100          =1    913              MOV     AL,OPSTS                ; GET CIB OPERATION STATUS
021F 8AD0            =1    914              MOV     DL,AL                   ; SAVE IT
0221 C606030000      =1    915              MOV     STSSEM,00H              ; CLEAR CIB STATUS SEMAPHORE
0226 8AD8            =1    916              MOV     BL,AL                   ; MOVE IT TO INDEX REGISTER
0228 81E33000        =1    917              AND     BX,0030H                ; MASK ALL BITS EXCEPT UNIT NUMBER
022C D1EB            =1    918              SHR     BX,1                    ; SHIFT UNIT NUMBER TO BITS 0 AND 1
022E D1EB            =1    919              SHR     BX,1
0230 D1EB            =1    920              SHR     BX,1
0232 D1EB            =1    921              SHR     BX,1
0234 250600          =1    922              AND     AX,0006H                ; MASK ALL BITS EXCEPT SEEK COMPLETE
                     =1    923                                              ;    AND PACK CHANGE
0237 D1E0            =1    924              SHL     AX,1                    ; SHIFT LEFT TO GET OFFSET INTO PROPER
                     =1    925                                              ;    BYTE IN DATA SEGMENT
0239 03D8            =1    926              ADD     BX,AX                   ; COMBINE WITH UNIT IN INDEX REGISTER
                     =1    927              ASSUME  DS:DATASEG
023B B8----        R =1    928              MOV     AX,DATASEG              ; GET POINTER TO DATA SEGMENT
023E 8ED8            =1    929              MOV     DS,AX
0240 8817            =1    930              MOV     [BX],DL                 ; MOVE OPERATION STATUS TO DATA SEGMENT
0242 BA5063          =1    931              MOV     DX,WUA*16               ; GET POINTER TO I/O WAKE-UP ADDRESS
0245 B002            =1    932              MOV     AL,02H                  ; GET CLEAR INTERRUPT COMMAND BYTE
0247 EE              =1    933              OUT     DX,AL                   ; OUTPUT TO iSBC 220
                     =1    934      ;
0248 1F              =1    935              POP     DS                      ; RESTORE REGISTERS
0249 5A              =1    936              POP     DX
024A 5B              =1    937              POP     BX
024B FA              =1    938              CLI                             ; DISABLE INTERRUPTS FOR RESTORE
                     =1    939                                              ; (RESTORATION OF INTERRUPT LOGIC STATE
                     =1    940                                              ; IS SYSTEM DEPENDENT. THIS EXAMPLE USES
                     =1    941                                              ; THE iSBC 86/12A CPU.)
024C B020            =1    942              MOV     AL,20H                  ;;; GET END-OF-INTERRUPT COMMAND
024E E6C0            =1    943              OUT     0C0H,AL                 ;;; OUTPUT EOI COMMAND TO 8259
0250 58              =1    944              POP     AX                      ;;;
0251 CF              =1    945              IRET                            ;;; INTERRUPT RETURN ENABLES INTERRUPTS
                     =1    946      ;
                     =1    947      INT220  ENDP
                           948      ;
----                       949      SBC220DRIVER    ENDS                    ; END OF iSBC 220 DRIVER CODE
                           950      ;
                           951 +1   $TITLE(SYMBOL TABLE AND CROSS REFERENCE)
                           952      ;
                           953              END                             ; END OF PROGRAMMING EXAMPLE
```

```
XREF SYMBOL TABLE LISTING
---- ------ ----- -------


NAME            TYPE     VALUE  ATTRIBUTES, XREFS

??SEG . . . .   SEGMENT          SIZE=0000H PARA PUBLIC
ACTCNT. . . .   V DWORD  0004H  IOPBSEG  129#
ACTCYL. . . .   V WORD   0013H  DATASEG  260#
ACTHD . . . .   V BYTE   0015H  DATASEG  261#
ACTSEC. . . .   V BYTE   0016H  DATASEG  262#
BSYFLG1 . . .   V BYTE   0001H  CCBSEG  79# 459
BSYFLG2 . . .   V BYTE   0009H  CCBSEG  85#
BUFOFF. . . .   V WORD   0012H  IOPBSEG  137# 511 513 584 585 641 702 854 858 867
BUFSEG. . . .   V WORD   0014H  IOPBSEG  138# 509 586 643 704 856 859 865
CCB . . . . .   L FAR    0000H  CCBSEG  64 77# 402 403
CCBPTR. . . .   V DWORD  0002H  SCBSEG  64# 402 403
CCBSEG. . . .   SEGMENT         SIZE=0010H PARA     75# 92 407 451 452
CCW1. . . . .   V BYTE   0000H  CCBSEG  78# 408
CCW2. . . . .   V BYTE   0008H  CCBSEG  84# 411
CH1PC . . . .   L FAR    0004H  CIBSEG  80 110# 409 410 423
CH1PTR. . . .   V DWORD  0002H  CCBSEG  80# 409 410
CH2PC . . . .   L FAR    000EH  CCBSEG  86 89# 412 413 414
CH2PTR. . . .   V DWORD  000AH  CCBSEG  86# 412 413
CIB . . . . .   L FAR    0000H  CIBSEG  105#
CIBCMD. . . .   V BYTE   0000H  CIBSEG  106# 421
CIBSEG. . . .   SEGMENT         SIZE=0010H PARA     103# 116 418 419 910 911
CLRLP . . . .   L NEAR   0074H  SBC220DRIVER  435# 438
CMDSEM. . . .   V BYTE   0002H  CIBSEG  108# 422
CONFIG. . . .   L FAR    0000H  USERSEG PUBLIC  325 328# 344
CYLNDR. . . .   V WORD   000EH  IOPBSEG  134# 581 637 698
DATASEG . . .   SEGMENT         SIZE=001AH PARA     216# 274 430 431 786 793 859 868 927 928
DESCYL. . . .   V WORD   000FH  DATASEG  257#
DESHD . . . .   V BYTE   0011H  DATASEG  258#
DESSEC. . . .   V BYTE   0012H  DATASEG  259#
DEVCOD. . . .   V WORD   0008H  IOPBSEG  130#
DONE. . . . .   L NEAR   01A7H  SBC220DRIVER  748 752#
ENDDAT. . . .   L FAR    001AH  DATASEG  272# 433
ENDSTK. . . .   L FAR    0040H  STACK  315# 333
ERROR . . . .   L NEAR   01D0H  SBC220DRIVER  750 848# 880
ERRSTS. . . .   V WORD   000CH  DATASEG PUBLIC  226 255# 858
FMT220. . . .   L FAR    00D6H  SBC220DRIVER PUBLIC  569 572# 596
FMTDN . . . .   L NEAR   010AH  SBC220DRIVER  592#
FUNC. . . . .   V BYTE   000BH  IOPBSEG  132# 506 587 650 711 860
GO220 . . . .   L NEAR   0197H  SBC220DRIVER  515 590 651 712 740# 756 863
HEAD. . . . .   V BYTE   0010H  IOPBSEG  135# 583 639 700
INIT220 . . .   L FAR    00A1H  SBC220DRIVER PUBLIC  497 500# 527
INITBLSEG . .   SEGMENT         SIZE=0020H PARA     168# 206 509
INITDN. . . .   L NEAR   00D3H  SBC220DRIVER  517 523#
INITLP. . . .   L NEAR   00C1H  SBC220DRIVER  513# 521
INT220. . . .   L FAR    0212H  SBC220DRIVER PUBLIC  336 337 901 903# 947
INTRCS. . . .   V WORD   0096H  SEG0000  303# 337
INTRIP. . . .   V WORD   0094H  SEG0000  302# 336
INTRPT. . . .   NUMBER   0005H    295# 300
IOPB. . . . .   L FAR    0000H  IOPBSEG  112 127# 424 425
IOPBOFF . . .   V WORD   0008H  CIBSEG  112# 424
IOPBSEG . . .   SEGMENT         SIZE=0015H PARA     113 125# 142 498 504 570 576 626 632 687 693 846 852
IOPBSG. . . .   V WORD   000AH  CIBSEG  113# 425
LSTSTS. . . .   V BYTE   0018H  DATASEG  268# 808 871
MODIFY. . . .   V WORD   000CH  IOPBSEG  133# 507 588 648 709 861
NMRTRY. . . .   V BYTE   0017H  DATASEG  263#
OPCMP . . . .   V BYTE   0000H  DATASEG PUBLIC  226 230#
OPCMP0. . . .   V BYTE   0000H  DATASEG  231#
OPCMP1. . . .   V BYTE   0001H  DATASEG  232#
OPCMP2. . . .   V BYTE   0002H  DATASEG  233#
OPCMP3. . . .   V BYTE   0003H  DATASEG  234#
OPSTS . . . .   V BYTE   0001H  CIBSEG  107# 913
PKCHG . . . .   V BYTE   0008H  DATASEG PUBLIC  226 246#
PKCHG0. . . .   V BYTE   0008H  DATASEG  247#
PKCHG1. . . .   V BYTE   0009H  DATASEG  248#
PKCHG2. . . .   V BYTE   000AH  DATASEG  249#
PKCHG3. . . .   V BYTE   000BH  DATASEG  250#
RD220 . . . .   L FAR    010FH  SBC220DRIVER PUBLIC  625 628# 657
REQCNT. . . .   V DWORD  0016H  IOPBSEG  139# 645 647 706 708
RES220. . . .   L FAR    0000H  SBC220DRIVER PUBLIC  384 386# 473
RESDN . . . .   L NEAR   009BH  SBC220DRIVER  461 466#
RESLP . . . .   L NEAR   0091H  SBC220DRIVER  459# 463
SBC220DRIVER.   SEGMENT         SIZE=0252H PARA     348# 350 949
SCB . . . . .   L FAR    0000H  SCBSEG  61#
SCBSEG. . . .   SEGMENT         SIZE=0006H PARA ABS    59# 66 398 399
SECTOR. . . .   V BYTE   0011H  IOPBSEG  136#
SEG0000 . . .   SEGMENT         SIZE=0098H PARA ABS    297# 305 326
```

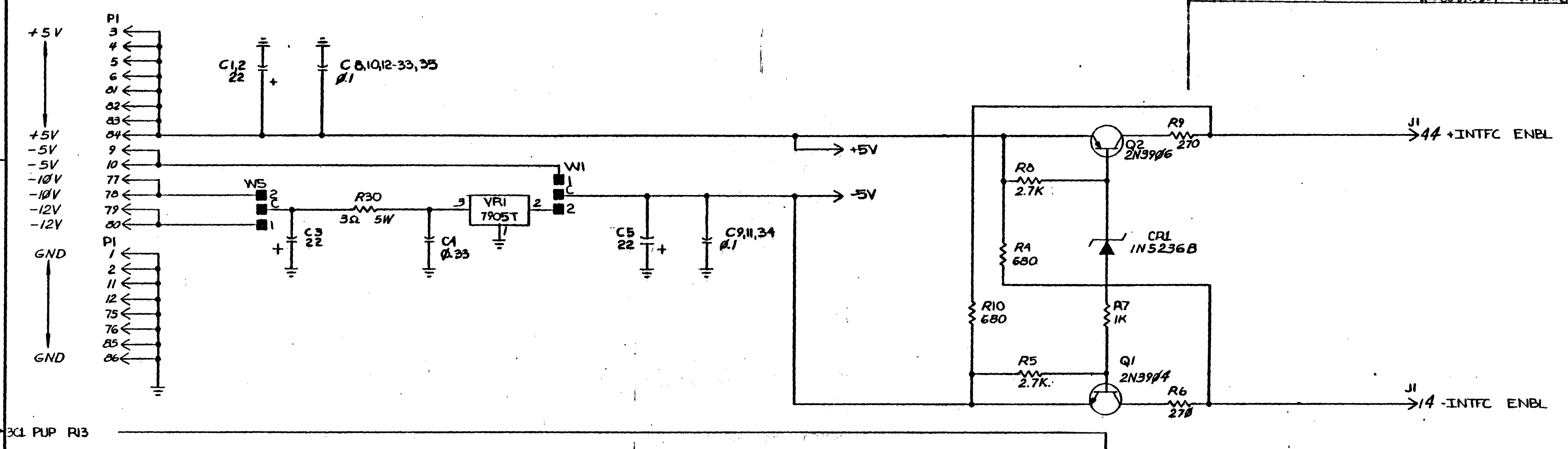
```
NAME            TYPE     VALUE  ATTRIBUTES, XREFS

SFERST. . . . V BYTE   000EH  DATASEG  256#
SFTERR. . . . L NEAR   020FH  SBC220DRIVER  873 876#
SKCMP . . . . V BYTE   0004H  DATASEG PUBLIC  226 238#
SKCMP0. . . . V BYTE   0004H  DATASEG  239#
SKCMP1. . . . V BYTE   0005H  DATASEG  240#
SKCMP2. . . . V BYTE   0006H  DATASEG  241#
SKCMP3. . . . V BYTE   0007H  DATASEG  242#
SOC . . . . . V BYTE   0000H  SCBSEG  62# 401
STACK . . . . SEGMENT         SIZE=0040H PARA
STSSEM. . . . V BYTE   0003H  CIBSEG  109# 915
UNIT. . . . . V BYTE   000AH  IOPBSEG  131# 514 579 635 696
USERSEG . . . SEGMENT         SIZE=0022H PARA    323# 346
WAIT220 . . . L NEAR   01AAH  SBC220DRIVER  747 788# 816
WAITDN. . . . L NEAR   01C9H  SBC220DRIVER  805 810#
WAITLP. . . . L NEAR   01B7H  SBC220DRIVER  798# 803
WRT220. . . . L FAR    0153H  SBC220DRIVER PUBLIC  686 689# 718
WUA . . . . . NUMBER   0635H    57# 59 444 744 931

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

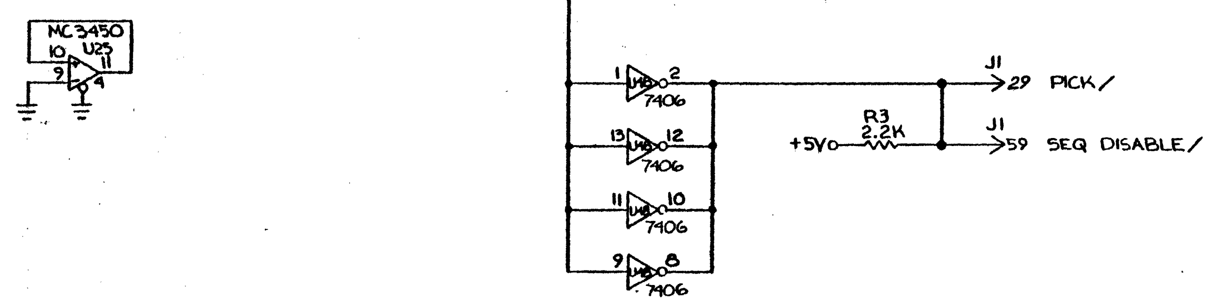
THIS DRAWING CONTAINS INFORMATION WHICH IS THE PROPRIETARY PROPERTY OF INTEL CORPORATION. THIS DRAWING IS RECEIVED IN CONFIDENCE AND ITS CONTENTS MAY NOT BE DISCLOSED WITHOUT THE PRIOR WRITTEN CONSENT OF INTEL CORPORATION.

DWG NO. 162033	SH 1	D

REVISIONS

ZONE	REV	DESCRIPTION	DFT	CHK	DATE	APPROVED
	A	ECO 14-0050 MKE PILOT RLSE	[illegible]	[illegible]	6-5-80	LA King
	B	ECO 14-0113	[illegible]	[illegible]	8-12-80	LA King
	C	ECO 14-0197	[illegible]	[illegible]	10-9-80	[illegible]
	D	ECO 14-0259	[illegible]	[illegible]	12-4-80	LA King





PWR & GRD LOCATOR CHART			
REF DESIG	POWER PINS		
	GND	+5	-5
U8-10,14,15,24,25	8	16	12
U 3-6, 11, 16	8	16	9
U12,13,17,18, 28-31,34,	7	14	
40,42-49, 58,60,62,63	↓	↓	
66-75, 87, 98-101	↓	↓	
U1,2,7,19,20, 32,33, 35-39	8	16	
41, 61, 65	↓	↓	
U59, 94-97	9	18	
U21-23, 26, 27, 50-57	10	20	
76-78, 80, 81, 85, 86, 88	↓	↓	
89, 90-93	↓	↓	
U64, 82, 83	12	24	
U79	1,20	40	

NOTES: UNLESS OTHERWISE SPECIFIED.
1. RESISTANCE VALUES ARE IN OHMS, 1/4W ±5%
2. CAPACITANCE VALUES ARE IN MICROFARADS.

SPARE GATES

TYPE	REF DESIG	OUTPUT PIN
74S02	U18	1,13
74LS32	U13	3,11
74LS32	U49	3
74LS10	U70	12
74LS01	U30	6
74LS02	U62	3
74LS02	U68	11

REF DESIGNATION

LAST USED	NOT USED
Y1	
W5	W3
VR1	
U101	
S2	
RP29	
R30	R16
Q2	
J3	
CR2	
C35	

NEXT ASSY	USED ON
[illegible]	iSBC 220
[illegible]	iSBC 220

QUANTITY PER DASH NO.

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

ITEM NO.	PART NUMBER	DESCRIPTION

PARTS LIST

SIGNATURE		DATE
DRN BY	A CIGNETTI	[illegible]
CHK BY	[illegible]	[illegible]
ENGR APVD	LA King	[illegible]
APVD		
APVD		

intel
3065 BOWERS AVE.
SANTA CLARA
CALIF. 95051

TITLE SCHEMATIC DIAGRAM— iSBC 220-SMD DISK CONTROLLER

SIZE	CODE	CLASS	DWG. NO.	REV
D	14	01	162033	D

SCALE SHEET 1 OF 12

REVISIONS

ZONE	REV	DESCRIPTION	DFT	CHK	DATE	APPROVED
		SEE SHEET 1				

9C1 (ADR-F)-(ADR-Ø)

+5V

S1

S2

ADR-F/ SW-F/
ADR-E/ SW-E/
ADR-D/ SW-D/
ADR-C/ SW-C/
ADR-B/ SW-B/
ADR-A/ SW-A/
ADR-9/ SW-9/
ADR-8/ SW-8/
ADR-7/ SW-7/
ADR-6/ SW-6/
ADR-5/ SW-5/
ADR-4/ SW-4/
ADR-3/ SW-3/
ADR-2/ SW-2/
ADR-1/ SW-1/
ADR-Ø/ SW-Ø/

R22 10K
RP28 1 10K
RP28 2 10K
RP28 3 10K
RP28 4 10K
RP28 5 10K
RP28 6 10K
RP28 7 10K
RP28 8 10K
RP29 3 10K
RP29 2 10K
RP29 4 10K
RP29 5 10K
RP29 6 10K
RP29 7 10K
RP29 8 10K
R26 10K
R23 10K
R15 10K

U72 U73 U74 U75 LS266

R21 680 +5V
R25 680 +5V

19 16BIT I/O/
20 16BIT SYS BUS/

U71 LS32
U70 LS10
U69 74LS14

P1
22 IOWC/

5C1 IADR-Ø/
5B1 IADR-Ø
5C1 GATE SWS/

U88 74LS244
U89 74LS244
U90 74LS244

SW-F/ SW-E/ SW-D/ SW-C/ SW-B/ SW-A/ SW-9/ SW-8/
SW-7/ SW-6/ SW-5/ SW-4/ SW-3/ SW-2/ SW-1/ SW-Ø/

OE

DAT-F/ DAT-E/ DAT-D/ DAT-C/ DAT-B/ DAT-A/ DAT-9/ DAT-8/
DAT-7/ DAT-6/ DAT-5/ DAT-4/ DAT-3/ DAT-2/ DAT-1/ DAT-Ø/

(DAT-F/)-(DAT-Ø/) 4B8.

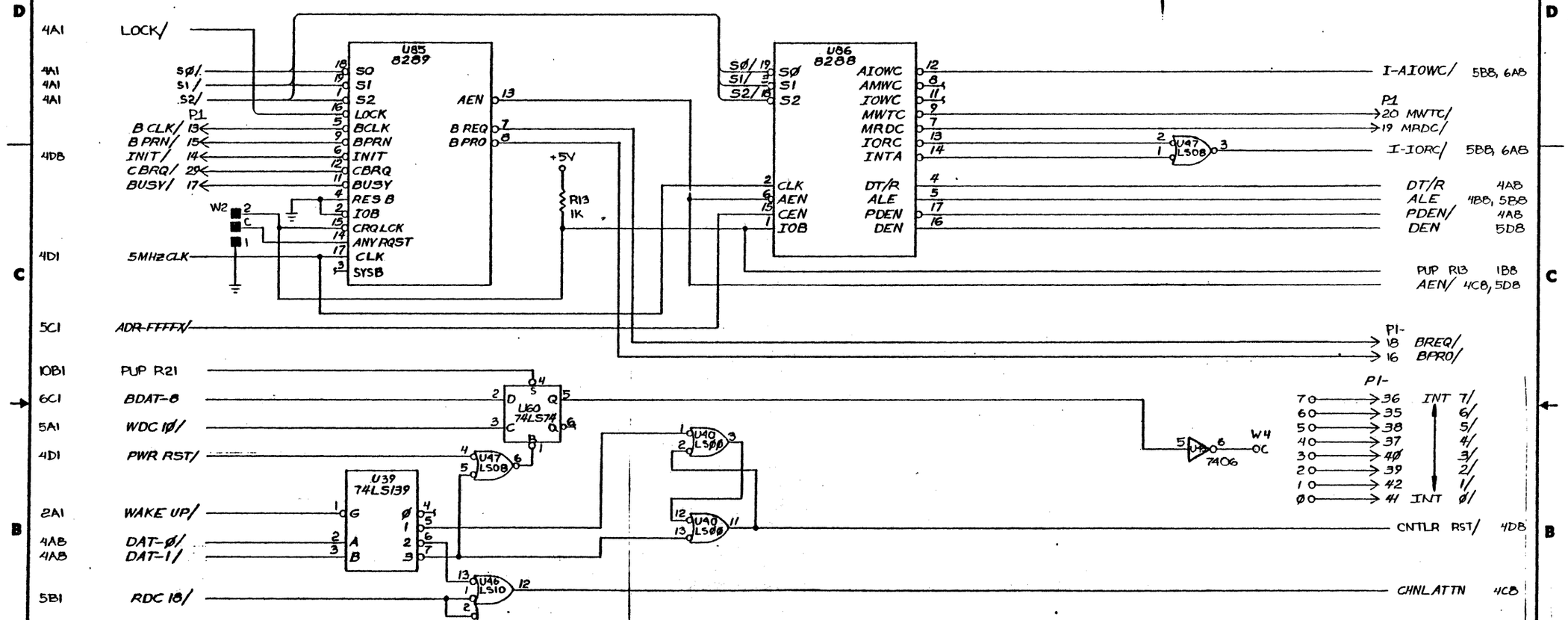
WAKE UP/ 3B8, 5D8
16 BIT SYS BUS/ 5D8

DRAWN	SIZE	CODE	CLASS	DWG NO.	REV
ISSUED	D	14	01	162033	D
	SCALE	NONE		SHEET 2	

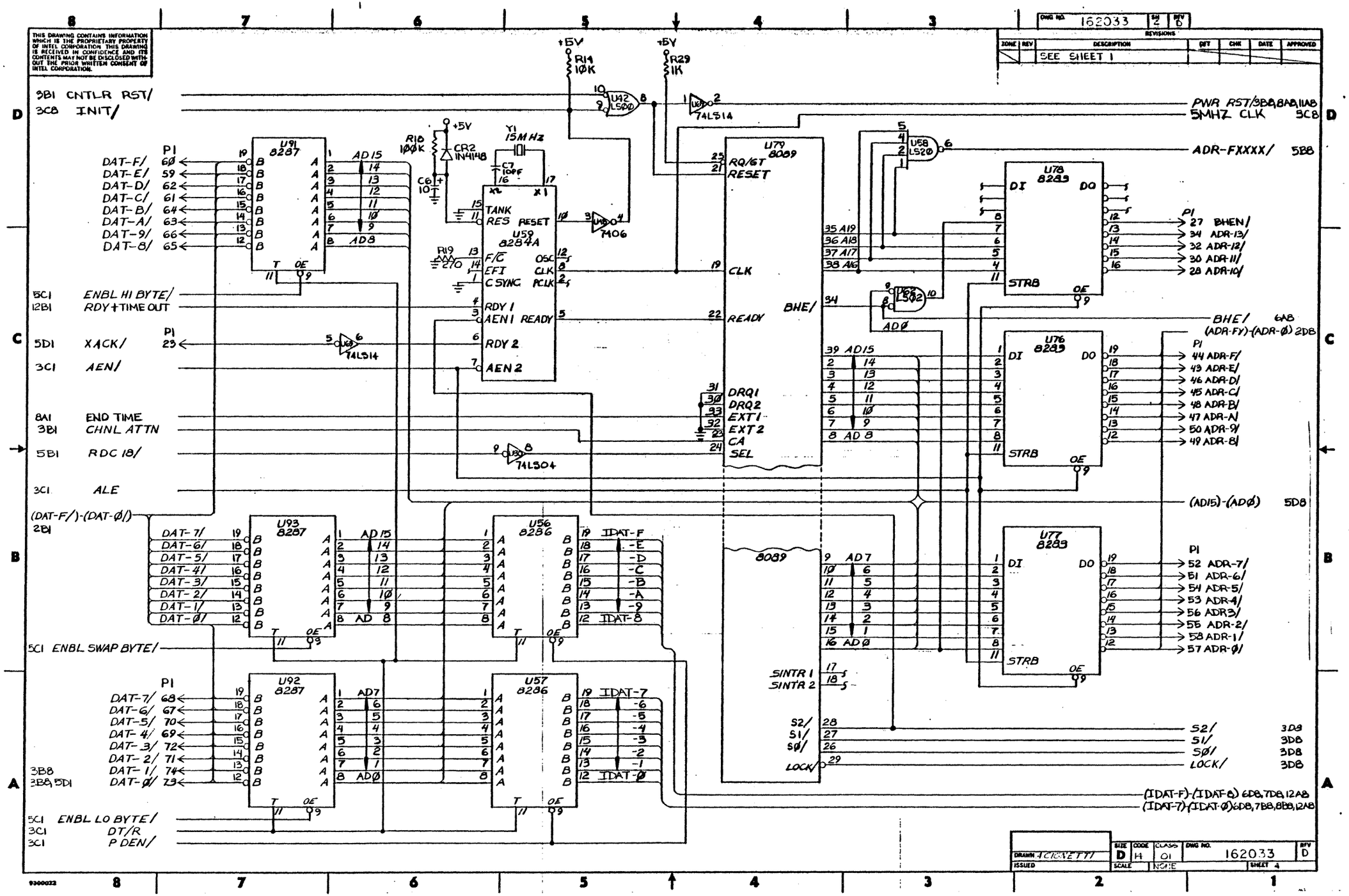
DWG NO. 162033 SH 3 REV D

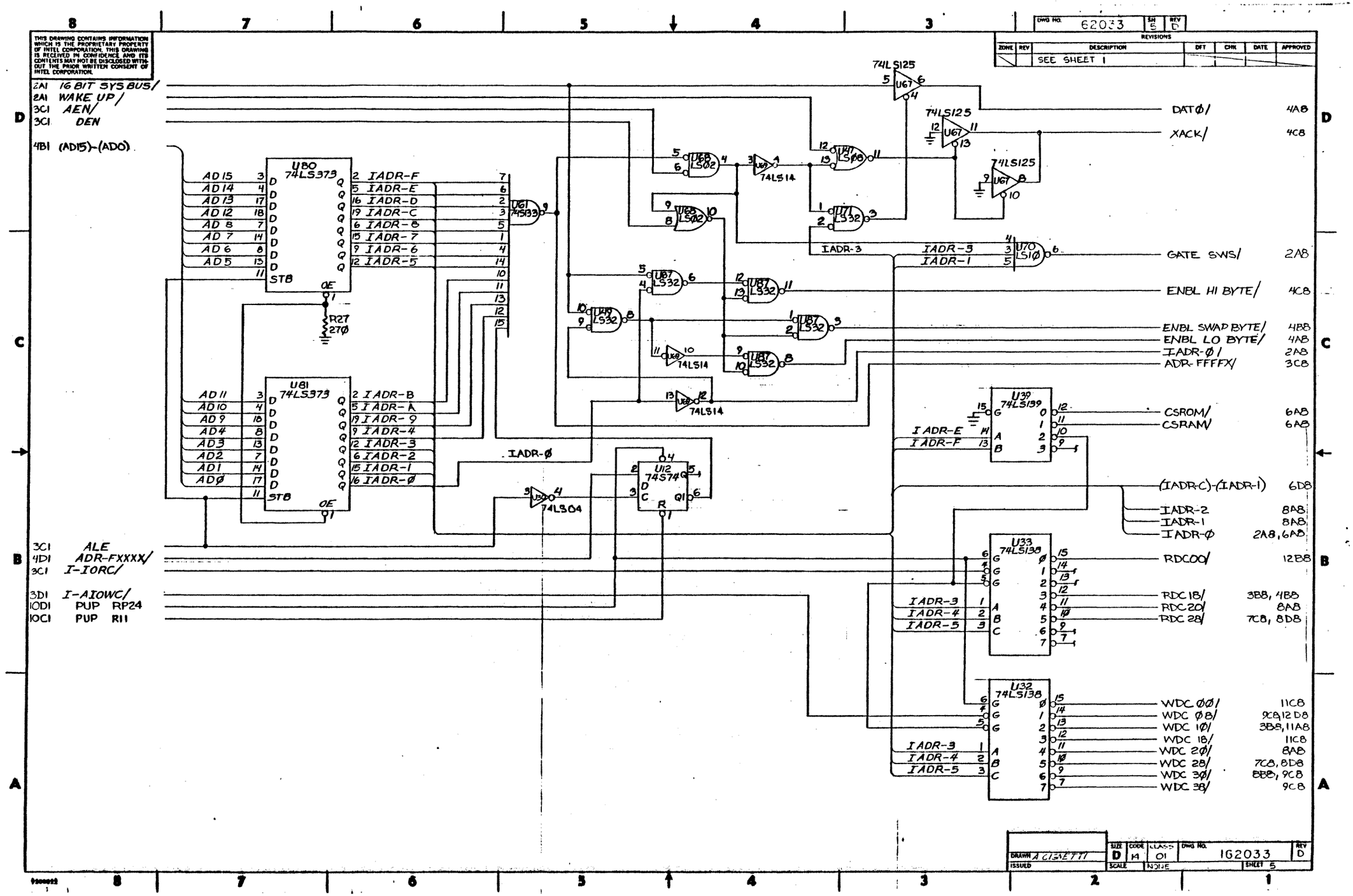
REVISIONS

ZONE	REV	DESCRIPTION	DFT	CHK	DATE	APPROVED
		SEE SHEET 1				



DRAWN	SIZE	CODE	CLASS	DWG NO.	REV
ISSUED	D	14	01	162033	D
	SCALE	NONE		SHEET 3	


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.
DWG NO. 162033 SH 2 REV D
REVISIONS
ZONE REV DESCRIPTION DFT CHK DATE APPROVED
SEE SHEET 1
3B1 CNTLR RST/
3C8 INIT/
+5V
R19 10K
R29 1K
U42 LS00
U69 74LS14
PWR RST/3B8,8A8,11A8
5MHZ CLK 3C8
ADR-FXXXX/ 5B8
U91 8287
U92 8287
U93 8287
U56 8286
U57 8286
U79 8089
U59 8284A
U58 LS20
U68 LS02
U78 8283
U76 8283
U77 8283
U80 74LS04
U67 7406
Y1 15MHz
R18 10K
CR2 IN4148
C7 100pF
C6 10
R19 270
P1
DAT-F/ 60
DAT-E/ 59
DAT-D/ 62
DAT-C/ 61
DAT-B/ 64
DAT-A/ 63
DAT-9/ 66
DAT-8/ 65
DAT-7/ 68
DAT-6/ 67
DAT-5/ 70
DAT-4/ 69
DAT-3/ 72
DAT-2/ 71
DAT-1/ 74
DAT-0/ 73
5C1 ENBL HI BYTE/
12B1 RDY+TIME OUT
5D1 XACK/ P1 23
3C1 AEN/
8A1 END TIME
3B1 CHNL ATTN
5B1 RDC 18/
3C1 ALE
(DAT-F/)-(DAT-0/) 2B1
5C1 ENBL SWAP BYTE/
3B8
3B8,5D1
5C1 ENBL LO BYTE/
3C1 DT/R
3C1 P DEN/
RQ/GT
RESET
CLK
READY
BHE/
DRQ1
DRQ2
EXT1
EXT2
CA
SEL
SINTR 1
SINTR 2
S2/
S1/
S0/
LOCK/
8089
BHE/ 6A8
(ADR-F)-(ADR-0) 2D8
(AD15)-(AD0) 5D8
(IDAT-F)-(IDAT-8) 6D8,7D8,12A8
(IDAT-7)-(IDAT-0) 6D8,7B8,8B8,12A8
DRAWN TORNETTA
ISSUED
SIZE D CODE 14 CLASS 01 DWG NO. 162033 RFV D
SCALE NONE SHEET 4
9300022


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.
DWG NO. 62033 SH 5 REV D
REVISIONS
ZONE REV DESCRIPTION DFT CHK DATE APPROVED
SEE SHEET 1
2A1 16 BIT SYS BUS/
2A1 WAKE UP/
3C1 AEN/
3C1 DEN
4B1 (AD15)-(AD0)
U80 74LS373
U81 74LS373
AD15 AD14 AD13 AD12 AD8 AD7 AD6 AD5
AD11 AD10 AD9 AD4 AD3 AD2 AD1 AD0
STB OE
IADR-F IADR-E IADR-D IADR-C IADR-8 IADR-7 IADR-6 IADR-5
IADR-B IADR-A IADR-9 IADR-4 IADR-3 IADR-2 IADR-1 IADR-0
R27 270
U61 74S133
U68 LS02
U69 74LS14
U71 LS08
U65 LS02
U67 74LS125
U70 LS10
U87 LS32
U89 LS32
U12 74S74
U84 74LS04
IADR-3 IADR-1 IADR-0
DAT0/ 4A8
XACK/ 4C8
GATE SWS/ 2A8
ENBL HI BYTE/ 4C8
ENBL SWAP BYTE/ 4B8
ENBL LO BYTE/ 4A8
IADR-0/ 2A8
ADR-FFFFX/ 3C8
U39 74LS139
IADR-E IADR-F
CSROM/ 6A8
CSRAM/ 6A8
(IADR-C)-(IADR-1) 6D8
IADR-2 8A8
IADR-1 8A8
IADR-0 2A8, 6A8
U33 74LS138
IADR-3 IADR-4 IADR-5
RDC00/ 12B8
RDC 18/ 3B8, 4B8
RDC 20/ 8A8
RDC 28/ 7C8, 8D8
U32 74LS138
IADR-3 IADR-4 IADR-5
WDC 00/ 11C8
WDC 08/ 9C8, 12D8
WDC 10/ 3B8, 11A8
WDC 18/ 11C8
WDC 20/ 8A8
WDC 28/ 7C8, 8D8
WDC 30/ 8B8, 9C8
WDC 38/ 9C8
3C1 ALE
4D1 ADR-FXXXX/
3C1 I-IORC/
3D1 I-AIOWC/
10D1 PUP RP24
10C1 PUP R11
DRAWN A CISZEWSKI
ISSUED
SIZE D CODE 14 CLASS 01 DWG NO. 162033 REV D
SCALE NONE SHEET 5

DWG NO. 162033 SH - REV D

REVISIONS

ZONE	REV	DESCRIPTION	DFT	CHK	DATE	APPROVED
		SEE SHEET 1				

4AI (IDAT-F)-(IDAT-8)
4AI (IDAT-7)-(IDAT-Ø)
5BI (IADR-C)(IADR-1)

U84 2114A-5
IADR-A 5, IADR-9 17, IADR-8 2, IADR-7 3, IADR-6 1, IADR-5 4, IADR-4 7, IADR-3 15, IADR-2 6, IADR-1 16
A A A A A A A A A A
I/O 14 IDAT -F
I/O 13 IDAT -E
I/O 12 IDAT -D
I/O 11 IDAT -C
WE 10, CS 8

U83 2732
IADR-C 21, IADR-B 19, IADR-A 22, IADR-9 23, IADR-8 1, IADR-7 2, IADR-6 3, IADR-5 4, IADR-4 5, IADR-3 6, IADR-2 7, IADR-1 8
A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
O O O O O O O O
17 IDAT -F
16 IDAT -E
15 IDAT -D
14 IDAT -C
13 IDAT -B
11 IDAT -A
10 IDAT -9
9 IDAT -8
CE 18, OE 20

U85 2114A-5
IADR-A 5, IADR-9 17, IADR-8 2, IADR-7 3, IADR-6 1, IADR-5 4, IADR-4 7, IADR-3 15, IADR-2 6, IADR-1 16
A A A A A A A A A A
14 IDAT -B
13 IDAT -A
12 IDAT -9
11 IDAT -8
WE 10, CS 8

U22 74LS244
IDAT -F 11, IDAT -E 8, IDAT -D 6, IDAT -C 13, IDAT -B 15, IDAT -A 4, IDAT -9 2, IDAT -8 17
9 BDAT-F
12 BDAT-E
14 BDAT-D — BDAT-D 11DB
7 BDAT-C — BDAT-C 11DB
5 BDAT-B
16 BDAT-A
18 BDAT-9
3 BDAT-8 — BDAT-8 3CB
OE 1, 19

(BDAT-F)-(BDAT-Ø) 7DB, 9DB
(BDAT-9)-(BDAT-0) 11DB

U82 2732
IADR-C 21, IADR-B 19, IADR-A 22, IADR-9 23, IADR-8 1, IADR-7 2, IADR-6 3, IADR-5 4, IADR-4 5, IADR-3 6, IADR-2 7, IADR-1 8
A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
O O O O O O O O
17 IDAT -7
16 IDAT -6
15 IDAT -5
14 IDAT -4
13 IDAT -3
11 IDAT -2
10 IDAT -1
9 IDAT -Ø
CE 18, OE 20

U86 2114A-5
IADR-A 5, IADR-9 17, IADR-8 2, IADR-7 3, IADR-6 1, IADR-5 4, IADR-4 7, IADR-3 15, IADR-2 6, IADR-1 16
A A A A A A A A A A
14 IDAT -7
13 IDAT -6
12 IDAT -5
11 IDAT -4
WE 10, CS 8

U23 74LS244
IDAT-7 8, IDAT-6 6, IDAT-5 4, IDAT-4 2, IDAT -3 17, IDAT -2 15, IDAT -1 13, IDAT -Ø 11
12 BDAT-7
14 BDAT-6
16 BDAT-5
18 BDAT-4
3 BDAT-3
5 BDAT-2
7 BDAT-1
9 BDAT-0
OE 1, 19

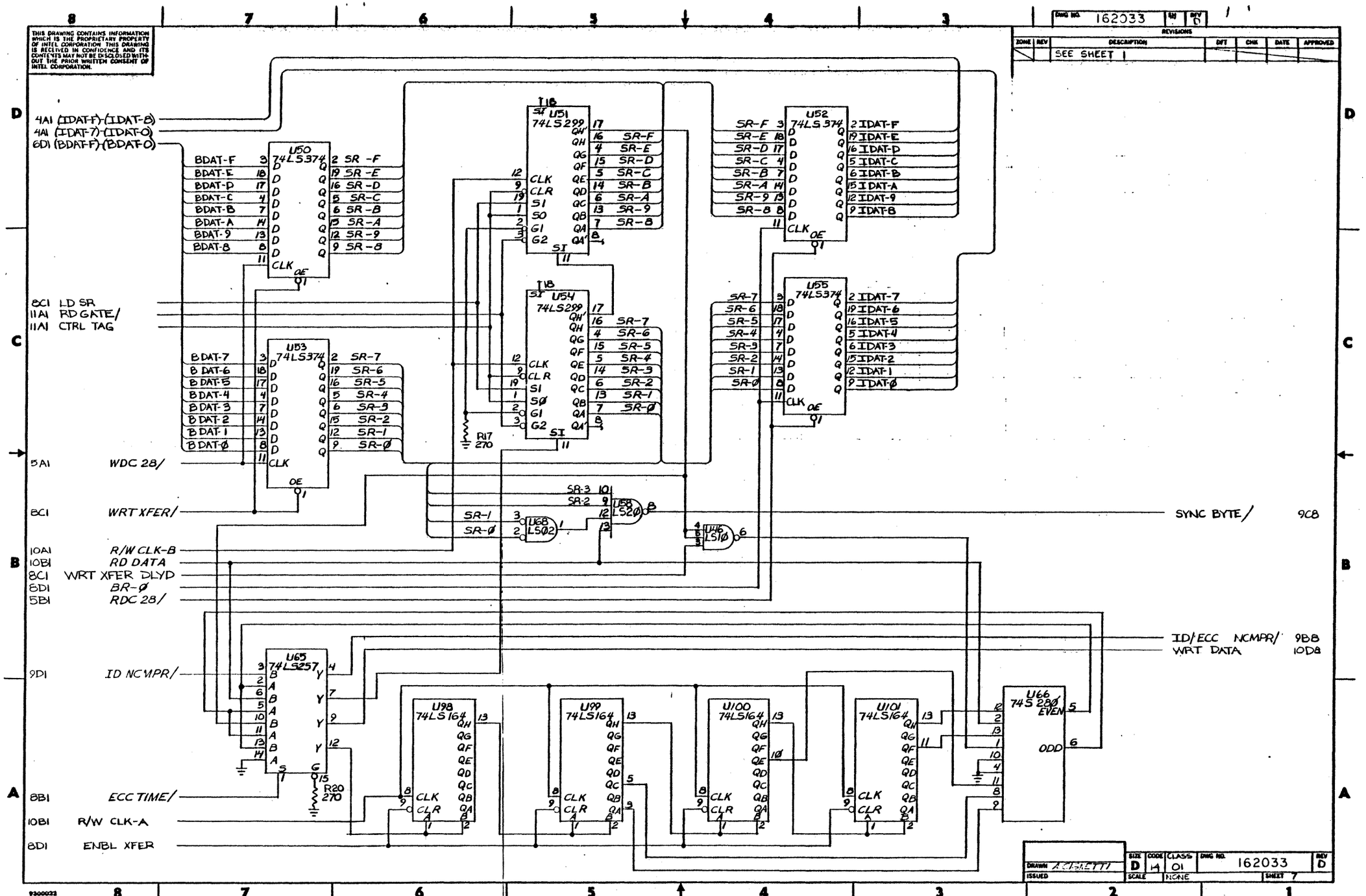
R12 270Ω

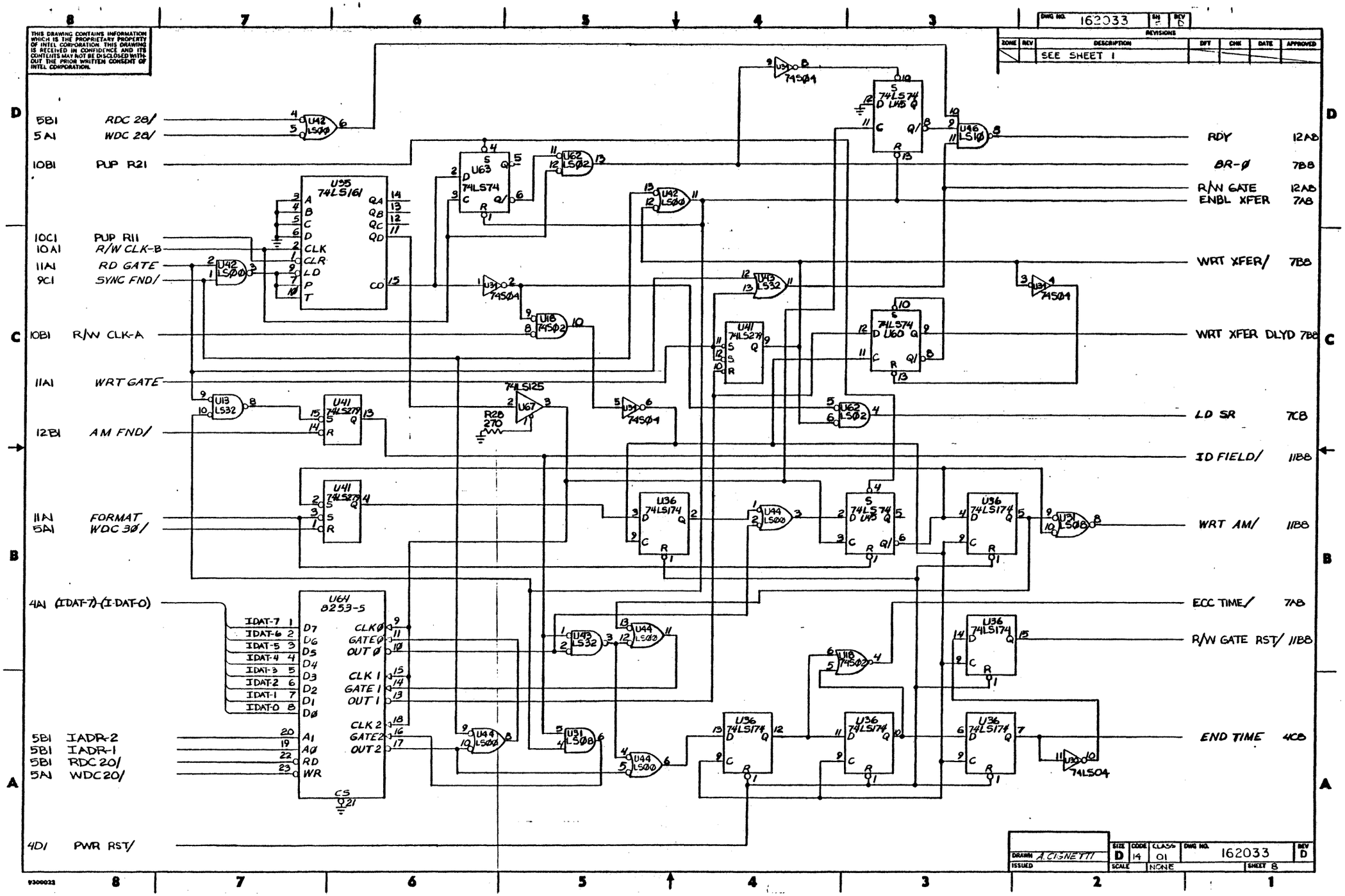
U87 2114A-5
IADR-A 5, IADR-9 17, IADR-8 2, IADR-7 3, IADR-6 1, IADR-5 4, IADR-4 7, IADR-3 15, IADR-2 6, IADR-1 16
A A A A A A A A A A
14 IDAT -3
11 IDAT -2
12 IDAT -1
13 IDAT -Ø
WE 10, CS 8

5CI CSROM/
3CI I-IDRC/
5CI CSRAM/
3DI I-AIOWC/
5BI IADR-Ø
4CI BHE/

U49 LS32: 4, 5 → 6
U47 LS08: 9, 10 → 8
U43 LS32: 5, 4 → 6
U43 LS32: 10, 9 → 8

DRAWN	SIZE	CODE	CLASS	DWG NO.	REV
ISSUED	D	14	01	162033	D
	SCALE NONE			SHEET 6	


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.
DWG NO. 162033 REV D
REVISIONS
ZONE REV DESCRIPTION DFT CHK DATE APPROVED
SEE SHEET 1
4A1 (IDAT-F)-(IDAT-8)
4A1 (IDAT-7)-(IDAT-0)
6D1 (BDAT-F)-(BDAT-0)
8C1 LD SR
11A1 RD GATE/
11A1 CTRL TAG
5A1 WDC 28/
8C1 WRT XFER/
10A1 R/W CLK-B
10B1 RD DATA
8C1 WRT XFER DLYD
8D1 BR-Ø
5B1 RDC 28/
9D1 ID NCMPR/
8B1 ECC TIME/
10B1 R/W CLK-A
8D1 ENBL XFER
SYNC BYTE/ 9C8
ID/ECC NCMPR/ 9B8
WRT DATA 10D8
U50 74LS374
U51 74LS299
U52 74LS374
U53 74LS374
U54 74LS299
U55 74LS374
U65 74LS257
U66 74S280 EVEN ODD
U98 74LS164
U99 74LS164
U100 74LS164
U101 74LS164
U68 LS02
U58 LS20
U46 LS10
R17 270
R20 270
DRAWN
ISSUED
SIZE D CODE 14 CLASS 01 DWG NO. 162033 REV D
SCALE NONE SHEET 7
9300022


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.
DWG NO. 162033 SH 2 REV D
REVISIONS
ZONE REV DESCRIPTION DFT CHK DATE APPROVED
SEE SHEET 1
5B1 RDC 28/
5A1 WDC 28/
10B1 PUP R21
10C1 PUP R11
10A1 R/W CLK-B
11A1 RD GATE
9C1 SYNC FND/
10B1 R/W CLK-A
11A1 WRT GATE
12B1 AM FND/
11A1 FORMAT
5A1 WDC 30/
4A1 (IDAT-7)-(I-DAT-0)
5B1 IADR-2
5B1 IADR-1
5B1 RDC 20/
5A1 WDC 20/
4D1 PWR RST/
U35 74LS161
U63 74LS74
U62 LS02
U42 LS00
U43 LS32
U18 74S02
U3 74S04
U41 74LS279
U13 LS32
74LS125
R28 270
U67
U64 8253-5
IDAT-7
IDAT-6
IDAT-5
IDAT-4
IDAT-3
IDAT-2
IDAT-1
IDAT-0
CLK Ø
GATE Ø
OUT Ø
CLK 1
GATE 1
OUT 1
CLK 2
GATE 2
OUT 2
U44 LS00
U31 LS08
U36 74LS174
U60 74LS74
U61 74LS74
RDY 12AB
BR-Ø 7BB
R/W GATE 12AB
ENBL XFER 7AB
WRT XFER/ 7BB
WRT XFER DLYD 7BB
LD SR 7CB
ID FIELD/ 11BB
WRT AM/ 11BB
ECC TIME/ 7AB
R/W GATE RST/ 11BB
END TIME 4CB
74LS04
DRAWN A. CIGNETTI
ISSUED
SIZE D CODE 14 CLASS 01 DWG NO. 162033 REV D
SCALE NONE SHEET 8
9300022

REVISIONS

ZONE	REV	DESCRIPTION	DFT	CHK	DATE	APPROVED
		SEE SHEET 1				

6DI (BDAT-F)-(BDAT-Ø)

U19 74LS165 — BDAT-7, BDAT-6, BDAT-5, BDAT-4, BDAT-3, BDAT-2, BDAT-1, BDAT-Ø
U37 74LS165 — BDAT-F, BDAT-E, BDAT-D, BDAT-C, BDAT-B, BDAT-A, BDAT-9, BDAT-8
U20 74LS165 — BDAT-7, BDAT-6, BDAT-5, BDAT-4, BDAT-3, BDAT-2, BDAT-1, BDAT-Ø
U28 74LS165 — BDAT-F, BDAT-E, BDAT-D, BDAT-C, BDAT-B, BDAT-A, BDAT-9, BDAT-8

U28 LS51 → ID NCMPR/ 7B8

10A1 R/W CLK-B
5A1 WDC-38/
5A1 WDC-30/
10B1 RD DATA/
10B1 RD DATA
5A1 WDC Ø8/
11A1 ALW SYNC SRCH
7B1 SYNC BYTE/
7B1 ID/ECC NCMPR/

SYNC FND/ 8C8
ID NCMPR-L 12C8

U29 74LS74

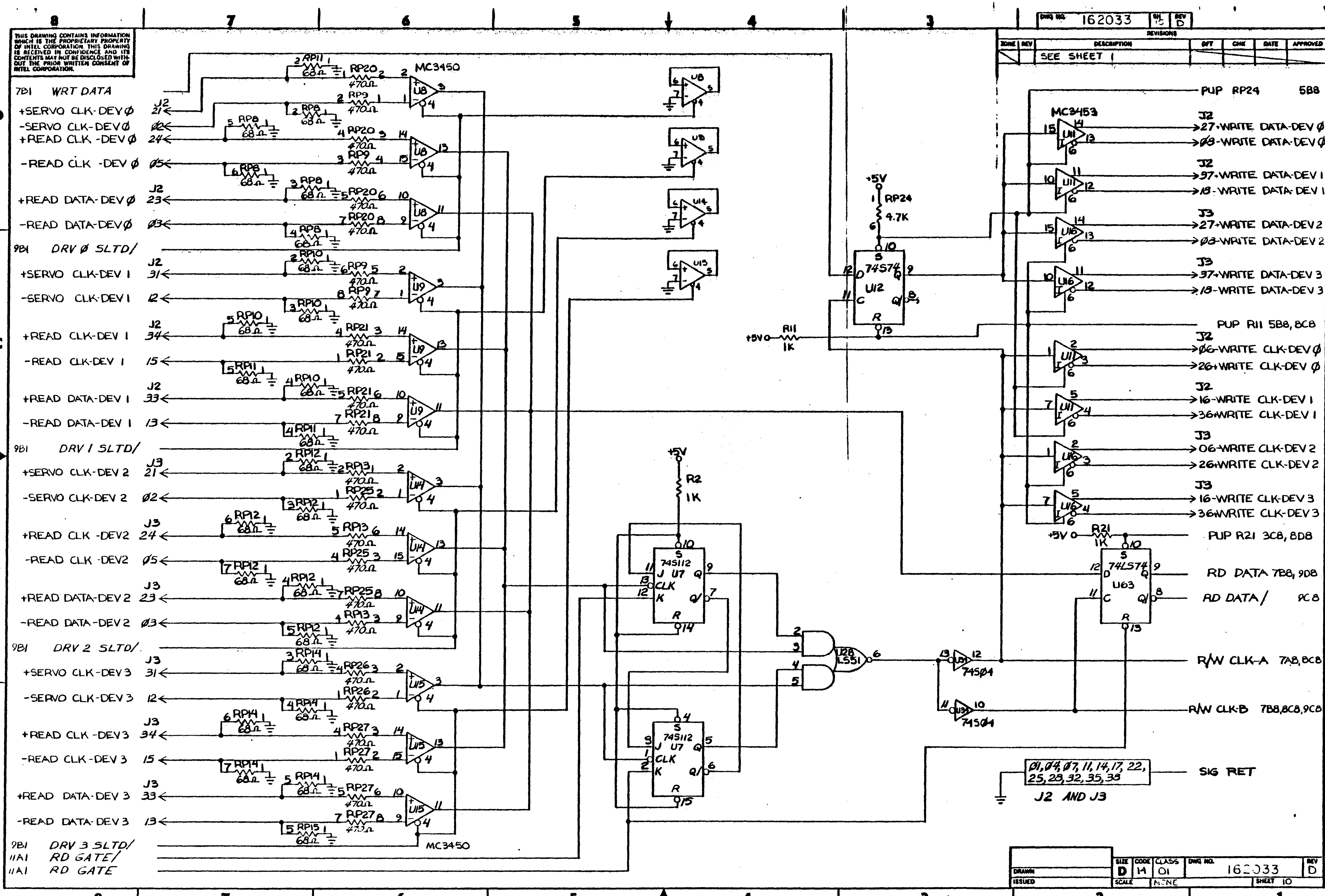
+5V — RP24 4.7KΩ

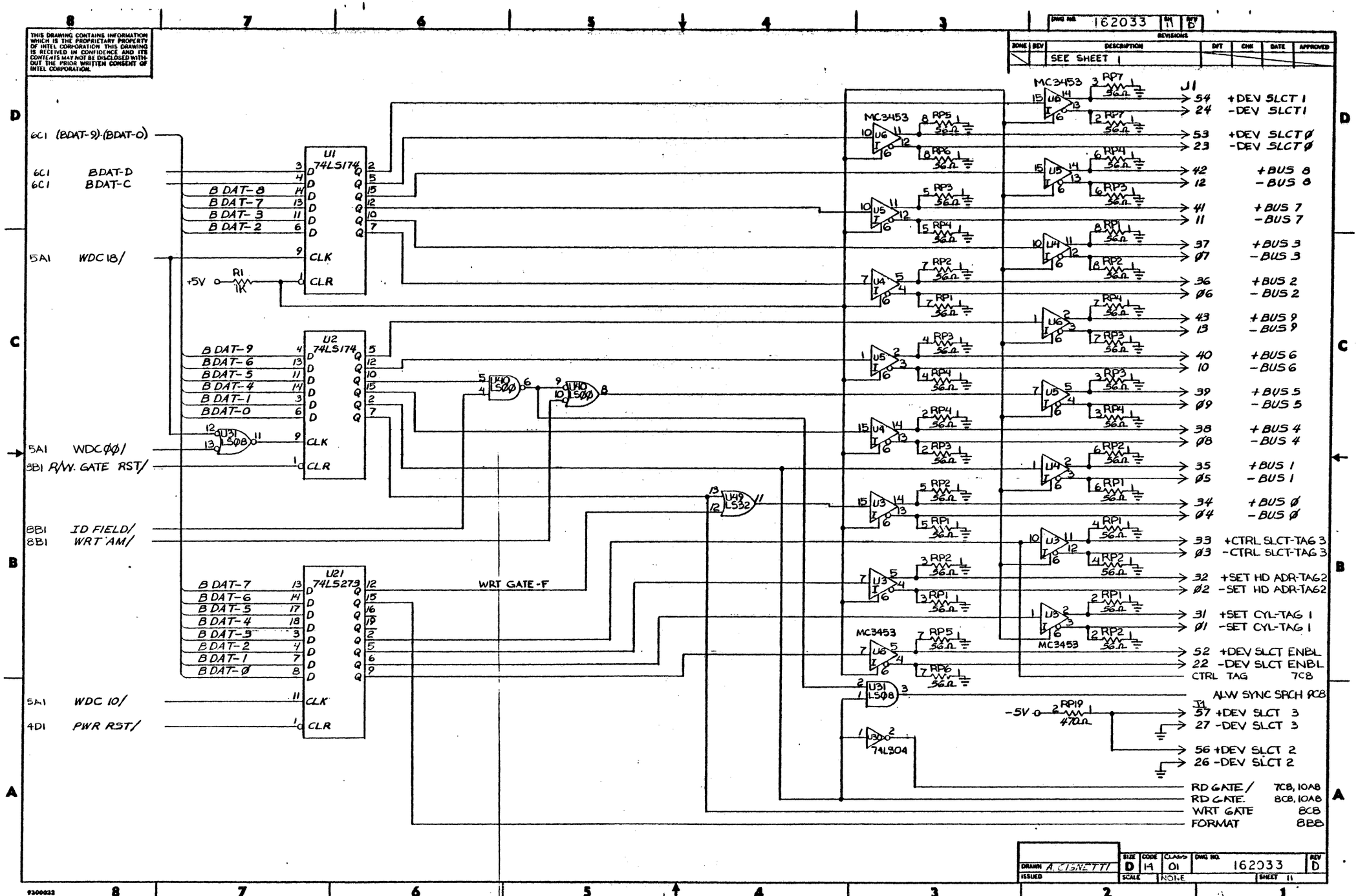
RP8, RP22 470Ω, RP23 470Ω, RP8 68Ω — MC3450 U10 → DRV Ø SLTD/ 10C8,12D8
RP10 68Ω, RP23 470Ω, RP23 470Ω, RP11 68Ω — MC3450 U10 → DRV 1 SLTD/ 10C8,12D8
RP12 68Ω, RP23 470Ω, RP13 470Ω, RP14 68Ω — MC3450 U10 → DRV 2 SLTD/ 10B8,12D8
RP15 68Ω, RP22 470Ω, RP22 470Ω, RP14 68Ω — MC3450 U10 → DRV 3 SLTD/ 10A8,12D8

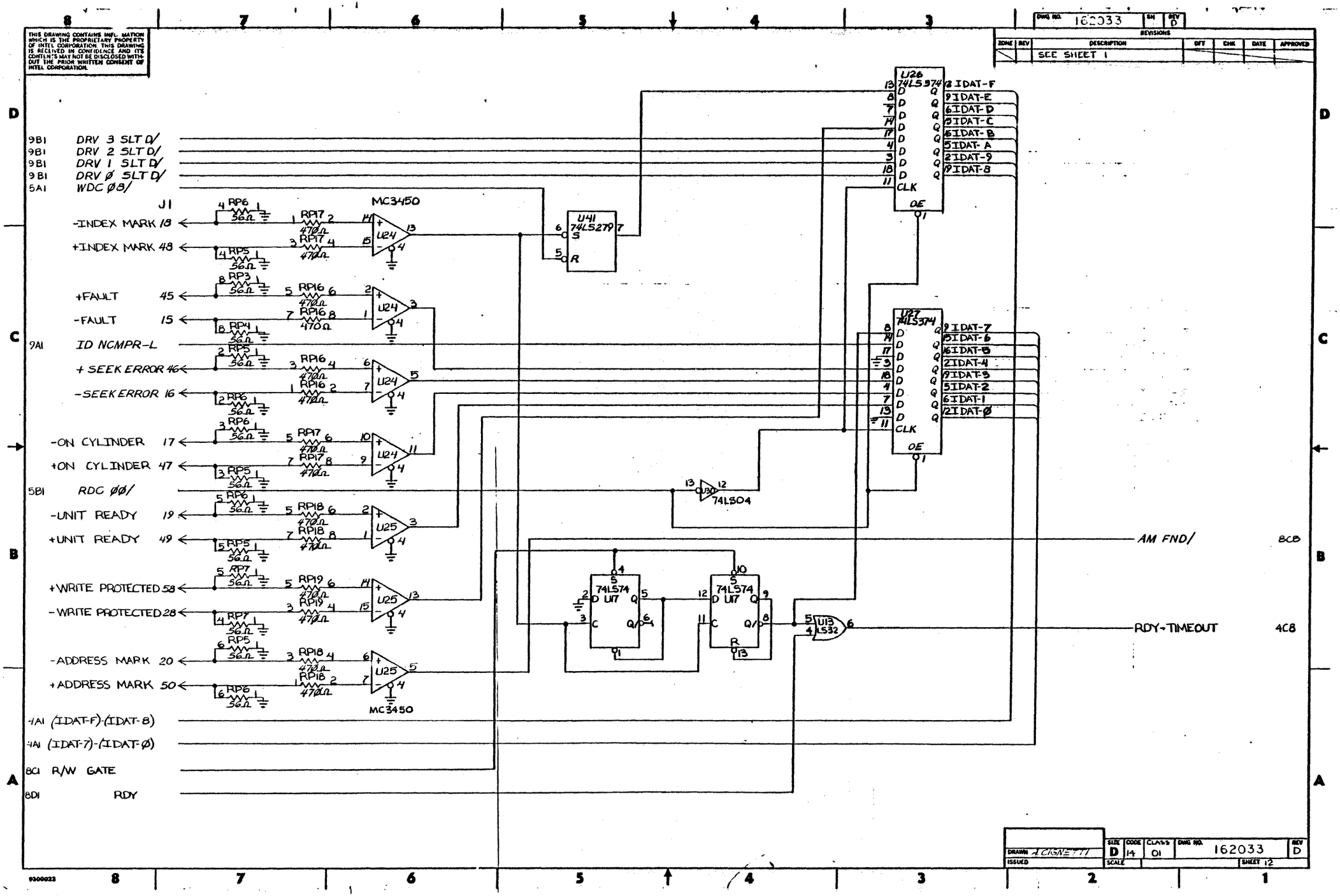
J2
-SELECTED-DEV Ø 29
+SELECTED-DEV Ø Ø9
J2
-SELECTED-DEV 1 39
+SELECTED-DEV 1 19
J3
-SELECTED-DEV 2 29
+SELECTED-DEV 2 Ø9
J3
-SELECTED-DEV 3 39
+SELECTED-DEV 3 19

DRAWN ACISNETTI	SIZE D	CODE 14	CLASS 01	DWG NO. 162033	REV D
ISSUED	SCALE 1-ONE			SHEET 9	

9200022


DWG NO. 162033 SH 10 REV D
REVISIONS
ZONE REV DESCRIPTION DFT CHK DATE APPROVED
SEE SHEET 1
7B1 WRT DATA
+SERVO CLK-DEV Ø J2 21
-SERVO CLK-DEV Ø Ø2
+READ CLK-DEV Ø 24
-READ CLK-DEV Ø Ø5
+READ DATA-DEV Ø J2 23
-READ DATA-DEV Ø Ø3
9B1 DRV Ø SLTD/
+SERVO CLK-DEV 1 J2 31
-SERVO CLK-DEV 1 12
+READ CLK-DEV 1 J2 34
-READ CLK-DEV 1 15
+READ DATA-DEV 1 J2 33
-READ DATA-DEV 1 13
9B1 DRV 1 SLTD/
+SERVO CLK-DEV 2 J3 21
-SERVO CLK-DEV 2 Ø2
+READ CLK-DEV2 J3 24
-READ CLK-DEV2 Ø5
+READ DATA-DEV 2 J3 23
-READ DATA-DEV 2 Ø3
9B1 DRV 2 SLTD/
+SERVO CLK-DEV3 J3 31
-SERVO CLK-DEV 3 12
+READ CLK-DEV3 J3 34
-READ CLK-DEV 3 15
+READ DATA-DEV 3 J3 33
-READ DATA-DEV 3 13
9B1 DRV 3 SLTD/
11A1 RD GATE/
11A1 RD GATE
MC3450
MC3453
74S74 U12
74S112 U7
LS51
74S04
74LS74 U63
+5V RP24 4.7K
R11 1K
R2 1K
R21 1K
PUP RP24 5B8
J2 27+WRITE DATA-DEV Ø
Ø8-WRITE DATA-DEV Ø
J2 37+WRITE DATA-DEV 1
18-WRITE DATA-DEV 1
J3 27+WRITE DATA-DEV 2
Ø8-WRITE DATA-DEV 2
J3 37+WRITE DATA-DEV 3
18-WRITE DATA-DEV 3
PUP R11 5B8, 8C8
J2 Ø6-WRITE CLK-DEV Ø
26+WRITE CLK-DEV Ø
J2 16-WRITE CLK-DEV 1
36+WRITE CLK-DEV 1
J3 Ø6-WRITE CLK-DEV 2
26+WRITE CLK-DEV 2
J3 16-WRITE CLK-DEV 3
36+WRITE CLK-DEV 3
PUP R21 3C8, 8D8
RD DATA 7B8, 9D8
RD DATA/ 9C8
R/W CLK-A 7A8, 8C8
R/W CLK-B 7B8, 8C8, 9C8
Ø1, Ø4, Ø7, 11, 14, 17, 22, 25, 28, 32, 35, 38
J2 AND J3
SIG RET
SIZE D CODE 14 CLASS 01 DWG NO. 162033 REV D
DRAWN
ISSUED
SCALE NONE SHEET 10
9300022


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.
DWG NO. 162033 REV D
REVISIONS
ZONE REV DESCRIPTION DFT CHK DATE APPROVED
SEE SHEET 1
6C1 (BDAT-9)-(BDAT-0)
6C1 BDAT-D
6C1 BDAT-C
5A1 WDC 18/
+5V R1 1K
U1 74LS174
B DAT-8
B DAT-7
B DAT-3
B DAT-2
CLK
CLR
U2 74LS174
B DAT-9
BDAT-6
BDAT-5
B DAT-4
B DAT-1
BDAT-0
U40 LS00
U41 LS00
U31 LS08
5A1 WDC ØØ/
8B1 R/W. GATE RST/
U49 LS32
8B1 ID FIELD/
8B1 WRT AM/
U21 74LS273
B DAT-7
B DAT-6
B DAT-5
B DAT-4
B DAT-3
B DAT-2
B DAT-1
B DAT-Ø
WRT GATE-F
5A1 WDC 10/
4D1 PWR RST/
U30 74LS04
MC3453
RP7 56Ω
RP5 56Ω
RP6 56Ω
RP4 56Ω
RP3 56Ω
RP1 56Ω
RP2 56Ω
U6
U5
U4
U3
RP19 470Ω
-5V
J1
54 +DEV SLCT 1
24 -DEV SLCT 1
53 +DEV SLCT Ø
23 -DEV SLCT Ø
42 +BUS 8
12 -BUS 8
41 +BUS 7
11 -BUS 7
37 +BUS 3
Ø7 -BUS 3
36 +BUS 2
Ø6 -BUS 2
43 +BUS 9
13 -BUS 9
40 +BUS 6
10 -BUS 6
39 +BUS 5
Ø9 -BUS 5
38 +BUS 4
Ø8 -BUS 4
35 +BUS 1
Ø5 -BUS 1
34 +BUS Ø
Ø4 -BUS Ø
33 +CTRL SLCT-TAG 3
Ø3 -CTRL SLCT-TAG 3
32 +SET HD ADR-TAG2
Ø2 -SET HD ADR-TAG2
31 +SET CYL-TAG 1
Ø1 -SET CYL-TAG 1
52 +DEV SLCT ENBL
22 -DEV SLCT ENBL
CTRL TAG 7C8
ALW SYNC SRCH 8C8
J1
57 +DEV SLCT 3
27 -DEV SLCT 3
56 +DEV SLCT 2
26 -DEV SLCT 2
RD GATE/ 7C8, 10A8
RD GATE 8C8, 10A8
WRT GATE 8C8
FORMAT 8B8
DRAWN A. CIGNETTI
ISSUED
SIZE D CODE 19 CLASS 01 DWG NO. 162033 REV D
SCALE NONE SHEET 11
9300023


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.
DWG NO. 162033 SH REV D
REVISIONS
ZONE REV DESCRIPTION DFT CHK DATE APPROVED
SEE SHEET 1
9B1 DRV 3 SLT D/
9B1 DRV 2 SLT D/
9B1 DRV 1 SLT D/
9B1 DRV Ø SLT D/
5A1 WDC ØB/
J1
-INDEX MARK 18
+INDEX MARK 48
+FAULT 45
-FAULT 15
9A1 ID NCMPR-L
+SEEK ERROR 46
-SEEK ERROR 16
-ON CYLINDER 17
+ON CYLINDER 47
5B1 RDC ØØ/
-UNIT READY 19
+UNIT READY 49
+WRITE PROTECTED 58
-WRITE PROTECTED 28
-ADDRESS MARK 20
+ADDRESS MARK 50
4A1 (IDAT-F)-(IDAT-8)
4A1 (IDAT-7)-(IDAT-Ø)
8C1 R/W GATE
8D1 RDY
MC3450
MC3450
U26 74LS374
U27 74LS374
U41 74LS279
74LS04
74LS74
74LS74
U13 LS32
AM FND/ 8C8
RDY-TIMEOUT 4C8
DRAWN A.CIGNETTI
SIZE D CODE 14 CLASS 01 DWG NO. 162033 REV D
ISSUED
SCALE
SHEET 12

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. 79 BEAVERTON, OR

POSTAGE WILL BE PAID BY ADDRESSEE

Intel Corporation
3585 S.W. 198th
Aloha, Oregon 97005

MCSO Technical Publications

INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, CA 95051 (408) 987-8080

Printed in U.S.A.