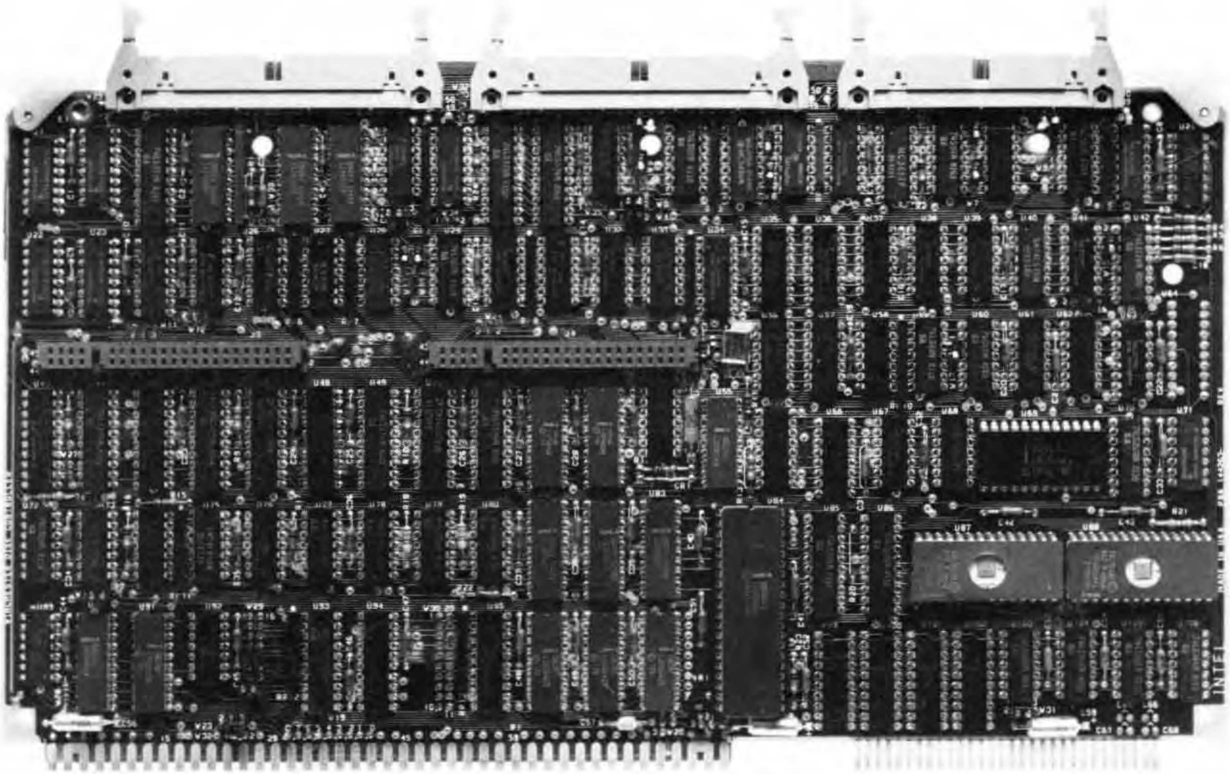
iSBC-215G = PBA 144263-014
iSBX-218A = PBA 145591-002
iSBX-217B = PBA 144287-008
iSBX-217C = PBA 146050-003

iSBC® 215 GENERIC WINCHESTER DISK CONTROLLER HARDWARE REFERENCE MANUAL



146979-001 and 146979-002
147931-001 and 147931-002
pwa 144263-016 ck

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

Order Number: 144780-002

iSBC® 215 GENERIC WINCHESTER DISK CONTROLLER HARDWARE REFERENCE MANUAL

Order Number: 144780-002

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

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

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

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

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

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

Intel Corporation makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

Intel retains the right to make changes to these specifications at any time, without notice.

Contact your local sales office to obtain the latest specifications before placing your order.

The following are trademarks of Intel Corporation and its affiliates and may be used only to identify Intel products:

BITBUS	iLBX	iPDS	OpenNET
COMMputer	im	iRMX	Plug-A-Bubble
CREDIT	iMDDX	iSBC	PROMPT
Data Pipeline	iMMX	iSBX	Promware
Genius	Insite	iSDM	QueX
△ 7	Intel	iSXM	QUEST
i	intel	Library Manager	Ripplemode
I²ICE	intelBOS	MCS	RMX/80
ICE	Intelevision	Megachassis	RUPI
iCS	inteligent Identifier	MICROMAINFRAME	Seamless
iDBP	inteligent Programming	MULTIBUS	SLD
iDIS	Intellec	MULTICHANNEL	SYSTEM 2000
	Intellink	MULTIMODULE	UPI
	iOSP		

MDS is an ordering code only and is not used as a product name or trademark. MDS® is a registered trademark of Mohawk Data Sciences Corporation.

*MULTIBUS is a patented Intel bus.

REV.	REVISION HISTORY	DATE
-001	Original Issue	7/82
-002	Manual updated to include new information for iSBX 217B/C board and iSBX 218A board support. Interconnect tables in chapter 2 are corrected and expanded. Additional programming information is provided in Chapter 3. New schematics diagram in Chaptér 5.	12/84

PREFACE

This manual includes several kinds of information. For introductory material on the iSBC 215 Generic Winchester Disk Controller Board, refer to Chapter 1. To configure the board and install it in your system, refer to Chapter 2. For programming information, refer to Chapter 3. For the functional description, refer to Chapter 4. For service assistance information or the schematic diagram, refer to Chapter 5. This manual is not intended as a tutorial document. The manual assumes that you are familiar with the standards of Intel single-board computers and the associated peripheral control boards and are familiar with programming in general and Intel device programming in particular.

In addition to this manual, you will need the manuals for the system of which the iSBC 215G board is a part. The following listed manuals provide information pertaining to the iSBC 215G board, its use, and its component parts. Intel documents are available from the Intel Literature Department (refer to page ii for the address).

Microsystem Components Handbook, Volumes I and II, Order No. 230843

Intel MULTIBUS® Handbook, Order No. 2108833
(This handbook includes information on the Intel iSBX™ Bus.)

8089 Assembler User's Guide, Order No. 9800938

8086 Family User's Manual, Order No. 9800722

Also, if you intend to use the iSBC 215G board with flexible-diskette and/or cartridge-tape drives, you will need one, two, or all of the following manuals.

iSBX™ 218A Flexible Diskette Controller Hardware Reference Manual, Order No. 145911

iSBX™ 217B Magnetic Cartridge Tape Interface MULTIMODULE™ Board Hardware Reference Manual, Order No. 145497

iSBX™ 217C Magnetic Cartridge Tape Interface MULTIMODULE™ Board Hardware Reference Manual, Order No. 146704

Tape interface support is limited to Archive Corporation and other QIC-02 tape drives. Programming information in this manual is current with and applicable to the following iSBC 215G, iSBX 218A, and iSBX 217B/C boards:

iSBC® 215G -- PBA Number: 144263-Ø14
iSBX™ 218A -- PBA Number: 145591-ØØ2
iSBX™ 217B -- PBA Number: 144287-ØØ8
iSBX™ 217C -- PBA Number: 146Ø5Ø-ØØ3

Differences between this version of the board firmware and earlier versions are noted as appropriate. Significant differences between operation of the iSBC 215G board and the iSBC 215A/B board are also noted as appropriate.

CONTENTS

PAGE

CHAPTER 3
PROGRAMMING (continued)

CONTENTS (continued)

PAGE

CONTENTS (continued)

PAGE

TABLES

PAGE

FIGURES

FIGURES (continued)

***

CHAPTER 1
GENERAL INFORMATION

1.1 INTRODUCTION

The Intel iSBC 215 Generic Winchester Disk Controller Board (referred to as the iSBC 215G board in this manual) allows as many as four hard-disk drives (typically, Winchester technology), as many as four flexible-diskette drives, and as many as four magnetic cartridge-tape drives to be interfaced to any Intel MULTIBUS interface compatible computer system. It supports disk drives that use open-loop head positioning, closed-loop head positioning, or ANSI X3T9/1226 interfaces. Figures 1-1 and 1-2 show examples of multiple hard-disk drive applications in non-ANSI and ANSI configurations, respectively.

The iSBC 215G board design is based on the Intel 8089 8/16-Bit HMOS I/O Processor, which features direct-memory-access (DMA) transfers, multiple-sector transfers, transparent error detection and correction (with automatic recovery and retry), and data management. The board operates in a multiprocessor environment and is fully compatible with all Intel 8- and 16-bit computers. The number of tracks per surface is software selectable for each drive unit. Seek operations on more than one drive can be overlapped with read/write operations on other drives. The iSBC 215G board is fully compatible with Intel 8086 16-Bit HMOS Microprocessor 20-bit addressing, and can be used in Intel MULTIBUS 24-bit address systems.

The board includes two Intel iSBX bus connectors, J3 and J4, that allow other storage devices such as flexible-diskette drives or magnetic cartridge-tape drives to be operated with MULTIBUS interface compatible systems. For example, the Intel iSBX 218A Flexible Diskette Controller Board attaches to iSBX bus connector J4, allowing the iSBC 215G board to control as many as four flexible-diskette drives. Figure 1-3 shows an example multiple-drive system using four 5 1/4- or 8-inch flexible-diskette drives, the iSBC 215G board, and the iSBX 218A Flexible Diskette Controller. As another example, the Intel iSBX 217B/C Magnetic Cartridge Tape Interface Board attaches to iSBX bus connector J3, allowing the iSBC 215G board to control as many as four tape drives.

1.2 DESCRIPTION

The iSBC 215G board is a single, multi-layer printed-circuit board assembly. It may be installed in any Intel backplane or custom-designed configuration that is physically and electrically compatible with the Intel MULTIBUS interface.

The host central processing unit (CPU) communicates with the iSBC 215G board via four blocks of information in host memory. Once the iSBC 215G board is initialized, a CPU I/O write to the board wake-up address initiates activities. The board accesses the four blocks in the host memory to determine the specific operation being performed, fetches the required parameters, and completes the specified operation without CPU intervention.

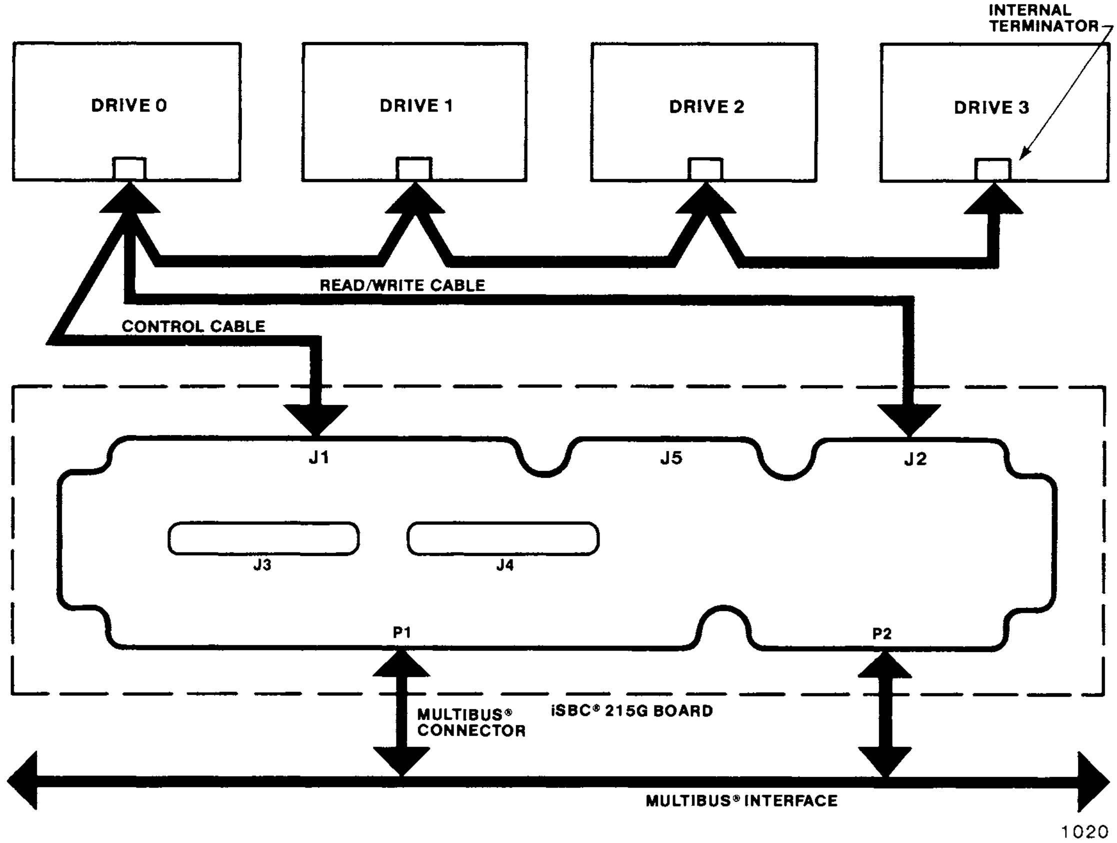


Figure 1-1. Example of Multiple Drive System Using Non-ANSI Interface

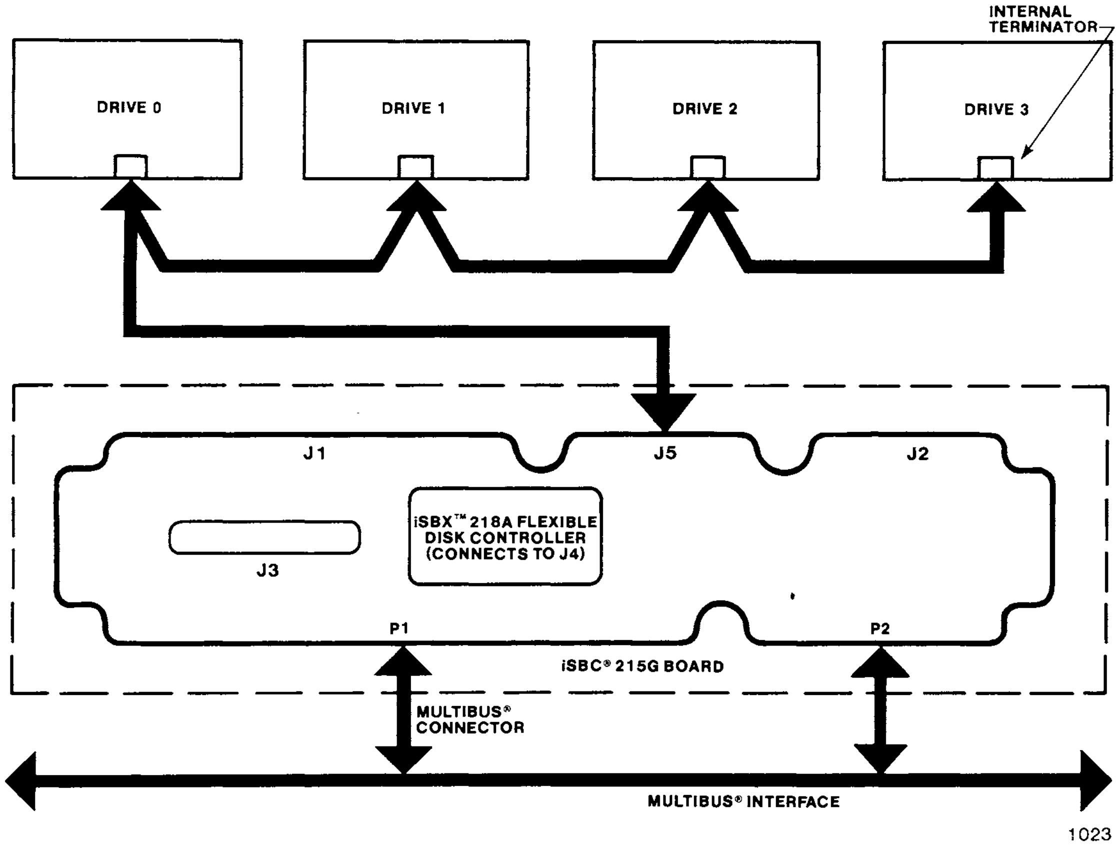


Figure 1-2. Example of Multiple Drive System Using ANSI Interface

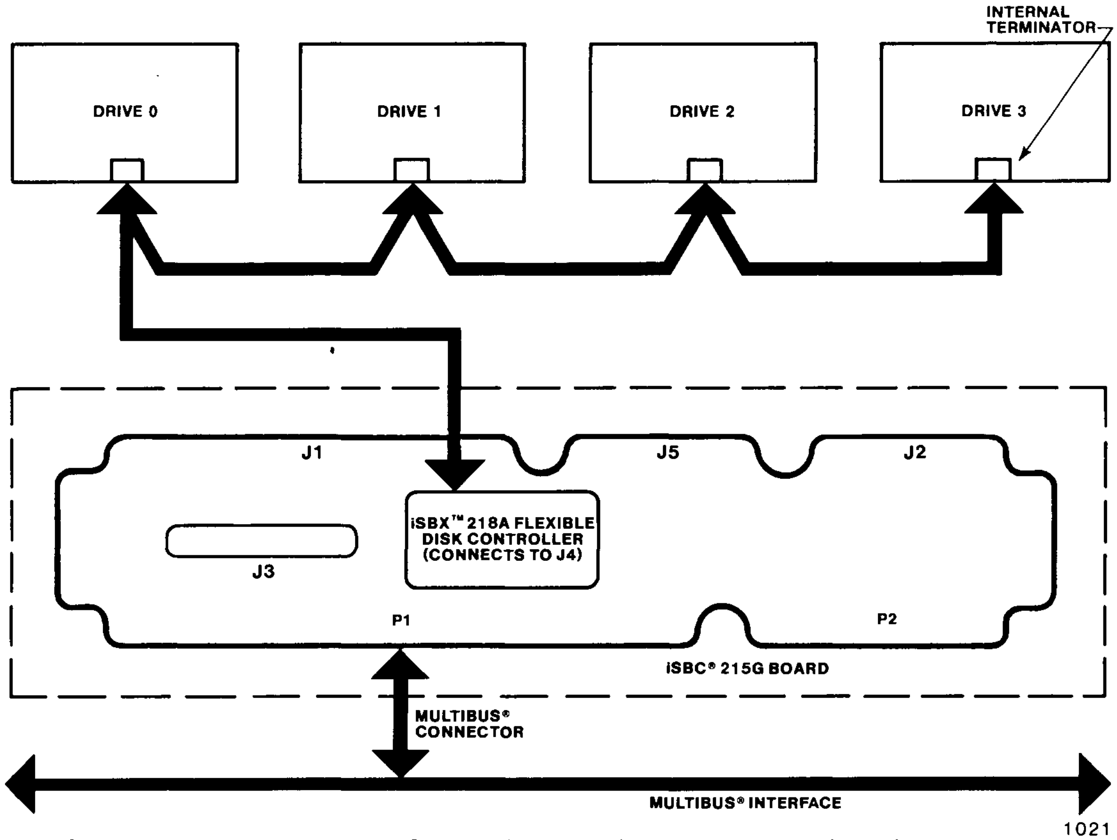


Figure 1-3. Example of Multiple Drive System Using iSBX™ 218A Flexible Diskette Controller

The iSBC 215G 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 board accepts serial data from the disk, interprets synchronizing bit patterns, verifies validity of the data, performs a serial-to-parallel data conversion, and passes parallel data or error condition indications to the CPU memory. During a disk write operation, the board performs parallel-to-serial data conversion and transmits serial write data and the write clock to the drive. As part of the disk format and write functions, the board appends an error checking code (ECC) at the end of each sector ID and data field. This ECC is used for checking and correcting data errors. It corrects all errors in bursts of as many as 11 bits, and detects all errors in bursts of as many as 32 bits (see Figure 1-4).

The Intel 8Ø89 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 2764 EPROM's provide on-board storage of the board I/O control program and a resident diagnostic exerciser, and Intel 2114 Static RAM's provide local memory data buffering and temporary storage for read/write parameters.

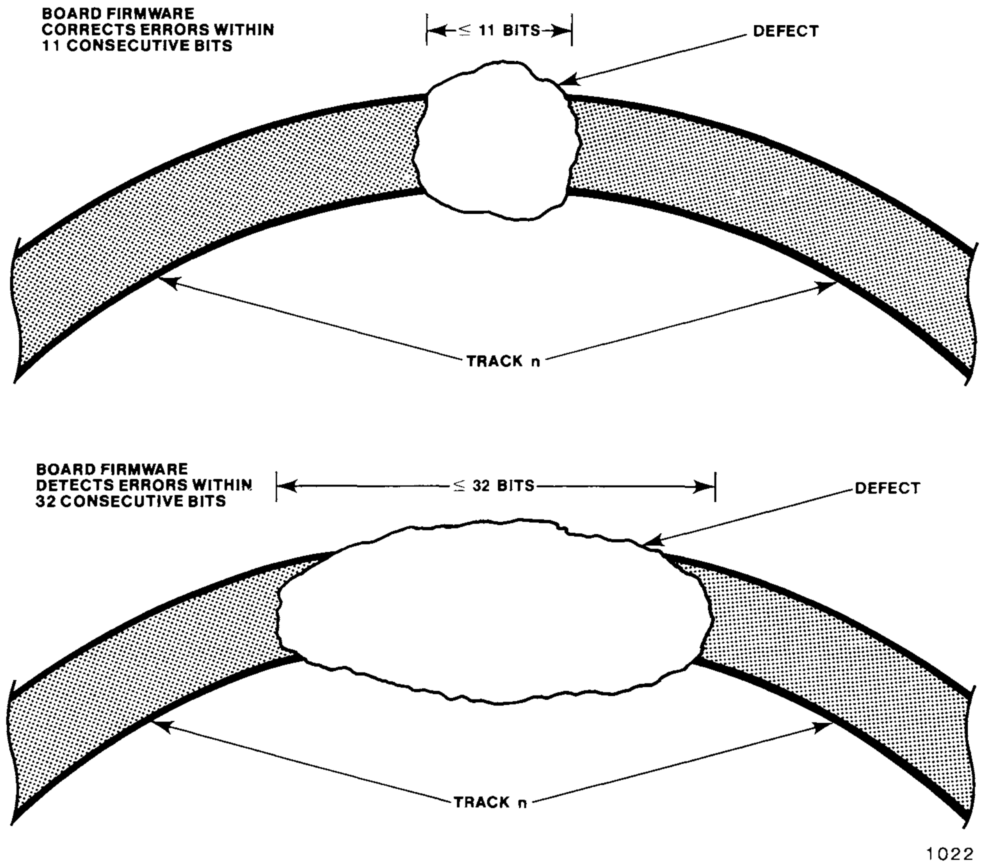


Figure 1-4. Automatic Error Checking and Correction

1.3 SPECIFICATIONS

Table 1-1 lists the specifications of the iSBC 215G board; Tables 1-2, 1-3, and 1-4 list typical characteristics of compatible disk drives. Note that the drives listed in Tables 1-2, 1-3, and 1-4 are representative only and are not qualified or endorsed by Intel, and that Intel assumes no responsibility to update or keep the list current.

Table 1-1. Board Specifications

Item	Specification
Processor	Any Intel mainframe or any MULTIBUS interface compatible CPU. (The iSBC 215G board can operate with 16- , 2Ø- , or 24-bit addresses and with 8- or 16-bit data bus widths.)
Drive Type	Disk Drives -- Either hard-disk (Winchester) or flexible-disk (through optional on-board iSBX Flexible Diskette Controller Board). Tape Drives -- Magnetic-cartridge, 1/4-inch, 9Ø- or 3Ø-ips QIC-Ø2 type drives (through optional on-board iSBX Magnetic Cartridge Tape Interface Board).
Drives per Controller	Hard-Disk Drives -- As many as two 5 1/4-inch or four 8-inch non-ANSI drives. As many as four ANSI X3T9/1226 drives. Flexible-Disk Drives -- As many as four 5 1/4- or 8-inch drives through iSBX 218A Flexible Disk Controller connected to iSBC 215G board iSBX bus connector J4. Tape Drives -- As many as four 1/4-inch magnetic cartridge-tape drives through iSBX 217B or C Magnetic Cartridge Tape Interface Board connected to iSBC 215G board iSBX bus connector J3.
Error Detection and Correction	Error detection to 32 bits in length; error correction to 11 consecutive bits in length.
Power Requirements	+5 V ±5 % @ 4.52 A maximum -5 V ±5 % @ Ø.Ø15 A maximum Note: Jumper selection and on-board voltage regulator allow use of -1Ø V or -12 V from MULTIBUS connector as alternate to direct -5 V source.
Mounting	Occupies one card slot in MULTIBUS compatible card cage or backplane connector. Occupies two slots in most card cages when optional iSBX MULTIMODULE is installed.

Table 1-1. Board Specifications (continued)

Item	Specification	
Physical		
Width	17.2 cm (6.8 inches)	
Length	3Ø.5 cm (12.Ø inches)	
Thickness	1.3 cm (Ø.5 inch)	
Weight	Ø.54 kg (19 ounces)	
Environment	Operating	Non-Operating
Temperature	Ø°C to +55°C (+32°F to +131°F)	-4Ø°C to +7Ø°C (-4Ø°F to +158°F)
Humidity	5 to 9Ø %, non-condensing	5 to 95 %, non-condensing

Table 1-2. Allowable Sectors Per Track for Non-ANSI Hard-Disk Drives

Disk Drive	Data Bytes per Sector			
	128	256	512	1Ø24
Priam 8-in	72	42	23	12
Priam 14-in	1Ø7	63	35	18
Ampex, RMS, CMI, Shugart, Quantum	54	31	17	9
Fujitsu, Memorex	64	38	21	11
Shugart 14-in	96	57	31	16
CDC	64	41	23	12

Table 1-3. Allowable Sectors Per Track for ANSI Hard-Disk Drives

Disk Drive	Data Bytes per Sector			
	128	256	512	1024
3M	82	51	29	16
Kennedy, BASF	74	43	23	12
Micropolis	71	44	25	13
Pertec	85	52	29	15

Table 1-4. Formatted Capacity Per Hard-Disk Drive (in Mbytes)

Disk Drive	Data Bytes per Sector			
	128	256	512	1024
Shugart SA 1004	7.08	8.12	8.91	9.43
Quantum Q2010	7.08	8.12	8.91	9.43
Shugart SA4008	19.86	23.58	25.65	26.48
Priam 3450, 9410-32	23.96	27.94	30.62	31.95
Fujitsu 2300, Memorex 101	7.99	9.49	10.49	10.98
RMS 512, CMI	8.40	9.65	10.58	11.21
CDC	19.50	24.98	28.03	29.25
3M 8432	11.76	14.62	16.63	18.35
Kennedy, BASF 6172	17.45	20.28	21.69	22.63
Micropolis 1203	26.65	32.67	37.12	38.60
Pertec D8035 (ANSI)	21.85	26.73	29.81	30.84

***

CHAPTER 2
PREPARATION FOR USE

2.1 INTRODUCTION

This chapter provides information for use in preparing and installing the iSBC 215G board. Included are instructions for unpacking and inspection, installation, installing jumpers, connecting the board to the MULTIBUS interface, and preparing and connecting cabling to the disk drives.

2.2 UNPACKING AND INSPECTION

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

For repair of a product damaged during shipment, contact the Intel Product Service Center to obtain a Return Authorization Number and further instructions. (Chapter 5 lists the telephone numbers for the various centers.) A purchase order is required to complete the repair. Submit a copy of the purchase order to the carrier with your claim.

2.3 INSTALLATION CONSIDERATIONS

The iSBC 215G board can be installed in any Intel cardcage/backplane or any user-designed backplane that meets the MULTIBUS interface specification. The board occupies one backplane slot. An additional slot may be required if an iSBX MULTIMODULE is installed.

Because the iSBC 215G board operates as a system master, the slot into which it is installed must include bus priority arbitration capability. Priority resolution can be done in either serial or parallel fashion.

2.3.1 POWER REQUIREMENTS

The board requires a source of +5 V ±5 % power, at a maximum current of 4.52 A. This is supplied through the MULTIBUS connector. When interfacing with 8-inch Shugart and Quantum drives, an additional source of -5 V ± 5 % power, at a maximum current of 15 mA, is required. This supply can be obtained directly from the MULTIBUS connector or from an on-board regulator that uses the MULTIBUS -10 V or -12 V source.

When interfacing with an iSBX bus through J3 or J4, additional voltage sources of +12 V, -12 V, or both, may be required. These can also be supplied through the MULTIBUS connector (see the individual iSBX Board specifications for tolerances and current requirements). Before installing the iSBC 215G board in a system chassis, make certain that the associated power supplies can supply the required additional current.

NOTE

If power is applied to or removed from the iSBC 215G board while a drive is ready, a spurious disk write operation could occur. To prevent this, always make certain that the drives are not turning when the iSBC 215G board power is switched on or off.

2.3.2 COOLING REQUIREMENTS

The iSBC 215G board (with no iSBX boards installed) dissipates 338.2 gram calories of heat (1.34 Btu's) per minute. Sufficient cooling air circulation (approximately 2ØØ linear feet per minute under ordinary conditions) must be provided to keep the board within the required operating temperature range (Ø to 55° C).

2.3.3 PHYSICAL CHARACTERISTICS

The dimensions, outline, and connector and jumper locations of the iSBC 215G board are shown in Figure 5-2.

2.4 JUMPER CONFIGURATIONS AND JUMPER INSTALLATION

Various configurations of the iSBC 215G board can be accommodated through the jumper stake pins provided on the board. A variable number of jumpers may be installed by the user on pairs of these stake pins to conveniently set up the board for the system environment in which it is to operate (8- or 16-bit system data bus; 16- ,2Ø- , or 24-bit addressing, etc.) and for the type of device to which it is to be interfaced (Shugart, Quantum, Memorex, etc. drive, or iSBX board). The default configuration includes approximately 4Ø jumpers, 2 of which are soldered and 1 wire-wrapped in place.

Each jumper is identified by its "W" number and the numbers of the two stake pins used (for example: W21-1 -- 2, or W3Ø-1 -- 2Ø). For jumper stake pin physical locations and details on jumper layouts, refer to Figure 5-2. In Figure 5-2, a § symbol following a jumper number denotes the default configuration. The board should be configured, as described in the following paragraphs, before its installation in a system.

NOTE

An asterisk or slash following a signal mnemonic denotes that the signal is active when in the low state.

2.4.1 WAKE-UP ADDRESS SELECTION

The iSBC 215G board communicates with the host CPU through four I/O communication blocks located in the host memory. The board receives instructions by reading the contents of the beginning address of the first I/O communication block. These contents are called the wake-up address, and may be at any address for a 2Ø- or 16-bit host system. Omitting jumper W36-1 -- 2 allows a 24-bit address host system to place the I/O communication blocks in the first 1-Mbyte page (that is, address ØXXXXXH); installing jumper W36-1 -- 2 allows placement in the last page (that is, address FXXXXXH). If the host CPU does not provide 24-bit addressing, or if the backplane does not provide for the upper four address lines, this jumper must not be installed.

Sixteen stake pin pairs are provided on the iSBC 215G board to allow the user to set the wake-up address. Eight of the pairs are identified as jumper W29. The other eight are identified as jumper W3Ø, which also includes two more pairs. (Of the additional W3Ø pairs, one specifies an 8-bit or 16-bit wake-up I/O port address and one specifies the system data bus width - see Sections 2.4.2 and 2.4.3.) The function, number, and location of each jumper are shown in Table 2-1 and Figure 5-2. An installed jumper represents a logical 1.

Table 2-1. Wake-Up Address Jumpers

Jumper		Wake-Up Address Bit
From	To	
W29-1	W29-16	F
W29-2	W29-15	E
W29-3	W29-14	D
W29-4	W29-13	C
W29-5	W29-12	B
W29-6	W29-11	A
W29-7	W29-1Ø	9
W29-8	W29-9	8
W3Ø-3	W3Ø-18	7
W3Ø-4	W3Ø-17	6
W3Ø-5	W3Ø-16	5
W3Ø-6	W3Ø-15	4
W3Ø-7	W3Ø-14	3
W3Ø-8	W3Ø-13	2
W3Ø-9	W3Ø-12	1
W3Ø-1Ø	W3Ø-11	Ø

The board 8089 I/O processor (IOP) treats the wake-up address as the segment portion of the standard segment and offset 20-bit addressing. For the wake-up address, the IOP uses an offset of 0. This multiplies the settings of the wake-up address jumpers by 2^4 (that is, it shifts the number four places to the left) to create a 20-bit wake-up address from 16-bits.

2.4.2 WAKE-UP I/O PORT ADDRESS SELECTION

The host CPU communicates with the iSBC 215G board through an I/O port, the number of which is also set by the wake-up address jumpers. For a host CPU with 8-bit I/O port addressing, bits 0 through 7 of the wake-up address determine the wake-up I/O port number; for a host CPU with 16-bit I/O port addressing, bits 0 through F determine the port number. Jumper W30-2 -- 19 (see Figure 5-2) determines the type of I/O port addressing used by the host CPU. It is installed for use with a 16-bit host CPU such as the Intel 8086; not installed for use with an 8-bit host CPU such as the Intel 8085.

2.4.3 SYSTEM DATA BUS WIDTH SELECTION

System data bus width selection jumper W30-1 -- 20 (see Figure 5-2) sets the board for the type of system data bus with which the iSBC 215G board is to interface. It is installed for a 16-bit data path, not installed for an 8-bit data path. Installing the jumper allows use of 16-bit data transfer mode to access the system bus (if the system memory supports 16-bit accesses), even though the host CPU supports only 8-bit accesses.

2.4.4 INTERRUPT PRIORITY NUMBER

The iSBC 215G board internal interrupt request signal can be assigned to any of eight MULTIBUS interrupt priority numbers (0* through INT7*). The number is selected by wire wrapping two jumper stake pins (see Figure 5-2) together as indicated in Table 2-2.

Table 2-2. Interrupt Priority Number Selection

Interrupt Number	Install Wire-Wrap Jumper	
	From Stake Pin:	To Stake Pin:
0	W19-C	W19-0
1	W19-C	W19-1
2	W19-C	W19-2
3	W19-C	W19-3
4	W19-C	W19-4
5	W19-C	W19-5
6	W19-C	W19-6
7	W19-C	W19-7

2.4.5 BUS PRIORITY ARBITRATION

Bus priority arbitration controls the sequence in which access is allowed to the MULTIBUS interface. Access priority is determined by three signals in combination: ANYRQST, CBRQ*, and BPRO*. These are described in the following paragraphs.

2.4.5.1 Common Bus Request (CBRQ*)/Any Request (ANYRQST) Signal Selection

The CBRQ* and ANYRQST signals provide the required mode select inputs to the 8289 Bus Arbiter. The arbitration options are shown in Table 2-3.

CBRQ* is a bi-directional interface signal that improves bus access time by allowing a bus master to retain control of the MULTIBUS interface without contending for it on each transfer cycle, as long as no other master is requesting control of the bus. The signal is either supplied from the bus via connector P1 or connected to ground, dependent upon the position of jumper W23. This signal operates the same in parallel and serial priority resolution schemes.

ANYRQST is a bus arbiter input signal that controls whether the iSBC 215G board will allow a lower-priority device to gain access to the MULTIBUS interface by the CBRQ* signal. The signal is either high (connected to +5 V through a resistor), or low (connected to ground), dependent upon the position of jumper W18. When ANYRQST is high, a lower-priority device may gain control of the bus by activating the CBRQ* signal. When ANYRQST is low, a lower-priority device cannot gain control of the bus until it gains priority through the BPRN* signal.

Table 2-3. Bus Arbitration Options

Signal	Jumper	Connect To	Description
CBRQ* ANYRQST	W23-1 -- 2 and W18-1 -- 3	Bus Ground	Arbitrate to gain access to MULTIBUS interface. If continued access is required, iSBC 215G board retains control until higher-priority device requests bus, at which time board arbitrates again and surrenders bus control to only that device.
CBRQ* ANYRQST	W23-1 -- 2 and W18-1 -- 2	Bus +5 V	Arbitrate to gain access to MULTIBUS interface. If continued access is required, iSBC 215G board retains control until another device requests bus, at which time board arbitrates again and surrenders bus control to requesting device (either higher or lower priority).
CBRQ* ANYRQST	W23-1 -- 3 and W18-1 -- 2	Ground +5 V	Arbitrate for every bus access.

2.4.5.2 Bus Priority Out (BPRO*) Signal Selection

The BPRO* signal is used in serial MULTIBUS priority schemes. BPRO* must be connected to the BPRN* input of the bus master with the next lower priority. The BPRO* signal is enabled for serial resolution by installing jumper W28-1 -- 2 (see Figure 5-2), or disabled for parallel resolution by omitting the jumper.

2.4.6 MULTIBUS® LOCK (LOCK*) SIGNAL

The LOCK* signal is used by the current bus master to exclude a dual-port RAM from use through the alternate port (for instance, the iSBC 86/35 single board computer, the iSBC Ø12CX memory board, etc.) when a multi-transfer operation (for instance, a read-modify-write) is required. The LOCK* signal is enabled by installing jumper W32-1 -- 2 (see Figure 5-2), or disabled by omitting the jumper.

2.4.7 iSBX™ BUS SELECTION

The iSBX bus control jumpers, W3, W4, W11, W12, and W24 (see Figure 5-2) select the external-terminate and DMA-request lines on the iSBX bus as shown in Table 2-4. Instructions are included in Chapter 3 for writing iSBC 215G board-to-drive interface software for I/O modules designed to iSBX Bus Specifications.

Table 2-4. iSBX™ Bus Control Jumpers

Jumper	Installed?	Function
W3-1 -- 2	Yes	EXT TRM -- External terminate (J3); terminated on iSBC 215G board.
	No	EXT TRM -- External terminate (J3); driven by iSBX I/O controller.
W4-1 -- 2	Yes	EXT TRM -- External terminate (J4); terminated on iSBC 215G board.
	No	EXT TRM -- External terminate (J4); driven by iSBX I/O controller.
W11-1 -- 2	Yes	OPØØ -- Option Ø (J3) driving.
	No	OPØØ -- Option Ø (J3) receiving.
W11-1 -- 3	Yes	OPØ1 -- Option Ø (J4) driving.
	No	OPØ1 -- Option Ø (J4) receiving.

Table 2-4. iSBX™ Bus Control Jumpers (continued)

Jumper	Installed?	Function
W12-1 -- 2	Yes	OP1Ø -- Option 1 (J3) driving.
	No	OP1Ø -- Option 1 (J3) receiving.
W12-1 -- 3	Yes	OP11 -- Option 1 (J4) driving.
	No	OP11 -- Option 1 (J4) receiving.
W24-1 -- 2	Yes	DREQØ -- iSBX controller on J4 uses DMA request; iSBX controller on J3 does not use DMA request or is not installed.
W24-1 -- 3	Yes	DREQ1 -- iSBX controller on J3 uses DMA request; iSBX controller on J4 does not use DMA request or is not installed.
W24-1 -- 2 and W24-1 -- 3	No	DREQØ/DREQ1 -- Both iSBX controllers use DMA requests, or neither uses DMA requests, or neither is installed.

2.4.8 HARD-DISK DRIVE INTERFACE

The iSBC 215G board is designed to communicate with either ANSI compatible (X3T9/1226) or proprietary non-ANSI hard-disk (Winchester technology) drive interfaces. It can control as many as four disk drives, except for certain units (for instance, Memorex, Shugart 14-inch, Priam, or CDC Finch Series). Two drives are supported for the excepted types. In all instances, drives from only one manufacturer at a time may be used, unless the drives are 1ØØ-percent compatible.

The jumpers listed in Table 2-5 allow the user to configure the iSBC 215G board for the listed drive types (see Figure 5-2). Other drive types may be used; however, Intel assumes no obligation to determine the appropriate jumper configuration. Interface cables must also be constructed and installed (according to the type of drive being used) as described later in this chapter.

Table 2-5. Jumper Configuration for Various Hard-Disk Drives

VENDOR CONFIGURATION TABLE

		DRIVE INTERFACE											
WIRE NO.	FUNCTION	SHUGART SA 1000 RMS 5 1/4 CMI 5 1/4	SHUGART SA 4000 MEMOREX FIJITSU	PRIAM	PRIAM ANSI	PERTEC SOFT-SECTOR ANSI	3M ANSI	MICROPOLIS ANSI	BASF ANSI	SLI ANSI	CDC	ANSI PROGRAM HARD SECTORED	ANSI PROGRAM SOFT SECTORED
W1	CMD BUS ENB*	1-3	1-3	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-3	1-2	1-2
W2	VENDOR Ø	---	1-2	1-2	1-2	---	1-2	1-2	1-2	1-2	1-2	1-2	---
W5	RDØ	1-3	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-3	1-2	1-2
W6	RDØ+	1-3	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-3	1-2	1-2
W7	RDCLØ	1-3	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-3	1-2	1-2
W8	RDCLØ+	1-3	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-3	1-2	1-2
W9	TRIPOLAR * (SA 1ØØØ)	1-2	---	---	---	---	---	---	---	---	1-2	---	---
W10	RADIAL SELECT	1-2	---	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2
W13	RD GATE	1-2	1-3	1-3	1-3	1-2	1-3	1-3	1-3	1-3	1-3	1-3	1-2
W14	AM CONTROL (SA 1ØØØ)	1-2	1-3	1-3	1-3	1-3	1-3	1-3	1-3	1-3	1-3	1-3	1-3
W15	SHUGART (GAP CONTROL)	---	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2
W16	HARD/SOFT SECTORING	1-2	1-3	1-3	1-3	1-2	1-3	1-3	1-3	1-3	1-3	1-3	1-2
W17	INDEX SELECT	1-2	1-2	---	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2
W22	RD CL	1-2	1-2	1-2	1-3	1-3	1-3	1-3	1-3	1-3	1-2	1-3	1-3
W26	VENDOR 1	---	1-2	1-2	---	---	1-2	---	1-2	---	1-2	1-2	1-2
W27	VENDOR 2	1-2	---	1-2	---	1-2	---	---	---	1-2	---	1-2	1-2
W37	VENDOR 3	1-2	1-2	1-2	---	---	---	1-2	1-2	1-2	1-2	---	---
W38	VENDOR 4	1-2	1-2	1-2	---	---	---	---	---	---	1-2	1-2	1-2
W33	SECTOR	1-2	1-2	1-2	1-3	1-2	1-3	1-3	1-3	1-3	1-2	1-3	1-2
W34	SKCOM *	---	---	---	1-2	1-2	1-2	1-2	1-2	1-2	---	1-2	1-2
W35	RDY *	---	---	---	1-2	1-2	1-2	1-2	1-2	1-2	---	1-2	1-2
W20	-10/-12V	1-2	---	---	---	---	---	---	---	---	---	---	---
W21	-5V	1-3	---	---	---	---	---	---	---	---	---	---	---

2.4.9 -5 Volt SELECTION

For interfacing with drives that require -5 V power (8-inch Shugart or Quantum or CDC drives), the power source and regulator voltage source must be selected. Install the two jumpers (see Figure 5-2) as indicated in Table 2-6 to select: 1) -5 V from either the MULTIBUS connector or the on-board regulator, and 2) the voltage source for the regulator.

Table 2-6. -5 V Selection Jumper Configuration

Install Jumper:	For:
W21-1 -- 2	-5 V from MULTIBUS interface
W21-1 -- 3	-5 V from regulator
W2Ø-1 -- 2	-12 V to regulator
W2Ø-1 -- 3	-1Ø V to regulator

2.4.10 I/O COMMUNICATION BLOCKS PAGE SELECTION

In the default configuration, all I/O communication blocks are located in the lowest 1-Mbyte page of the 16-Mbyte MULTIBUS address space. The user can select the highest page by installing jumper W36-1 -- 2 (see Figure 5-2).

2.4.11 RAM JUMPER

Jumper W31 is factory default connected as W31-1 to W31-2. It is not re-configurable.

2.5 INTERFACE CONNECTIONS

The iSBC 215G board communicates with the CPU and other boards via the MULTIBUS interface (connectors P1 and P2), and with the various storage drives via special cables (connector J1, J2, or J5, as appropriate).

2.5.1 MULTIBUS® INTERFACE

All interconnections between the iSBC 215G board and the MULTIBUS interface are accomplished through the two MULTIBUS edge connectors, P1 and P2. Tables 2-7 and 2-8 list the pins and signals for connectors P1 and P2, respectively. Tables 2-9 and 2-1Ø describe the signals listed for the connector pins. With reference to Tables 2-7 through 2-1Ø, see the considerations below.

1. All odd-numbered pins (1,3,5, etc.) are on the component side of the board; even-numbered pins are opposite. Pin 1 is the left-most pin when viewed from the component side with the extractors at the top.

2. Cable and board connector numbering convention may not agree.

3. An asterisk or slash following a signal mnemonic denotes that the signal is active when in the low state.

Table 2-7. Connector P1 Pin Assignments

	(Component Side)			(Circuit Side)		
	Pin	Mnemonic	Description	Pin	Mnemonic	Description
Power Supplies	1	GND	Signal Ground	2	GND	Signal Ground
	3	+5 V	+5 V Supply	4	+5 V	+5 V Supply
	5	+5 V	+5 V Supply	6	+5 V	+5 V Supply
	7	+12 V	+12 V Supply	8	+12 V	+12 V Supply
	9	-5 V	-5 V Supply	1Ø	-5 V	-5 V Supply
	11	GND	Signal Ground	12	GND	Signal Ground
Bus Controls	13	BCLK*	Bus Clock	14	INIT*	Initialize
	15	BPRN*	Bus Priority In	16	BPRO*	Bus Priority Out
	17	BUSY*	Bus Busy	18	BREQ*	Bus Request
	19	MRDC*	Mem Read Cmd	2Ø	MWTC*	Mem Write Cmd
	21	IORC*	I/O Read Cmd	22	IOWC*	I/O Write Cmd
	23	XACK*	XFER Acknowledge	24	INH1*	Inh 1 Dis RAM
Bus Controls and Address	25	LOCK*	Bus Lock	26	INH2*	Inh 2 Dis ROM
	27	BHEN*	Byte High Enable	28	AD1Ø*	
	29	CBREQ*	Common Bus Request	3Ø	AD11*	Address
	31	CCLK*	Constant Clock	32	AD12*	Bus
	33	INTA*	Interrupt Ack	34	AD13*	
Interrupts	35	INT6*	Parallel	36	INT7*	Parallel
	37	INT4*	Interrupt	38	INT5*	Interrupt
	39	INT2*	Requests	4Ø	INT3*	Requests
	41	INTØ*		42	INT1*	
Address	43	ADRE*		44	ADRF*	
	45	ADRC*		46	ADRD*	
	47	ADRA*		48	ADRB*	
	49	ADR8*	Address	5Ø	ADR9*	Address
	51	ADR6*	Bus	52	ADR7*	Bus
	53	ADR4*		54	ADR5*	
	55	ADR2*		56	ADR3*	
	57	ADRØ*		58	ADR1*	
Data	59	DATE*		6Ø	DATF*	
	61	DATC*		62	DATD*	
	63	DATA*		64	DATB*	
	65	DAT8*	Data	66	DAT9*	Data
	67	DAT6*	Bus	68	DAT7*	Bus
	69	DAT4*		7Ø	DAT5*	
	71	DAT2*		72	DAT3*	
	73	DATØ/		74	DAT1/	
Power Supplies	75	GND	Signal Ground	76	GND	Signal Ground
	77	-1Ø V	-1Ø V Supply	78	-1Ø V	-1Ø V Supply
	79	-12 V	-12 V Supply	8Ø	-12 V	-12 V Supply
	81	+5 V	+5 V Supply	82	+5V	+5 V Supply
	83	+5 V	+5 V Supply	84	+5V	+5 V Supply
	85	GND	Signal Ground	86	GND	Signal Ground

Table 2-8. Connector P2 Pin Assignments

(Component Side)				(Circuit Side)		
	Pin	Mnemonic	Description	Pin	Mnemonic	Description
	1		Not Connected	2		Not Connected
	3		Not Connected	4		Not Connected
Power	5		Not Connected	6		Not Connected
Supplies	7		Not Connected	8		Not Connected
	9		Not Connected	10		Not Connected
	11		Not Connected	12		Not Connected
	13		Not Connected	14		Not Connected
	15		Not Connected	16		Not Connected
	17		Not Connected	18		Not Connected
	19		Not Connected	20		Not Connected
	21		Not Connected	22		Not Connected
	23		Not Connected	24		Not Connected
	25		Not Connected	26		Not Connected
	27		Not Connected	28		Not Connected
	29		Not Connected	30		Not Connected
	31		Not Connected	32		Not Connected
	33		Not Connected	34		Not Connected
	35		Not Connected	36		Not Connected
	37		Not Connected	38		Not Connected
	39		Not Connected	40		Not Connected
	41		Not Connected	42		Not Connected
	43		Not Connected	44		Not Connected
	45		Not Connected	46		Not Connected
	47		Not Connected	48		Not Connected
	49		Not Connected	50		Not Connected
	51		Not Connected	52		Not Connected
	53		Not Connected	54		Not Connected
Address	55	ADR16*	Address Bus	56	ADR17*	Address Bus
	57	ADR14*	Address Bus	58	ADR15*	Address Bus
	59		Not Connected	60		Not Connected

Table 2-9. Connector P1 Input/Output Signals

Signal Mnemonic	Description
ADRØ* -- ADR13*	Address Bits: Specify part of memory address or I/O port address to be accessed. ADRØ* through ADR13* are used for normal 16-bit address selection and are shifted 4 places to derive 2Ø-bit addresses. Address bits ADR14* through ADR17* (which are listed in Table 2-9), are also used for address selection. When bits specify memory address, ADRØ*, in conjunction with BHEN*, enables even-byte bank on MULTIBUS interface. When bits specify I/O port, only address bits ADRØ* through ADRF* are used.
BHEN*	Byte High Enable: Determines, in conjunction with ADRØ*, byte bank data to be transferred.
BCLK*	Bus Clock: Synchronizes bus contention logic on all bus masters.
CCLK*	Constant Clock: Provides for synchronization of all devices using MULTIBUS interface. Master clock signal.
BPRN*	Bus Priority In: Indicates to particular bus master that no higher priority master is requesting use of bus. BPRN* is synchronized with BCLK*.
BPRO*	Bus Priority Out: In serial priority resolution scheme, indicates to lower priority bus master that neither it (master issuing BPRO* signal) nor higher master is requesting use of bus.
BREQ*	Bus Request: In parallel priority resolution scheme, indicates that issuing bus master requires control of bus for one or more data transfers. BREQ* is synchronized with BCLK*.
BUSY*	Busy: Indicates that bus is in use and prevents other bus masters from gaining control. BUSY* is synchronized with BCLK*.
CBRQ*	Common Bus Request: Indicates that bus master requires control of bus but does not have such control. As soon as control is attained, controlling master raises CBRQ* signal.

Table 2-9. Connector P1 Input/Output Signals (continued)

Signal Mnemonic	Description
LOCK*	MULTIBUS Bus Lock: Prevents off-board requests for on-board dual-port RAM use.
DATØ* -- DATF*	Data Lines: Provide for transmitting or receiving 16 parallel bits of data to or from selected memory address or I/O port. For byte data operations, bits DATØ* through DAT7* constitute even byte and bits DAT8* through DATF* constitute odd byte.
INH1*	Inhibit RAM: Inhibits local RAM cycles.
INH2*	Inhibit ROM: Inhibits local ROM cycles.
INIT*	Initialize: Resets system to known state.
INTØ* -- INT7*	Interrupt Request: Provide for transmitting 8 interrupt requests to assigned interrupt handlers.
INTA*	Interrupt Acknowledge: Not used.
IORC*	I/O Read Command: Indicates that address of I/O port is on MULTIBUS address lines and that port output is to be placed on MULTIBUS data lines.
IOWC*	I/O Write Command: Indicates that address of I/O port is on MULTIBUS address lines and that information on MULTIBUS data lines is to be accepted by addressed port.
MRDC*	Memory Read Command: Indicates that memory address is on MULTIBUS address lines and that contents of address are to be placed on MULTIBUS data lines.
MWTC*	Memory Write Command: Indicates that memory address is on MULTIBUS address lines and that information on MULTIBUS data lines are to be written into address.
XACK*	Transfer Acknowledge: Indicates that specified read or write operation has been completed at memory address or I/O port (that is, data have been placed on or accepted from MULTIBUS data lines).

Table 2-1Ø. Connector P2 Input/Output Signals

Signal Mnemonic	Description
ADR14* -- ADR17*	Address Bits: Specify high-order four bits of memory address to be accessed. (See Table 2-8 for bits ADRØ* -- ADR13*.) ADR14* through ADR17* are used in conjunction with shifted address bits ADRØ* through ADR13* to derive 24-bit addresses (for 16-Mbyte MULTIBUS memory), and are transferred in separate CPU operation.

2.5.2 iSBX™ MULTIMODULE™ INTERFACE

Connectors J3 and J4 on the iSBC 215G board are designed to interface with Intel iSBX I/O controllers or other I/O modules designed to meet the Intel iSBX Bus Specifications. A detailed description of the iSBX bus is given in the MULTIBUS Handbook.

Note that the iSBC 215G board does not comply fully with the iSBX Specification in regard to signals DREQ (DMA Request, pin 34), MWAIT (Expansion Module Wait, pin 16), and EXTR (External Terminate, pin 26). According to the specification, these signals must be uniquely identifiable by the base board for each channel. The iSBC 215G board logically OR's these signals, which thus may be active for only one channel at any time.

The Intel iSBX 218A Flexible Diskette Controller Board connects to the J4 connector and provides an interface between the iSBC 215G board and as many as four 5 1/4- or 8-inch double-density flexible disk drives. The iSBX 218A board interfaces directly with the iSBC 215G board software as described in Chapter 3.

The Intel iSBX 217B/C Magnetic Cartridge Tape Interface Board connects to the J3 connector and provides an interface between the iSBC 215G board and as many as four industry-standard QIC-Ø2 type 1/4-inch magnetic cartridge-tape drives. The iSBX 217B/C board interfaces directly with the iSBC 215G board software as described in Chapter 3.

I/O modules that interface the iSBC 215G board with other storage devices such as bubble memories can also be designed and connected to J3 and/or J4. The device select function of the iSBC 215G board software allows the board to be interfaced with as many as 256 different devices through both iSBX connectors J3 and J4.

The schematic diagram mnemonics for the signal and control lines (from the iSBC 215G board) that are connected to iSBX connectors J3 and J4 often differ from the respective line mnemonic from the iSBX bus specifications. Table 2-11 lists both iSBX bus and iSBC 215G board mnemonics for each signal in the iSBX bus that the board supports. Note that DMA acknowledge pin 32 is not connected on the iSBC 215G board.

Table 2-11. iSBX™ Bus Mnemonics/iSBC® 215G Board Mnemonics

Pin Number	iSBX™ Bus Mnemonic	iSBC® 215G Board Mnemonic J3	J4
1	+12 V	+12 V	+12 V
2	-12 V	-12 V	-12V
3	GND	GND	GND
4	+5 V	+5 V	+5 V
5	RESET	PWR RST	RST
6	MCLK	CCLK	CCLK
7	MA2	IADR-2	IADR-2
8	MPST*	MØPST*	M1PST*
9	MA1	IADR-1	IADR-1
1Ø	Reserved	Reserved	Reserved
11	MAØ	IADR-Ø	IADR-Ø
12	MINTR1	INTR1Ø	INTR11
13	IOWRT*	I-AIOWC*	I-AIOWC*
14	MINTRØ	INTRØØ	INTRØ1
15	IORD*	I-IORC*	I-IORC*
16	MWAIT*	MWAITØ*	MWAIT1*
17	GND	GND	GND
18	+5 V	+5 V	+5 V
19	MD 7	IDAT-7	IDAT-7
2Ø	MCS1*	CSMMIO1*	CSMMIO3*
21	MD6	IDAT-6	IDAT-6
22	MCSØ*	CSMMIOØ*	CSMMIO2*
23	MD5	IDAT-5	IDAT-5
24	Reserved	Reserved	Reserved
25	MD4	IDAT-4	IDAT-4
26	TDMA	EXTRØ	EXTR1
27	MD3	IDAT-3	IDAT-3
28	OPT1	OP1Ø	OP11
29	MD2	IDAT-2	IDAT-2
3Ø	OPTØ	OPØØ	OPØ1
31	MD1	IDAT-1	IDAT-1
32	MDACK*	N/C	N/C
33	MDØ	IDAT-Ø	IDAT-Ø
34	MDRQT	DREQØ	DREQ1
35	GND	GND	GND
36	+5 V	+5 V	+5 V
37	MDE	IDAT-E	IDAT-E
38	MDF	IDAT-F	IDAT-F
39	MDC	IDAT-C	IDAT-C
4Ø	MDD	IDAT-D	IDAT-D
41	MDA	IDAT-A	IDAT-A
42	MDB	IDAT-B	IDAT-B
43	MD8	IDAT-8	IDAT-8
44	MD9	IDAT-9	IDAT-9

2.5.3 CABLING REQUIREMENTS

Interface cables between the iSBC 215G board and the disk drives must be fabricated according to the type of drive being used and the number of drives. Tables 2-12, 2-13, and 2-14 and Figures 2-1 through 2-6 show the signal mnemonics and connector pin assignments for the board and for each type of drive. A 5Ø-pin mass-terminated socket connector (3M 3425/6Ø5Ø or equivalent) is recommended for mating with J1 or J5 of the iSBC 215G board. A 4Ø-pin connector (3M 3417-6Ø4Ø or equivalent) is recommended for mating with J2.

The mass-terminated sockets are easily attached to flat ribbon cable using the jig supplied by the connector manufacturer. The control cables that connect to J1 and J5 require a 5Ø-conductor ribbon cable; the read/write cable that connects to J2 requires one or two 2Ø-conductor ribbon cables, depending on the drive configuration. Total length for either the control cable or the read/write cable must not exceed 1Ø feet. See the respective service manual for the type of connectors required for the cable end that connects to the drives.

Interconnecting cables are shown in Figures 2-7 through 2-12. Most of these require a number of wire cross-overs (scrambling) between the iSBC 215G board connectors and the drives. It is suggested that the scrambling be done at the drive interface connector. Scrambling is not required for the ANSI configuration (Figure 2-14).

NOTE

> The cabling and drive interconnecting information given in this paragraph, and in Figures 2-3 through 2-14, reflect the specifications at the time this manual was printed. Before proceeding with cable construction, check the drive hardware reference manual for current pin assignments and interface requirements.

Table 2-12. Drive Interface Pin-Out Data

Pin Number			Mnemonic
J5	J1	J2	
J5-1	J1-1	-	No connection
J5-2	J1-2	-	BUS-Ø* (HSØ*)
J5-3	J1-3	-	BUS-1* (HS1*)
J5-4	J1-4	-	BUS-2* (HS2*)
J5-5	J1-5	-	BUS-3* (HS3*)
J5-6	J1-6	-	BUS-4*
J5-7	J1-7	-	BUS-5*
J5-8	J1-8	-	BUS-6*
J5-9	J1-9	-	BUS-7*
J5-1Ø	J1-1Ø	J2-1	GND
J5-11	J1-11	-	No connection
J5-12	J1-12	J2-7	GND
J5-13	J1-35	-	SELECTOUT*
J5-14	J1-14	-	GND
J5-15	J1-13	-	CMND*
J5-16	J1-16	-	GND
J5-17	J1-36	-	PARA*
J5-18	J1-18	-	GND
J5-19	J1-39	-	STEP-DIR*
J5-2Ø	J1-2Ø	J2-13	GND
J5-21	J1-43	-	USØ*
J5-22	J1-22	J2-15	GND
J5-23	J1-34	-	ADMKEN*
J5-24	J1-24	J2-17	GND
J5-25	J1-23	-	RD GATE*
J5-26	J1-26	J2-19	GND
J5-27	J1-25	-	WR GATE*
J5-28	J1-28	J2-25	GND
J5-29	J1-42	-	BUS ACK*
J5-3Ø	J1-3Ø	J2-31	GND
J5-31	J1-29	-	INDEX*
J5-32	J1-32	J2-32	GND
J5-33	J1-31	-	SECTOR*
J5-34	-	-	GND
J5-35	J1-15	-	ATTN*
J5-36	-	-	GND
J5-37	J1-33	-	BUSY*
J5-38	J1-38	J2-33	GND
J5-39	-	J2-2	RDØ +
J5-4Ø	-	J2-3	RDØ -
J5-41	J1-46	J2-35	GND
J5-42	-	J2-6	RDCLØ +
J5-43	-	J2-5	RDCLØ -
J5-44	J1-44	J2-37	GND
J5-45	-	J2-12	WRCLØ -
J5-46	-	J2-11	WRCLØ +
J5-47	J1-48	-	GND
J5-48	-	J2-9	WRØ +

Table 2-12. Drive Interface Pin-Out Data (continued)

Pin Number			Mnemonic
J5	J1	J2	
J5-49	-	J2-8	WRØ -
J5-5Ø	J1-5Ø	-	No connection
-	-	J2-2Ø	RD1 +
-	-	J2-21	RD1 -
-	-	J2-23	RDCL1 +
-	-	J2-24	RDCL1 -
-	-	J2-27	WR1 -
-	-	J2-26	WR1 +
-	-	J2-3Ø	WRCL1 -
-	-	J2-29	WRCL1 +
-	J1-17	-	FAULT*
-	J1-21	-	RDY*
-	J1-37	-	SKCOM*
-	-	J2-18	SKCOMØ*
-	-	J2-36	SKCOM1*
-	-	J2-16	SECTØ*
-	-	J2-34	SECT1*
-	J1-4Ø	J2-14	BA1*
-	J1-41	-	BAØ*
-	J1-27	-	BACK*
-	J1-19	-	SAFE*
-	J1-49	-	US3*
-	J1-45	-	US1*
-	J1-47	-	US2*
-	-	J2-4	No connection
-	-	J2-1Ø	No connection
-	-	J2-22	No connection
-	-	J2-26	No connection
-	-	J2-27	No connection
-	-	J2-28	No connection
-	-	J2-29	No connection
-	-	J2-3Ø	No connection
-	-	J2-38	No connection
-	-	J2-39	No connection
-	-	J2-4Ø	No connection

Table 2-13. Control Cable Functions (J1 Complete/J5 Partial)

Signal Mnemonic	Function	Description
Device Select		
USØ*--US3*	Unit Select	Four lines; each selects one of four disk drives.
Head Select		
HSØ*--HS3* (Same as BUS Ø* through BUS 3* below)	Head Select	Four lines; in iSBC 215A/B board, select one of sixteen heads in selected drive. In iSBC 215G board, can be put to various uses; for example: HSØ -- Not used HS1 -- SAFE* HS2 -- SELECTOUT* HS3 -- PARA*
General Purpose Data Bus (Priam and ANSI)		
BUSØ*--BUS7*	Data Bus	Eight-bit, bi-directional data bus; transmits command and status information between iSBC 215G board and drives (includes head and cylinder data).
Command Data		
WRGATE*	Write Select	Enables write circuitry in drive, permitting write data sent to drive through read/write cable to be written on selected disk surface. Used with ADMKEN* line to write address mark on soft-sectored disk.
RDGATE*	Read Select	Enables read circuitry in drive, permitting data to be read from selected sector of disk. Used with ADMKEN* line to read address mark from soft-sectored disk.
DIR*	Direction	Level controls direction of head movement (low for "in", high for "out") when stepping head positioner.
STEP*	Step Head	Initiates head movement in selected direction.
CMND*	Command Data	Indicates that command data are present. Used in bus cycle handshaking.

Table 2-13. Control Cable Functions (J1 Complete/J5 Partial) (continued)

Signal Mnemonic	Function	Description
PARA*	Parameter Data	Indicates that parameter data are present. Used in bus cycle handshaking.
DRIVE REQ* (ATTN* for ANSI only)	Status Data	Indicates that status data are present. Used in bus cycle handshaking.
BUS ACK	Bus Acknowledge	Acknowledges bus cycle. Used in bus cycle handshaking with commands, parameters, and status.
BACK*	Bus Acknowledge	Not used.
ADMKEN*	Address Mark Enable	Enables writing or detecting address marks (beginning of sectors) when used in conjunction with WRGATE* and RDGATE*, respectively. (See SECTOR*.)
SELECTOUT*	Select Unit	Selects and strobes selected drive.
BAØ*/BA1*	Bus Address	Two binary coded lines; specify source or destination register in selected drive for bus data.
SAFE*	Power Safe	Indicates that board power is at safe level.
	Status Data	
INDEX*	Index	Indicates start of each disk revolution on selected disk drive.
SECTOR*	Start of Sector	Indicates start of sector (address mark for soft-sectored disks, sector pulse for hard-sectored disks).
FAULT*	Fault Condition	Indicates that unsafe condition has been detected in selected drive, making read/write operation reliability questionable. Ordinarily, drive logic disables read, write, and positioning circuitry until rezero operation, fault clear operation, or operator intervention.

Table 2-13. Control Cable Functions (J1 Complete/J5 Partial) (continued)

Signal Mnemonic	Function	Description
SKCOM*	Seek Complete	Indicates that selected drive has successfully completed initial head load, seek operation, or rezero operation within specified time limit.
RDY*	Drive Ready	Indicates that drive is powered up and ready to receive or transmit data.
BUSY*	Track Zero/ Busy	Indicates that heads of selected drive have been positioned to cylinder (track) zero or that command is in progress.
BUS ACK*	Bus Acknowledge	Indicates that drive acknowledges parameter request or command.
CHNL ATTN*	Channel Attention	Indicates to IOP that select line should be checked to determine either board master/slave status (following reset), or channel selection.

Table 2-14. Read/Write Cable Functions (J2 Complete/J5 Partial)

Signal Mnemonic	Function	Description
WRØ/WR1 (+ and -)	Write Data	Line pairs; transmitted serial NRZ data (converted from TTL levels to differenttial signals) for recording on disk surface. Write clock synchronizes data transfer.
WRCLØ/WRCL1 (+ and -)	Write Clock	Line pairs; transmitted clock signal for synchronizing write data transmission. Write clock is derived from read clock received from selected drive. Being obtained from rotating disk, read clock signal reflects speed variations and thus ensures proper bit transmission rate when writing as well as reading.
RDØ/RD1 (+ and -)	Read Data	Line pairs; transmitted serial NRZ data from disk drive to iSBC 215G board (to be converted from differential signals to TTL levels) for transmission to host memory. Read clock synchronizes read data transfers.
RDCLØ/RDCL1 (+ and -)	Read Clock	Line pairs; transmitted clock signal used to synchronize read data transmission and as timing signal for board/disk interface circuitry. Read clock is derived from rotating disk.
SECTØ*/ SECT1*	Start of Sector	See SECTOR* in Table 2-12. Binary-coded signal outputs; one from each unit.
SKCOMØ*/ SKCOM1*	Seek Complete	See SKCOM* in Table 2-12. Binary-coded signal outputs; one from each unit Ø and 1.
RDWCUR*	Reduced Write Current	Controls write electronics for inner tracks with higher bit densities.

8″ Shugart/Quantum Drive 0 Meeting Connector 50-Pin ①	Signal	iSBC* 215G Board Mating Connector J1 50-Pin ②
1		
2		
3	Ground (GND)	10
4	− Head Select² (− HS2/)	4
5		
6		
7		
8	− SEEK COMPLETE (SKCOM/)	37
9	Ground (GND)	38
10		
11		
12		
13	Ground (GND)	12
14	− HEAD SELECT 2⁰(H − HS0/)	2
15		
16		
17	Ground (GND)	14
18	− HEAD SELECT 2¹ (− HS1/)	3
19	Ground (GND)	28
20	− INDEX (INDEX/)	29
21	Ground (GND)	20
22	− READY (RDY)	21
23		
24		
25		
26	− DRIVE SELECT 1 (US0/)	43
27		
28	− DRIVE SELECT 2 (US1/)	45
29	Ground (GRD)	44
30	− DRIVE SELECT 3 (US2/)	47
31	Ground (GND)	46
32	− DRIVE SELECT 4 (US3/)	49
33	Ground (GND)	48
34	− DIRECTION IN (DIR/)	41
35		
36	− STEP (STEP/)	39
37		
38		
39		
40	− WRITE GATE (WRGATE/)	25
41	Ground (GND)	26
42	− TRACK 000 (BUSY/)	33
43	Ground (GND)	32
44	− WRITE FAULT (FAULT/)	17
45	Ground (GND)	18

46 through 50 — no connections

Shugart Data Separator Mating Connector J5 20-Pin ④	Signal	iSBC* 215G Board Connectors* J1 Mating Connectors 50-Pin ②	J2 40-Pin ③
1	− READ GATE (RDGATE)	23	
2	Ground (GND)	22	
3	− AMF (SECTOR/)*	31	
4	Ground (GND)	24	
5	− WRAM (ADMKEN/)	34	
6	Ground (GND)	30	
7	− RWC (RDWRCUR/)		14
8	Ground (GRD)		15
9	+ NRZ WRITE DATA (WR +)		26
10	− NRZ WRITE DATA (WR −)		27
11	Ground (GRD)		10
12	+ WRITE CLOCK (WRCL +)		29
13	− WRITE CLOCK (WRCL −)		30
14	Ground (GND)		13
15	+ READ CLOCK (RDCLO +)		6
16	− READ CLOCK (RDCLO −)		5
17	Ground (GND)		7
18	+ NRZ READ DATA (RD0 +)		2
19	− NRZ READ DATA (RD0 −)		3
20	Ground (GRD)		4

Shugard Data Separator Mating Connector 20-Pin ④	Signal	8″ Shugart/Quantum Drive 0 Mating Connector 20-Pin ④
1	− DRIVE SELECTED/	1
2	Ground (GND)	2
3	SPARE	3
4	Ground (GND)	4
5	SPARE	5
6	Ground (GND)	6
7	SPARE	7
8	Ground (GND)	8
9	+ TIMING CLK	9
10	− TIMING CLK	10
11	Ground (GND)	11
12	Ground (GND)	12
13	+ MFM Write Data	13
14	− MFM Write Data	14
15	Ground (GND)	15
16	Ground (GND)	16
17	+ MFM READ DATA	17
18	− MFM READ DATA	18
19	Ground (GND)	19
20	Ground (GND)	20

iSBC 215G Board (signal name) in parentheses

x-753A

Figure 2-1. 8-Inch Shugart/Quantum Drive Interconnection Listing

Fujitsu/Memorex/14" Shugart Drive Mating Connector 50-Pin ①	Signal	iSBC* 215G Board Connector J1 50-Pin ②
1		1
2	– Head Select 0 (– HS0/)	2
3	Ground (GND)	10
4	– Head Select 1 (– HS1/)	3
5	Ground (GND)	12
6	– Head Select 2 (– HS2/)	4
7	Ground (GND)	14
8		
9		
10	– Index (INDEX)	29
11	Ground (GND)	30
12	– Drive Ready (RDY)	21
13	Ground (GND)	22
14	– Sector/Byte Clock (SECTOR/)	31
15	Ground (GND)	32
16	– Drive Select 1 (US0/)	43
17	Ground (GND)	44
18	– Drive Select 2 (US1/)	45
19	Ground (GND)	46
20	– Drive Select 3 (US2/)	47
21	Ground (GND)	48
22	– Drive Select 4 (US3/)	49
23		
24	– Direction (DIR/)	41
25	Ground (GND)	38
26	– Step (STEP/)	39
27		
28	– Fault Clear (FLT CLR/)	35
29		
30	– Write Gate (WRGATE/)	25
31	Ground (GND)	26
32	– Track 0 (BUSY/)	33
33	Ground (GND)	16
34	– Write Fault (FAULT/)	17
35	Ground (GND)	20
36	– Read Gate (RDGATE/)	23
37	Ground (GND)	24
38	Ground (GND)	18

iSBC 215G Board (signal name) in parentheses.
**When interface with a 14" Shugart drive pins 15 and 16 on both radial connectors should be swapped: pin 15. + PLO Clock (RDCLO –): pin 16. – PLO Clock (RDCLO +).

Fujitsu/Memorex/14" Shugart Drive Mating Connector Drive 0 20-Pin ④	Signal	iSBC* 215G Board Connector J2 40-Pin ③
1		
2		
3		
4		
5		
6		
7	Seek Complete (SKCOM0/)	18
8	Ground (GND)	19
9	+ Write Data (WRO +)	9
10	– Write Data (WRO –)	8
11	Ground (GND)	4
12	+ Write Clock (WRCLO +)	11
13	– Write Clock (WRCLO –)	12
14	Ground (GND)	7
15	+ PLO Clock (RDCLO +)	6
16	– PLO Clock (RDCLO –)	5
17	Ground (GND)	10
18	+ Read Data (RD0 +)	2
19	– Read Data (RD0 –)	3
20	Ground (GND)	13

Memorex/14" Shugart Drive Mating Connector 20-Pin ④	Signal	iSBC* 215G Board Connector J2 40-Pin ③
1		
2		
3		
4		
5		
6		
7	– Seek Complete (SKCOM1)	36
8	Ground (GND)	25
9	+ Write Data (WR1 +)	26
10	– Write Data (WR1 –)	27
11	Ground (GND)	28
12	+ Write Clock (WRCL1 +)	29
13	– Write Clock (WRCL1 –)	30
14	Ground (GND)	22
15	+ PLO Clock (RDCL1 +)	23
16	– PLO Clock (RDCL1 –)	24
17	Ground (GND)	31
18	+ Read Data (RD1 +)	20
19	– Read Data (RD1 –)	21
20	Ground (GND)	32

x-752A

Figure 2-2. Fujitsu 2300/Memorex/14-Inch Shugart Drive Interconnection Listing

Priam Drive Mating Connector — iSBC* 215G Board Connector

Priam 50-Pin ①	Signal	J1 50-Pin ②	J2 40-Pin ③
1		1	
2	+DBUS 0 (BUS 0/)	2	
3	+DBUS 1 (BUS 1/)	3	
4	+DBUS 2 (BUS 2/)	4	
5	+DBUS 3 (BUS 3/)	5	
6	+DBUS 4 (BUS 4/)	6	
7	+DBUS 5 (BUS 5/)	7	
8	+DBUS 6 (BUS 6/)	8	
9	+DBUS 7 (BUS 7/)	9	
10	Ground (GND)	10	
11	−READ GATE (RDGATE/)	23	
12	Ground (GND)	12	
13	Reset	33	
14	Ground (GND)	14	
15	−WRITE GATE (WR GATE/)	25	
16	Ground (GND)	16	
17	−RD (Comnd/)	13	
18	−WR (Step/)	39	
19	+AD1 (BA1/)	40	
20	+AD0 (BA0/)	41	
21	Ground (GND)	20	
22	−DRIVE SELECT 1 (US0/)	43	
23	−DRIVE SELECT 2 (US1/)	45	
24	−DRIVE SELECT 3 (US2/)	47	
25	−DRIVE SELECT 4 (US3/)	49	
26	Ground (GND)	26	
27	Ground (GND)	28	
28			
29	−HEAD SELECT 3 PARA/)	36	
30	−HEAD SELECT 2 (Select Out/)	35	
31	−HEAD SELECT 1 (Safe/)	19	
32	Ground (GND)	32	
33	−INDEX (Index/)	29	
34	Ground (GND)	37	
35	−READY (RDY/)	21	
36	Ground (GND)	38	
37	−SECTOR MARK	31	
38	Ground (GND)	44	
39	+WRITE DATA (WR0 +)		9
40	−WRITE DATA (WR0 −)		8
41	Ground (GND)		4
42	+WRITE CLOCK (WRCLO +)		11
43	−WRITE CLOCK (WRCLO −)		12
44	Ground (GND)		7
45	+READ/REFERENCE CLOCK (WRCLO +)		6
46	−READ/REFERENCE CLOCK (WRCLO −)		5
47	Ground (GND)		10
48	+READ DATA (RD0 +)		2
49	−READ DATA (RD0 −)		3
50	Ground (GND)		13

iSBC 215G Board (signal name) in parentheses

x-754A

Figure 2-3. Priam Drive Interconnection Listing

5¼"Drive 0 Mating Connector 34-Pin ①	Signal	iSBC* 215G Board Mating Connector J1 50-Pin ②
1		
2		
3		
4	– Head Select 2^2 (– HS2/)	4
5	Ground (GND)	10
6	Write Gate (WRGATE/)	25
7	Ground (GND)	26
8	– SEEK COMPLETE (SKCOM/)	37
9	Ground (GND)	38
10	Track 000 (Busy/)	33
11	Ground (GND)	32
12	Write Fault (FAULT/)	17
13	Ground (GND)	18
14	– HEAD SELECT 2^0 (– HS0/)	2
15	Ground (GND)	12
16		
17		
18	– HEAD SELECT 2^1 (– HS1/)	3
19	Ground (GND)	14
20	– INDEX (INDEX/)	29
21	Ground (GND)	28
22	– READY (RDY/)	21
23	Ground (GND)	20
24	Step (STEP/)	39
25		
26	– DRIVE SELECT 1 (US0/)	43
27	Ground (GND)	44
28	– DRIVE SELECT 2 (US1/)	45
29	Ground (GND)	46
30	– DRIVE SELECT 3 (US2/)	47
31	Ground (GND)	48
32	– DRIVE SELECT 4 (US3/)	49
33		
34	– DIRECTION IN (DIR/)	41

NOTE: This interconnection listing is for information only. Interface to an ST412 compatible 5¼" hard-disk drive requires one of the iSBC* 215G kits, which provides, in addition to the iSBC* 215G board, the iSBC* 213 data separator board and all required cables.

Data Seperator Mating Connector 20-Pin ④	Signal	Drive 0 Mating Connector 20-Pin ④
1	– DRIVE SELECTED/	1
2	Ground (GND)	2
3	SPARE	3
4	Ground (GND)	4
5	SPARE	5
6	Ground (GND)	6
7	SPARE	7
8	Ground (GND)	8
9	+ TIMING CLK	9
10	Ground (GND)	10
11	Ground (GND)	11
12	Ground (GND)	12
13	+ MFM Write Cock	13
14	– MFM Write Clock	14
15	Ground (GND)	15
16	Ground (GND)	16
17	+ MFM READ DATA	17
18	– MFM READ DATA	18
19	Ground (GND)	19
20	Ground (GND)	20

Data Separator Mating Connector J5 20-Pin ④	Signal	iSBC* 215G Board Connectors* J1 Mating Connectors 50-Pin ②	J2 40-Pin ③
1	– READ GATE (RDGATE/)	23	
2	Ground (GND)	22	
3	– AMF (SECTOR/)	31	
4	Ground (GND)	24	
5	– WRAM (ADMKEN/)	34	
6	Ground (GND)	30	
7	– RWC (RDWRCUR/)		14
8	Ground (GND)		15
9	+ NRZ WRITE DATA (WR0 +)		9
10	– NRZ WRITE DATA (WR0 –)		8
11	Ground (GND)		10
12	+ WRITE CLOCK (WRCL0 +)		11
13	– WRITE CLOCK (WRCL0 –)		12
14	Ground (GND)		13
15	+ READ CLOCK (RDCL0 +)		6
16	– READ CLOCK (RDCL0 –)		5
17	Ground (GND)		7
18	+ NRZ READ DATA (RD0 +)		2
19	– NRZ READ DATA (RD0 –)		3
20	Ground (GND)		4

iSBC 215G Board (signal name) in parentheses.

x-755A

Figure 2-4. 5 1/4-Inch Drive Interconnection Listing

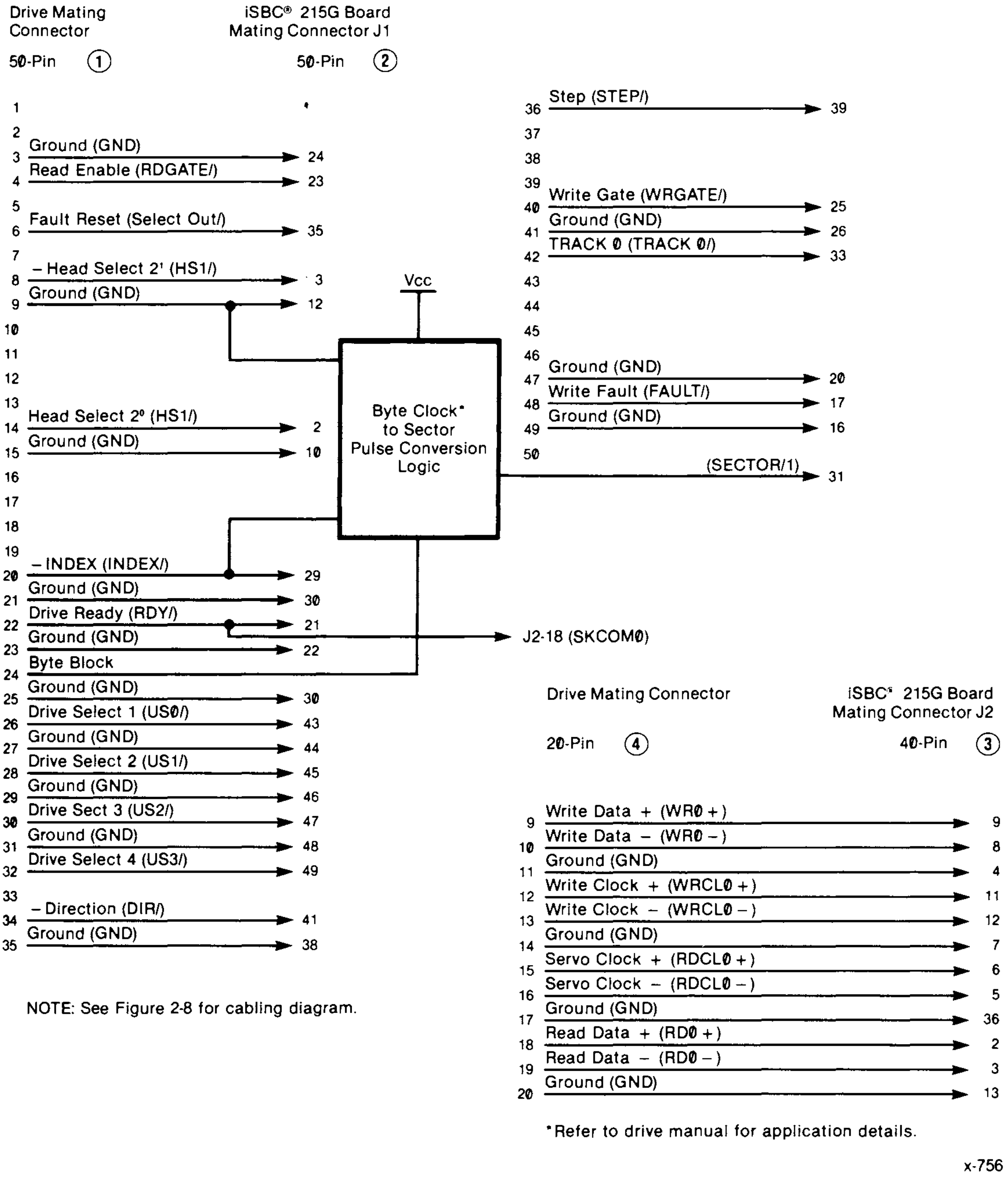


Figure 2-5. CDC Drive Interconnection Listing

ANSI Drive 0 Mating Connector 50-Pin ①	Signal	iSBC® 215G Board Connector J5 50-Pin ②
1		1
2	BUS-0★ (HS 0★)	2
3	BUS-1★ (MS 1★)	3
4	BUS-2★ (HS 2★)	4
5	BUS-3★ (HS 3★)	5
6	BUS-4★	6
7	BUS-5★	7
8	BUS-6★	8
9	BUS-7★	9
10	GND	10
11		11
12	GND	12
13	SELECTOUT★	13
14	GND	14
15	CMND★	15
16	GND	16
17	PARA★	17
18	GND	18
19	STEP-DIR★	19
20	GND	20
21	US0★	21
22	GND	22
23	ADMKEN★	23
24	GND	24
25	RD GATE★	25
26	GND	26
27	WR GATE★	27
28	GND	28
29	BUS ACK★	29
30	GND	30
31	INDEX★	31
32	GND	32
33	SECTOR★	33
34	GND	34
35	ATTN★	35
36	GND	36
37	BUSY★	37
38	GND	38
39	RD0+	39
40	RD0−	40
41	GND	41
42	RDCL0+	42
43	RDSL0−	43
44	GND	44
45	WRCL0−	45
46	WRCL0+	46
47	GND	47
48	WR0+	48
49	WR0−	49
50		50

x-757A

Figure 2-6. ANSI Drive Interconnection Listing

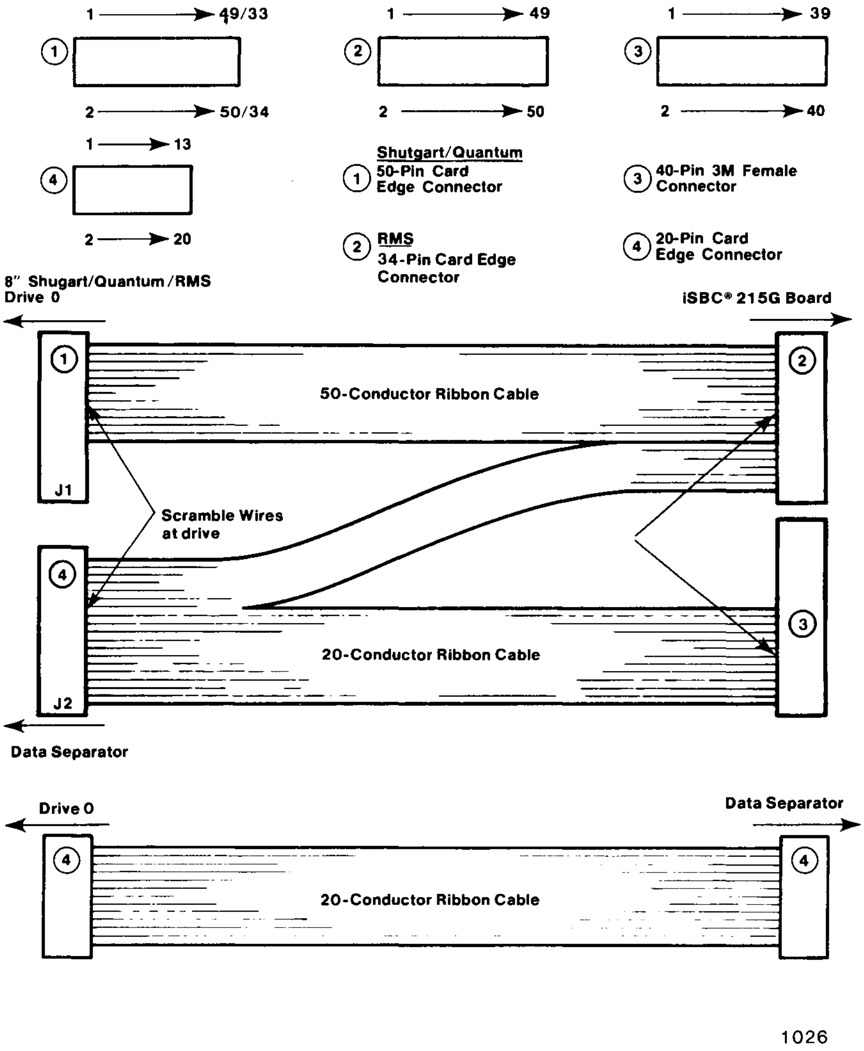


Figure 2-7. 8-Inch Shugart/Quantum Drive Interconnecting Cable

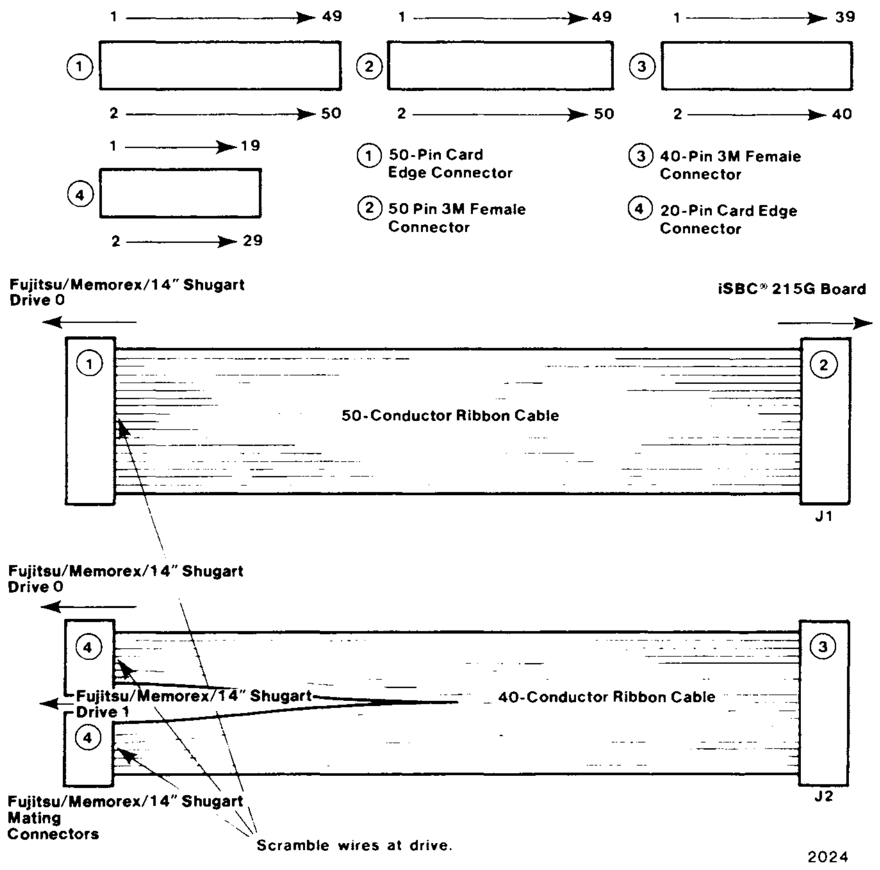


Figure 2-8. Fujitsu 23ØØ/Memorex/14-Inch Shugart Drive Interconnecting Cable

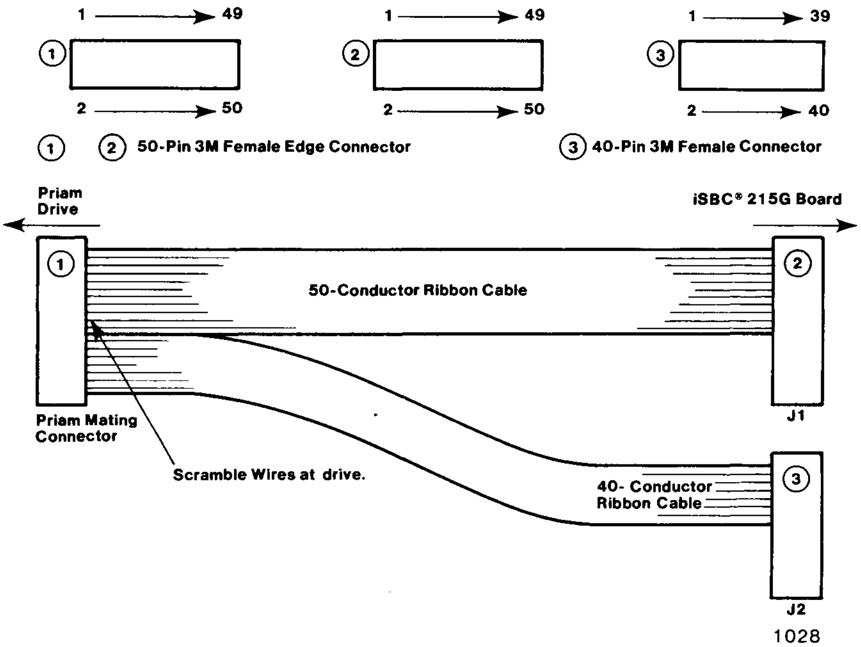


Figure 2-9. Priam Drive Interconnecting Cable

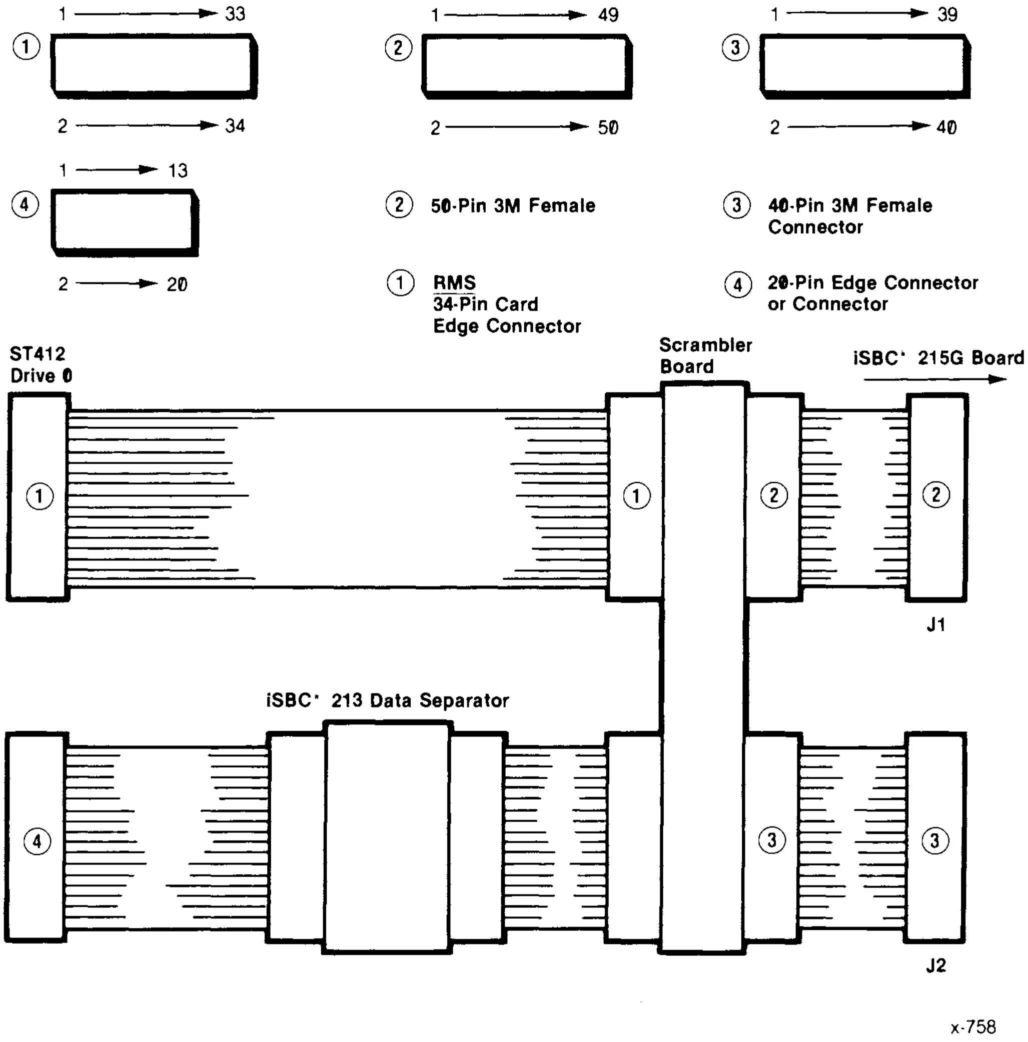


Figure 2-1Ø. 5 1/4-Inch Drive Interconnecting Cable

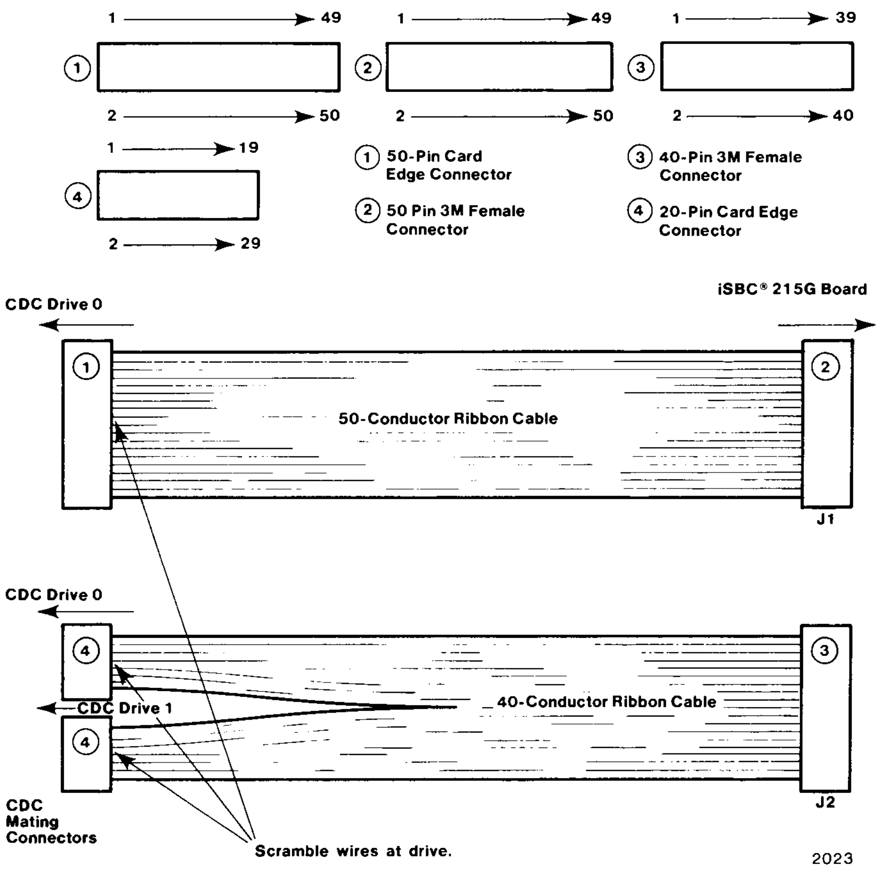


Figure 2-11. CDC Drive Interconnecting Cable

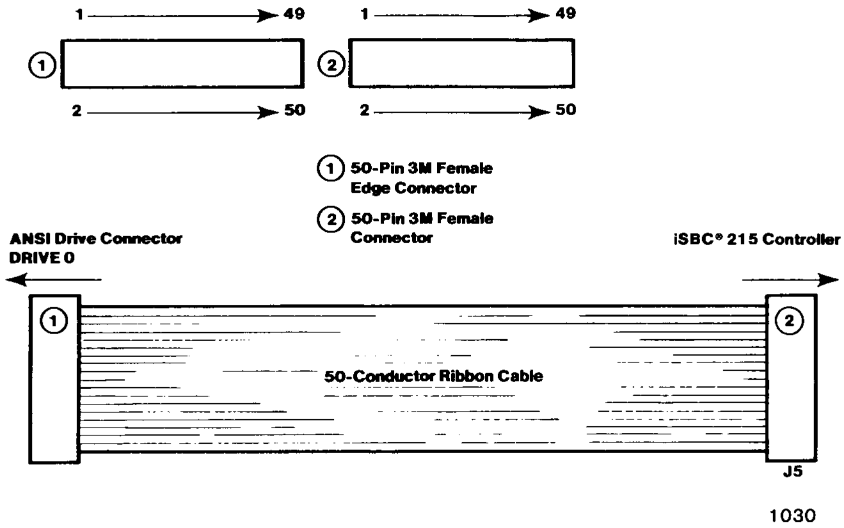


Figure 2-12. ANSI Drive Interconnecting Cable

2.6 BOARD INSTALLATION

The iSBC 215G board can be installed in any MULTIBUS compatible cardcage or chassis. Installation is accomplished as outlined in the following procedure:

1. Remove power from cardcage or chassis.

2. Connect all jumpers for desired configuration (refer to paragraph 2.4).

3. Install iSBX MULTIMODULE board or boards as outlined in appropriate procedure (paragraph 2.6.2 for iSBX 218A board; paragraph 2.6.3 for iSBX 217B/C board).

4. Slide board into desired slot and press firmly to make certain that both connectors are properly seated.

5. Apply power to cardcage or chassis.

2.6.1 DRIVE INSTALLATION

The requirements for connecting the iSBC 215G board to the disk drive vary among drive types. Drives that conform to the ANSI X3T9/1226 interface can simply be plugged into iSBC 215G board center connector J5 (see Figure 2-13). Drives that do not conform to the ANSI interface require different interconnection schemes. Connectors J1 (left-hand connector) and J2 (right-hand connector) are used for non-ANSI interfacing. The iSBC 215G board is pin-compatible with earlier board versions. Connectors J1 and J2 pin functions are consistent with those versions, but pin numbering has been altered (see Figure 2-14). (Earlier versions of the board do not support connector J5.) The following paragraphs describe several non-ANSI interconnection schemes.

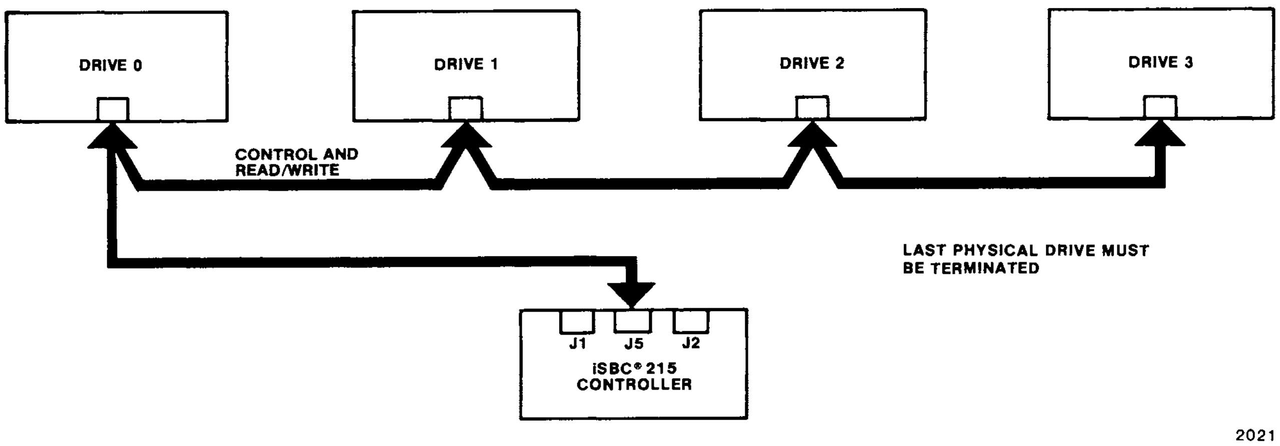


Figure 2-13. ANSI Drive Interface

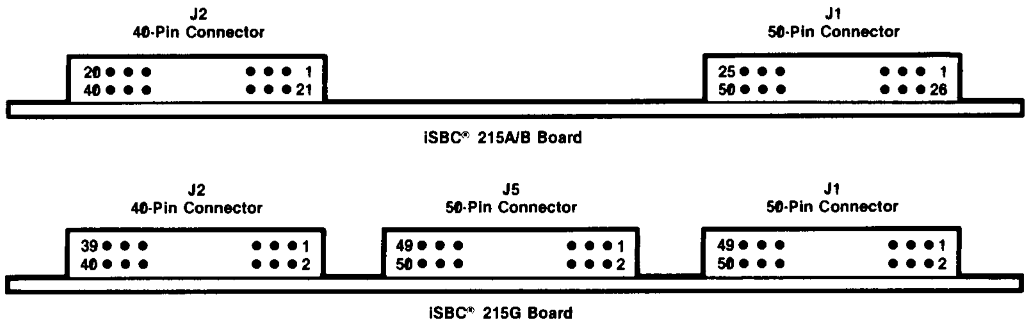


Figure 2-14. Pin Numbering Conventions

2.6.1.1 14/8-Inch Drives With Integral Data Separator

Connecting the iSBC 215G board to drives with integral data separators requires only constructing and attaching the interconnecting cables. Refer to Figure 2-15. It may be helpful also to refer to Figures 2-1 through 2-12.

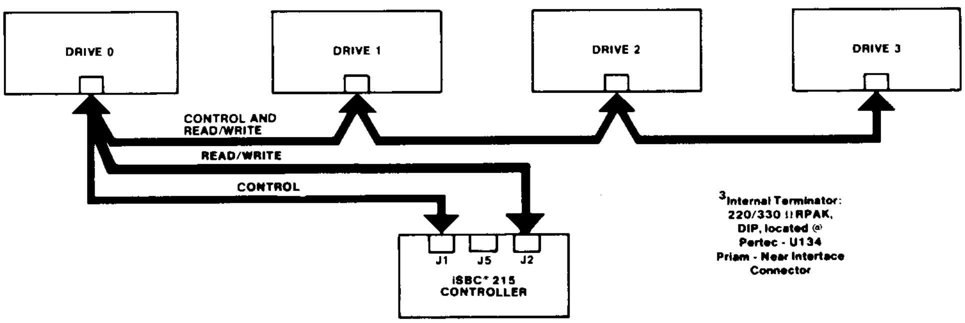


Figure 2-15. Interface for Drive With Integral Data Separator

2.6.1.2 5 1/4-Inch Drives

All 5 1/4-inch drives used with the iSBC 215G board must be ST5Ø6/412 interface compatible (the iSBC 215G board does not provide step-pulse buffering). These drives require the use of the iSBC 215G Kit, which consists of an iSBC 215G board, an iSBC 213 data separator, and a cable set. Refer to Figure 2-16.

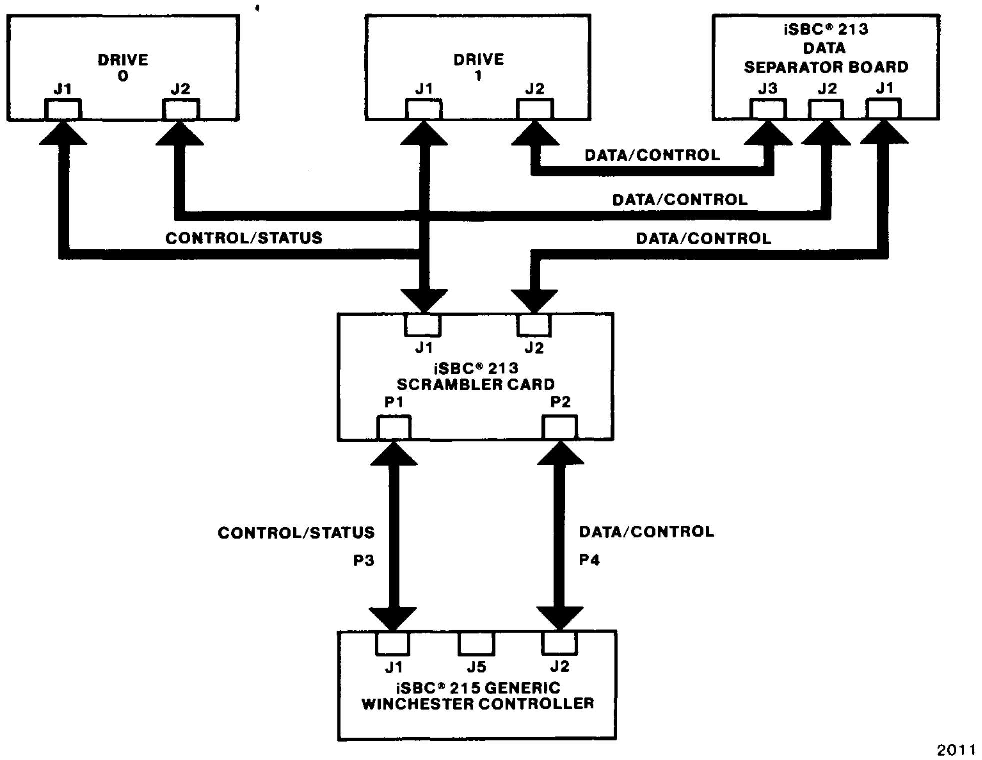


Figure 2-16. Interface for Drives and iSBC® 215G Kit

2.6.1.3 Other Drives Without Data Separator

Drives without data separators, other than ST5Ø6/412 compatible types, require the use of a Shugart, RMS, or equivalent data separator, and require constructing and attaching the interconnecting cables. Refer to Figure 2-17. It may be helpful also to refer to Figures 2-1 through 2-12.

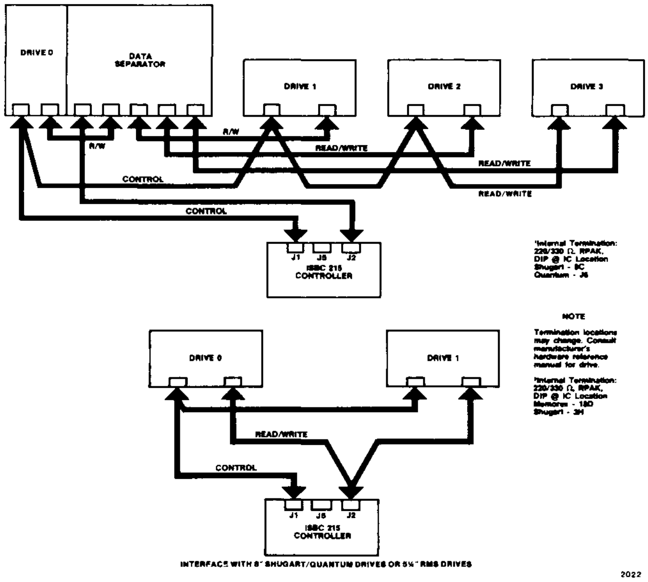


Figure 2-17. Interface for Drives Without Data Separator

2.6.2 iSBX™ 218A BOARD INSTALLATION

The iSBX 218A board connects to J4 on the iSBC 215G board. Before installing the iSBX 218A board, install iSBC 215G board jumper W4-1 -- 2. Also, install jumper W24 as appropriate for DMA conditions (see Table 2-4). A single cable that transmits both control and read/write information is required to connect the iSBX 218A board to the flexible-disk drives as shown in Figure 1-1. Refer to the iSBX 218A Flexible Diskette Controller Board Hardware Reference Manual for further installation details and operating information.

Install the board as follows:

1. Install supplied threaded spacers on solder side of iSBX 218A board (at holes near connectors J1 and J2 and at J2 end of board), securing spacers by inserting and hand-tightening supplied 1/4-inch screws from component side of iSBX 218A board (see Figure 2-18.)

2. Locate pin 1 on iSBC 215G board iSBX connector J4. Similarly, locate pin 1 on iSBX 218A board connector (see Figure 2-18.)

3. Carefully match connectors at pin 1, insert iSBX 218A board connector into iSBC 215G board iSBX connector, and press gently to seat fully. (iSBX board connector should be oriented in same direction as iSBC 215G board I/O connectors.)

4. Insert remaining 1/4-inch screws from solder side of iSBC 215G board into spacers, and tighten all screws.

2.6.3 iSBX™ 217B/C BOARD INSTALLATION

The iSBX 217B/C board connects to J3 on the iSBC 215G board. Before installing the iSBX 217B/C board, install iSBC 215G board jumper W3-1 -- 2. Also, install jumper W24 as appropriate for DMA conditions (see Table 2-4). A single cable that transmits both control and read/write information is required to connect the iSBX 217B/C board to the magnetic cartridge-tape drive or drives. Refer to the iSBX 217B/C Magnetic Cartridge Tape Interface Board Hardware Reference Manual for further installation details and operating instructions.

Install the board as follows:

1. Install supplied threaded spacer on solder side of iSBX 217B/C board (at hole near connector J1), securing spacer by inserting and hand-tightening one of supplied 1/4-inch screws from component side of iSBX 217B/C board (see Figure 2-19.)

2. Locate pin 1 on iSBC 215G board iSBX connector J3. Similarly, locate pin 1 on iSBX 217B/C board connector (see Figure 2-19.)

3. Carefully match connectors at pin 1, insert iSBX 217B/C board connector into iSBC 215G board iSBX connector, and press gently to seat fully. (iSBX board connector should be oriented in same direction as iSBC 215G board I/O connectors.)

4. Insert remaining 1/4-inch screw from solder side of iSBC 215G board into spacer, and tighten both screws.

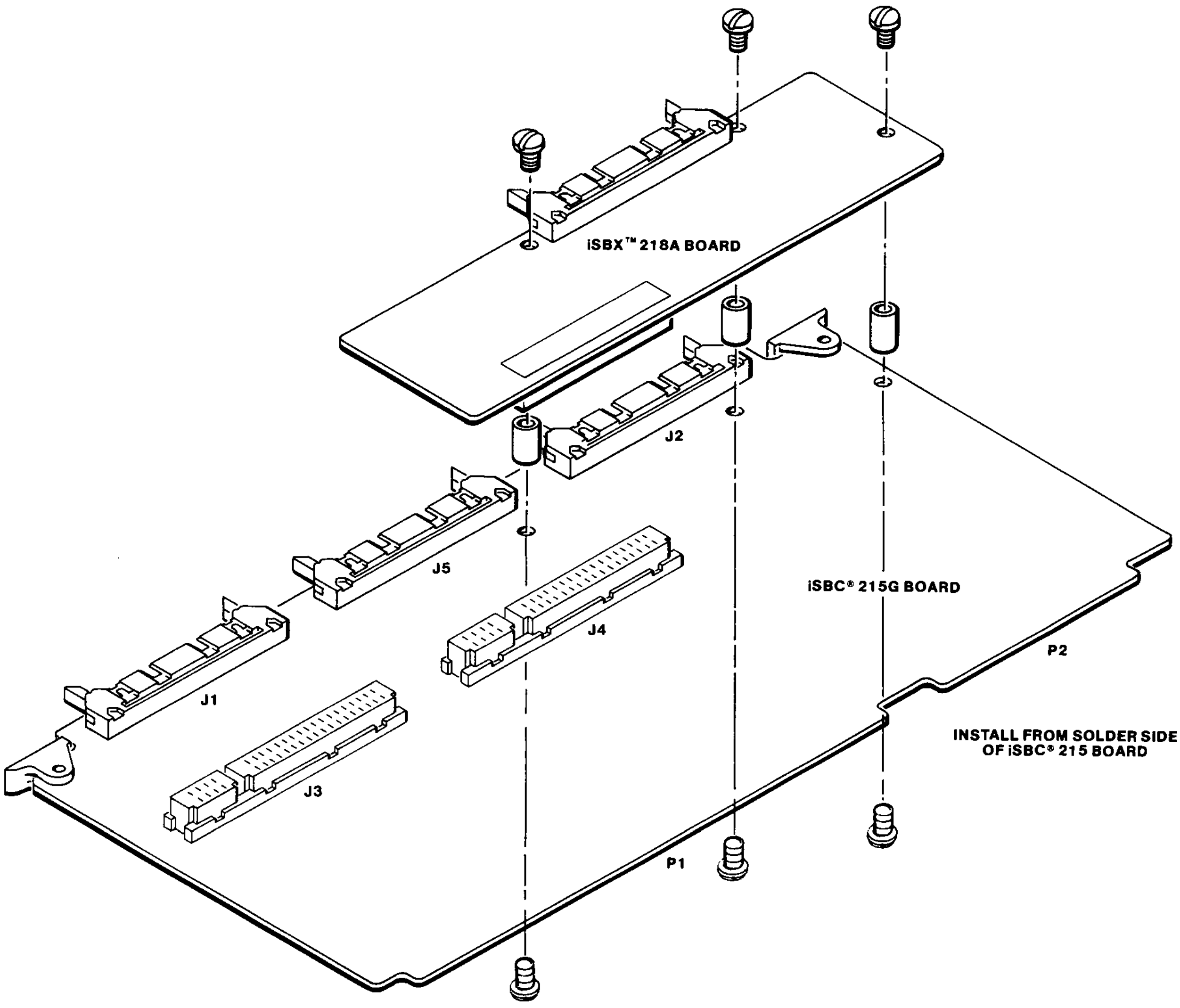


Figure 2-18. iSBX™ 218A Board Installation

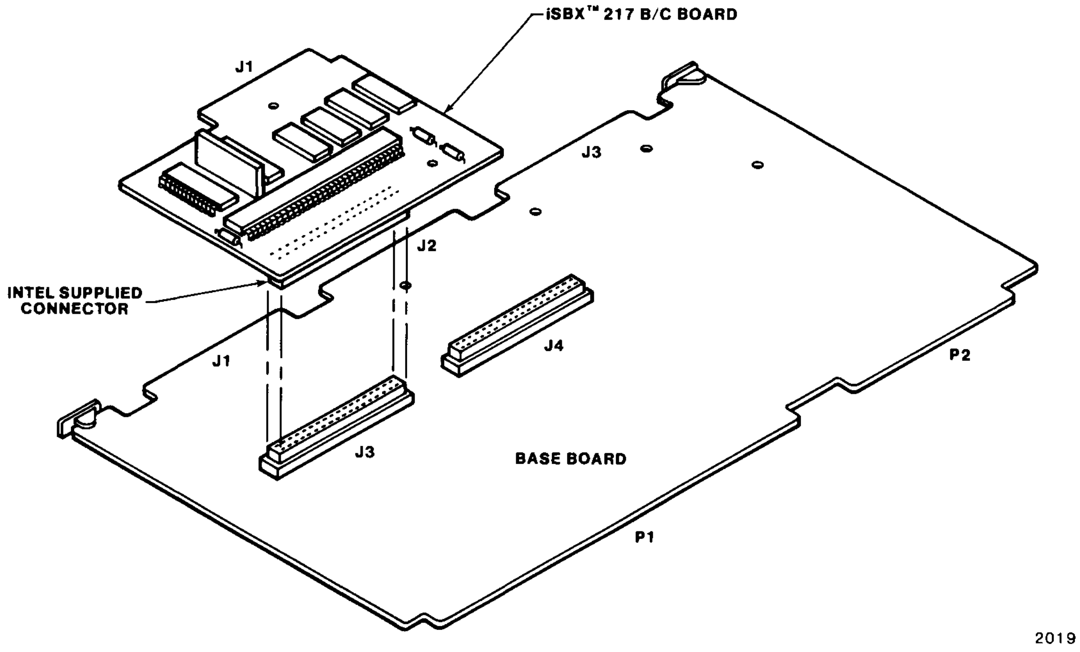


Figure 2-19. iSBX™ 217B/C Board Installation

***

CHAPTER 3
PROGRAMMING

3.1 INTRODUCTION

This chapter describes the programming conventions that must be followed to initiate and monitor the transfer of data between the system memory and a disk (hard or flexible) or tape drive. The firmware installed on the iSBC 215G board includes direct support for hard-disk (Winchester technology) interface, flexible-disk interface via the iSBX 218A Flexible Diskette Controller Board, and the QIC-Ø2 cartridge-tape interface via the iSBX 217B/C Magnetic Cartridge Tape Interface Board.

NOTE

> If power is applied to or removed from the iSBC 215G board while a drive is ready, a spurious disk-write operation could occur. To prevent this, make certain that the disks are stopped whenever iSBC 215G board power is switched on or off.

Included in this section are: 1) descriptions of disk organization, track sectoring format, disk controller communications protocol, interrupt handling, the use of disk control functions; and 2) special procedures for programming I/O transfers to flexible-disk and cartridge-tape drives through the iSBX interface.

3.2 PROGRAMMING OPTIONS

The iSBC 215G board is designed to interface with hard-disk (Winchester technology) drives as specified in Chapters 1 and 2. The board also has two iSBX connectors for communication with other I/O devices through an iSBX I/O controller such as the iSBX 218A Flexible Diskette Controller Board or the iSBX 217B/C Magnetic Cartridge Tape Interface Board.

The iSBC 215G board contains a ROM-resident I/O transfer program designed to control data transfers between the board and hard-disk drives as well as between the board and flexible-disk and cartridge-tape drives connected to the iSBX 218A and iSBX 217B/C controllers. In addition, the iSBC 215G board can also execute programs that the user has written in 8Ø89 I/O Processor (IOP) assembler code to control other I/O devices through the iSBX bus on the board.

3.3 MASS STORAGE PRINCIPLES

Mass storage, for purposes of this manual, is defined as the capacity of a peripheral device or combination of devices for storage of large amounts of data in applications requiring low-cost, high-capacity, random-access storage, but not the very low access times of such integrated-circuit devices as RAM's. The iSBC 215G board can accommodate storage devices of three types: hard-disk (Winchester technology), flexible-disk, and cartridge-tape. While hard-disk devices offer greater capacity per device and somewhat faster access, flexible-disk and cartridge-tape devices offer virtually unlimited capacity through interchangeable storage media. Thus, with the iSBC 215G board, the user can augment the system memory in the most efficient fashion, using any required combination of storage media.

3.3.1 HARD-DISK ORGANIZATION

In this description, a head is assumed to be associated with a single disk surface. Each surface can have as many as 4Ø96 tracks (circular data paths numbered Ø through 4Ø95). The set of tracks on multiple recording surfaces (one track per surface) at a given head position or location is referred to as a "cylinder" (see Figure 3-1). A drive that has 4Ø96 tracks per surface thus has 4Ø96 cylinders.

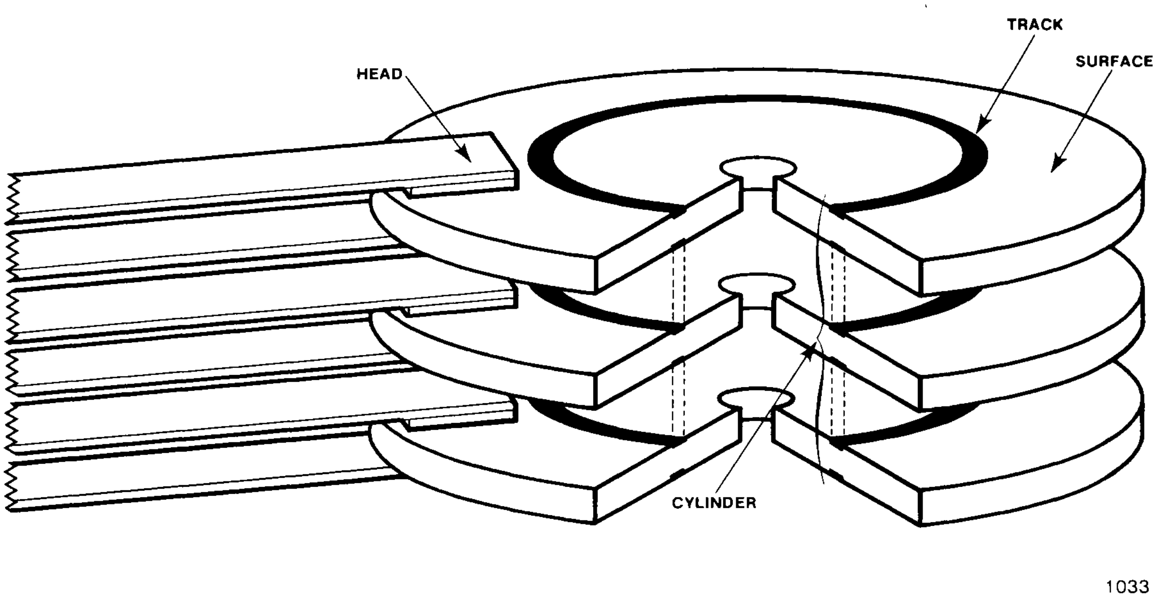


Figure 3-1. Disk Drive Organization and Terminology

Each track is divided into equal-sized sectors. Each of these sectors includes a sector identification block with error checking information. The iSBC 215G board allows the user to select the size of the data block, which then determines the maximum number of sectors permitted per track (as shown in Table 1-1). The iSBC 215G board 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 board organizes this information for 8-inch hard-disk drives. Refer to paragraph 3.5.3 for further information on track formatting.

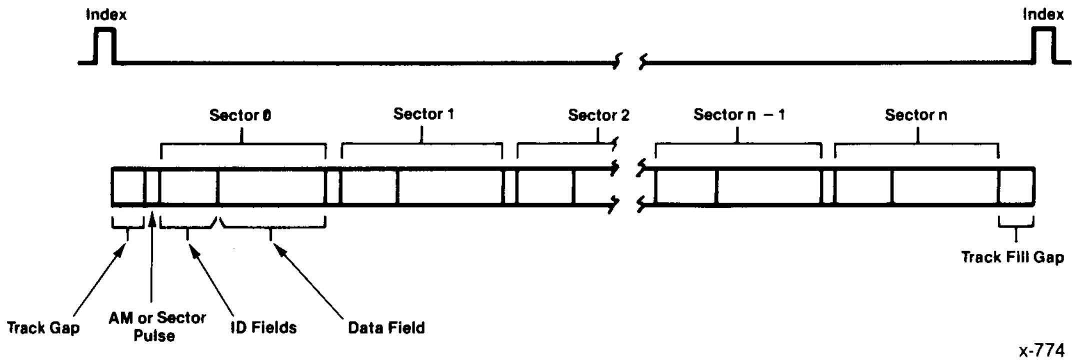


Figure 3-2. Sector Fields

3.3.2 FLEXIBLE-DISK ORGANIZATION

The organization of data on the flexible disk is much the same as that of the hard-disk. The primary difference is that flexible-disk drives use only one disk at a time, which provides one or two recording surfaces. Hard-disk drives, however, can contain multiple disks, thus providing multiple recording surfaces. Refer to the iSBX 218A Flexible Diskette Controller Board hardware reference manual for information on flexible-disk track formatting.

3.3.3 CARTRIDGE-TAPE ORGANIZATION

Data stored on tape differs from that on disk in that files are not limited in length as on disk. A file is simply a string of data that ends with an end-of-file character. Refer to the iSBX 217B/C Magnetic Cartridge Tape Interface Board hardware reference manual for information on cartridge-tape file formatting.

3.4 HOST/BOARD COMMUNICATIONS

The iSBC 215G board is a full DMA device that is capable of operating as a bus master for transferring information to and from system memory. However, it cannot operate as the system master (host), and depends upon the system master to provide operation programming. The board responds to any host CPU that provides the necessary operation programming. Thus, the iSBC 215G board can be used in multiprocessor systems if the necessary software interlocks are used to assure that multiple, concurrent mass storage accesses are not attempted. All mass storage operations are initiated by the output from the host CPU of a command byte to the wake-up port assigned to the iSBC 215G board. Once the operation is initiated by the host CPU, all subsequent communication between the host CPU and the board, until the operation is complete, take place using the I/O communications blocks established in memory by the host CPU prior to initiating the mass-storage operation.

The I/O communication blocks structure for the board, exclusive of any data buffer, consists of 68 bytes of memory that are arranged into 4 blocks. The format of each of the 4 blocks is specifically defined. However, the blocks can be arranged in any order or in any location within a 1-Mbyte page of memory (dedicated memory locations excluded). Each of the blocks has a defined format and the memory bytes that make up each must be contiguous. Each of the blocks also has a defined function related to the overall operation of the iSBC 215G board.

The most important advantage of such a communications block structure is flexibility. Though some of the blocks should be limited in use to only one such block in memory, the system may contain multiple copies of blocks used directly to specify operations. Thus, by merely changing a few pointers, software can specify a different storage operation without structuring an all-new I/O block.

3.4.1 WAKE-UP I/O PORT

The wake-up I/O port is the I/O address to which the iSBC 215G board responds. This single I/O address is user selectable through jumpers on the board and may be either 8 or 16 bits, as applicable to the host CPU and the application. The command signal that controls the number of bits in the address to which the board will respond is also jumper selectable.

NOTE

> The jumpers that select the I/O port address (shifted to the left four places) also select the first address in the wake-up block. Thus wake-up I/O port address 1ØØH also specifies wake-up block address 1ØØØH.

To invoke iSBC 215G board activity, the host CPU transmits a wake-up command byte to the board through the wake-up I/O port. Three wake-up commands are allowed, as shown in Table 3-1. Note that only the two least significant bits of the command are used to determine which of the three hardware functions to implement. Note that only MULTIBUS I/O write operations are recognized.

Refer to paragraph 2.4.2 for detailed information on the 8- and 16-bit I/O port addresses. Also, refer to paragraph 2.4.1 for detailed information on wake-up addresses.

Table 3-1. I/O Control Commands

Command	Function
ØØH	Clear Interrupt -- iSBC 215G board to host CPU interrupt is reset; board reset is cleared.
Ø1H	Start Operation -- Instructs board to start operation defined by I/O parameter block elements.
Ø2H	Reset Board -- Performs hardware reset of board. Clear interrupt (ØØH) must be initiated following this command. (Each time board is reset, communications link between board and host CPU must be re-established through initializing.)
Ø3H through FFH	Reserved.

3.4.2 I/O COMMUNICATIONS BLOCKS

The host CPU and the iSBC 215G board use the four blocks of system memory and one MULTIBUS I/O port to exchange instructions and status. The I/O communications blocks are entitled: wake-up block, channel control block, controller invocation block, and I/O parameter block. The iSBC 215G board uses these blocks to perform three basic functions: initialize the board, check and transmit status, and obtain user-selected drive access functions and parameters. In addition to these I/O blocks, certain board functions (such as track formatting) also require data/parameter buffers in system memory. Dedicated locations, however, are not required.

NOTE

Following the iSBC 215G board initialization, the wake-up block, channel control block, and controller invocation block must remain at the assigned locations. The location of the I/O parameter block can be changed only if the I/O parameter block pointer in the controller invocation block is changed to indicate the new location.

One I/O port in the host CPU addressable (MULTIBUS) I/O space is also required. The host CPU uses this port, called the wake-up I/O port, to initiate iSBC 215G board activity. The sequence (see Figure 3-3) in which the board 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 control and data blocks, as required, in system memory with the command and parameters for the function the iSBC 215G board is to perform (for example, read-data).

2. The host then transmits a wake-up command (Ø1H) to the wake-up I/O port, signalling the board to read the I/O communications blocks for instructions.

3. The board reads the wake-up block and links its way to the channel control block, through the controller invocation block, to the I/O parameter block. (The wake-up block is used once during board initialization and by host iSBC 215G board IOP firmware. All subsequent wake-up commands cause the iSBC board to read the channel control block.)

4. At the I/O parameter block, the iSBC 215G board reads the command and parameter data into local RAM and begins the data transfer function.

5. The board reads data from the selected drive into local RAM, then DMA-transfers the data from RAM into system memory.

6. When the data transfer is complete, the board posts the status in the controller invocation block, sends an interrupt to the host CPU, and awaits further instructions.

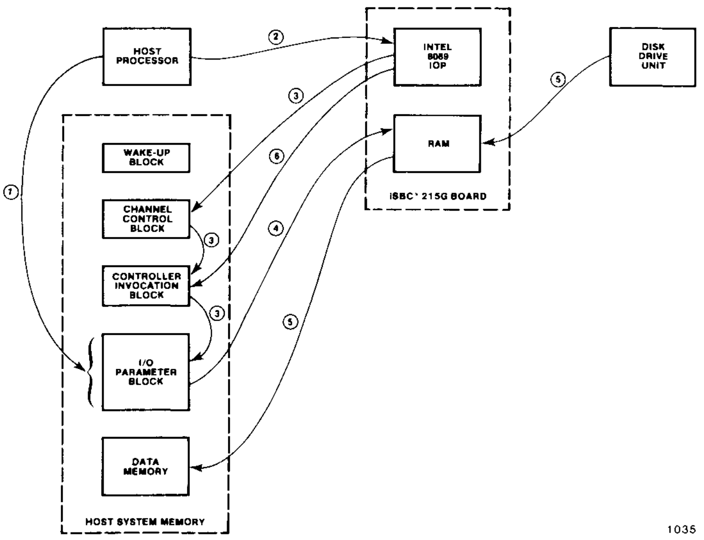


Figure 3-3. Host CPU/Board Interaction

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

The host CPU initiates board activity through the wake-up I/O port, which it addresses through the MULTIBUS interface. The 8Ø89 I/O Processor (IOP) handles all communications between the host CPU, host memory, and disk drives, once the host has initiated board activity. Board operations software is contained in on-board PROM. Local RAM on the board facilitates intermediate data storage between the host CPU and the disk drive. The iSBX bus provides a second I/O transfer path between the board and an I/O controller such as the iSBX 218A Flexible Diskette Controller Board.

Note that, in the following command block descriptions, all bytes shown as reserved in the illustrations should be set to Ø unless specified otherwise. Also note that some of the unused bytes are intended for future expansion or are required for compatibility with other devices that use a similar command structure.

3.4.2.1 Wake-Up Block

The wake-up block (see Figure 3-4 and Table 3-2) is used to establish a link between the board and the I/O communications blocks in host system memory. It is the first of the I/O communications blocks and requires 6 bytes of memory. It is used once at board initialization by the host CPU and by iSBC 215G board IOP firmware.

Note that the first address in the wake-up block in system memory is the wake-up address and that it is selected by the same set of jumpers that select the I/O port address. The hexadecimal value contained in the configuration of these jumpers is multiplied by 2^4 (that is, shifted four places to the left) to derive the 2Ø-bit MULTIBUS address. (Thus, the jumper selected wake-up address is used as the segment value, with an offset value of Ø, to derive the actual wake-up address.) Upon recognition of the first I/O start command to the wake-up I/O port, the board starts reading the wake-up block at this address. It fetches the wake-up block and internally saves the channel control block address (the next link in the communications blocks chain). This operation is necessary only after a board reset.

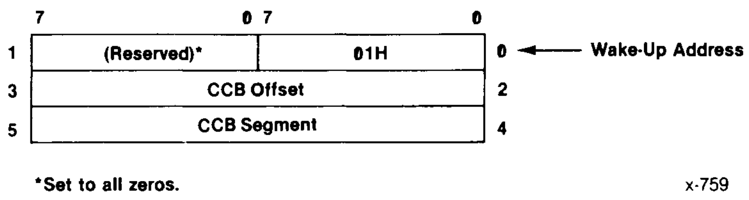


Figure 3-4. Wake-Up Block

Table 3-2. Wake-Up Block Byte Contents

Byte	Function
Ø	SYSTEM OPERATION COMMAND -- Must be set to Ø1H.
1	Reserved.
2 through 5	CHANNEL CONTROL BLOCK ADDRESS -- Address (segment X 2 + offset) of first byte in channel control block.

3.4.2.2 Channel Control Block

The channel control block (see Figure 3-5 and Table 3-3) indicates the IOP status IOP and invokes program operations. It requires 16 bytes of memory (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 board operations that are transparent to the host CPU. The busy-1 flag is posted (by the iSBC 215G board, except during cold-start initialization) when the board is busy processing a command, and cleared (also by the iSBC 215G board) after the processing is completed (when the IOP halts). It is used in handshaking and status commands between the iSBC 215G board and the host CPU.

The channel control and controller invocation block addresses are stored in the iSBC 215G board logic while processing the first start I/O command to the wake-up I/O port after a reset operation. These addresses <u>may not</u> be changed without also commanding a board reset and initialization.

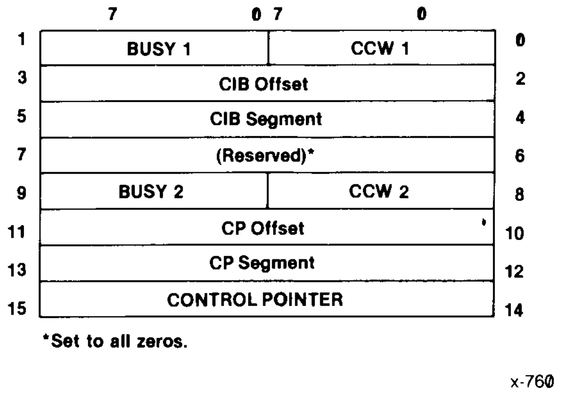


Figure 3-5. Channel Control Block

Table 3-3. Channel Control Block Byte Contents

Byte	Function
Ø	CHANNEL CONTROL WORD 1 -- Indicates location of IOP control store program. Ø1H -- On-board ROM. Ø3H -- Host system memory (used only when executing user-written I/O program from host memory).
1	BUSY 1 FLAG -- Indicates board state (busy or idle). ØØH -- Idle FFH -- Busy
2 through 5	INVOCATION BLOCK ADDRESS -- Address of fifth byte of controller invocation block.
6 and 7	Reserved.
8	CHANNEL CONTROL WORD 2 -- Must contain Ø1H.
9	BUSY 2 WORD -- Not used by host CPU.
1Ø through 13	CONTROL POINTER ADDRESS -- Must point to control pointer address in bytes 14 and 15.
14 and 15	CONTROL POINTER -- Must be set to ØØØ4H.

3.4.2.3 Controller Invocation Block

The controller invocation block (see Figure 3-6 and Table 3-4) posts status to the host CPU and locates the starting address for the on-board drive interface program. The status semaphore byte (byte 3) has a special purpose. The host CPU uses this byte to indicate to the board 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 of memory.

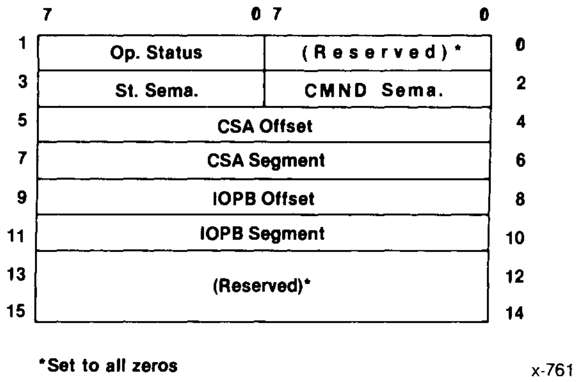


Figure 3-6. Controller Invocation Block

Table 3-4. Controller Invocation Block Byte Contents

Byte	Function
Ø	Reserved.
1	BOARD OPERATION STATUS -- Indicates board status (see paragraph 3.7.
2	COMMAND SEMAPHORE -- Board does not use byte. (Byte functions as multiprocessor interlock when required.)
3	STATUS SEMAPHORE -- Indicates state of status posting. Board posts status only when byte contains ØØH. When new status has been posted, board sets byte to FFH. When host CPU has read status, it sets byte to ØØH.
4 through 7	CONTROL STORE PROGRAM ADDRESS -- Starting address of on-board mass storage interface program. Set to ØØØØH.
8 through 11	I/O PARAMETER BLOCK ADDRESS -- Address of first byte of parameter block.
12 through 15	Reserved.

3.4.2.4 I/O PARAMETER BLOCK

The I/O parameter block (see Figure 3-7 and Table 3-5) functions as the primary communications channel between the host CPU and the iSBC 215G board. It contains the board operating commands, which define the function the board is to perform (read, write, etc.), and the parameters of the function (memory address, disk head, cylinder, etc). The I/O parameter block requires 3Ø bytes of memory; however, the first four bytes are reserved for expansion and must always be set to all Ø's.

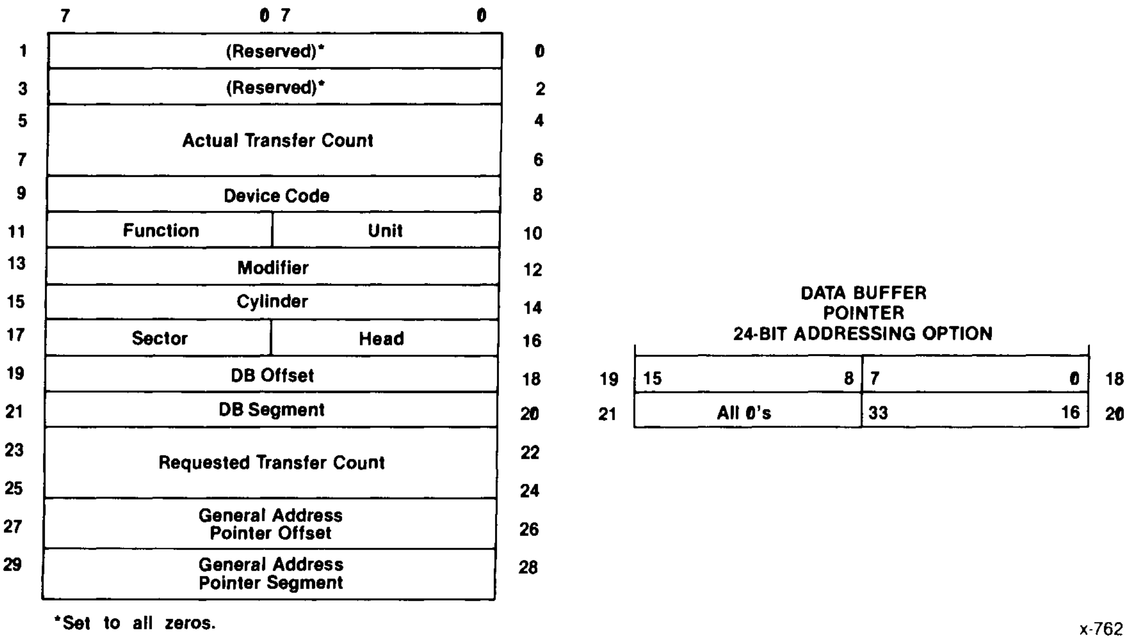


Figure 3-7. I/O Parameter Block

Table 3-5. I/O Parameter Block Byte Contents

Byte	Function
Ø through 3	Reserved.
4 through 7	ACTUAL TRANSFER COUNT -- Four-byte binary number, least significant bits in first byte. Indicates count of bytes actually transferred between host CPU and iSBC 215G board.
8 and 9	DEVICE CODE -- Code for type of device being accessed. ØØØØH -- hard-disk. ØØØ1H -- 8-in flexible-disk drive. ØØØ3H -- 5 1/4-in flexible-disk drive. ØØØ4H -- QIC-Ø2 cartridge-tape drive.
1Ø	UNIT -- Indicates binary number for drive being accessed (bits Ø and 1 provide four numbers, bits 2 through 7 are reserved).
11	FUNCTION -- Indicates code for operation to be performed (see paragraph 3.5)
12 and 13	MODIFIER -- Two-byte binary word. Indicates code to modify function codes (see paragraph 3.6).
14 and 15	CYLINDER -- Two-byte binary number, bit Ø is least significant. Indicates logical cylinder code.
16	HEAD -- One-byte binary number, bit Ø is least significant. Indicates logical head code.
17	SECTOR -- One-byte binary number, bit Ø is least significant. Indicates logical sector code.
18 through 21	DATA BUFFER ADDRESS -- Indicates address of first byte in host memory data (parameter) buffer. (For 24-bit addressing information, refer to paragraph 3.6.)
22 through 25	REQUESTED TRANSFER COUNT -- Four-byte binary number, least significant bits in first byte. Indicates count of bytes requested to be transferred between host CPU and iSBC 215G board.
26 through 29	GENERAL ADDRESS POINTER -- Indicates general-purpose address pointer. When iSBC 215G board is configured for ANSI X3T9/1226 interface, bytes have following values, where n is calculated as in Appendix A: 26 -- $n_Ø$ 27 -- n_1 28 -- n_2 29 -- n_3

As shown in Table 3-5, the board writes the actual count of bytes transferred into bytes 4 through 7 of the I/O parameter block. This is done following either termination or completion of an operation. If the count does not match the requested transfer, this indicates that the operation was terminated prematurely and that a status check is in order. When the board is to perform the track formatting operation, the host CPU writes a count of 6 into the actual-transfer-count word. When the board is to perform a status transfer, a count of 12 is written. When initializing hard-disk drive Ø, this word is used to display the board firmware and revision numbers. Bits 7 and 6 contain the version number minus 1; bits 5, 4, 3, and 2 contain the revision number.

3.4.3 COLD-START BOARD INITIALIZATION

The iSBC 215G board cold-start initialization must be performed to establish the link between the IOP and the I/O communications blocks in host system memory at any time that power has been removed from and restored to the board (before any data transfer activities between the host system memory and the drives can be initiated). After the board is initialized, any of the data transfer functions can be performed in any sequence. (Refer to paragraph 4.4.1.3.2 for a detailed explanation of board initialization.)

The following procedure outlines the sequence in which the board initializing activities are performed. Prior to initializing the board, make certain that the system data bus jumpers, the host system I/O address jumpers, the wake-up address jumpers, and the interrupt level jumper have been set as described in the jumper configuration procedures in Chapter 2.

To initialize the board, the host CPU performs the following steps:

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

Wake-Up Block	6 Bytes
Channel Control Block	6 Bytes
Invocation Block	6 Bytes
I/O Parameter Block	3Ø Bytes

NOTE

The address of the wake-up block first byte in host memory must be equal to the wake-up address set in the board wake-up address jumpers times 2_4. For example, if the jumpers are set to Ø673H, the address of byte Ø of the wake-up block is Ø673ØH for 2Ø-bit addressing and 673ØH for 16-bit addressing

2. Sets up the bytes in the wake-up block (see Figure 3-4 and Table 3-2).

3. Sets the BUSY 1 FLAG (optional, byte 1 of the channel control block) to other than Ø (FFH). Because the iSBC 215G board resets the BUSY 1 FLAG to Ø at the completion of the cold-start operation (Ø1H), the host CPU can monitor the flag to determine when the initialization procedure is completed.

4. Writes Ø2H to the wake-up I/O port to reset the iSBC 215G board.

5. Writes ØØH to the wake-up I/O port to clear the reset.

6. Writes Ø1H to the wake-up I/O port to establish the host-CPU-to-iSBC-215G-board communications link. The board reads the wake-up block in host memory and records the address of the channel control block in local RAM, then proceeds to the channel control block and clears the BUSY 1 FLAG. On all subsequent Ø1H commands to the wake-up I/O port, the board reads the channel control block.

3.5 FUNCTION COMMANDS

The function commands included in the iSBC 215G board firmware take full advantage of the capabilities of the board and its attached peripheral devices. These commands provide for a full set of operations for the hard-disk drives attached directly to the board, and also include a set that is used specifically for flexible-disk and cartridge-tape drives attached via iSBX 218A and iSBX 217B/C boards. Modified definitions for some of the hard-disk commands, combined with the additional commands for the iSBX boards, allow direct use of these boards while using the same general programming used with the iSBC 215G board. Each of the function commands is invoked by setting up the command blocks as required for the command and then issuing a start operation (Ø1H) command to the wake-up port address of the iSBC 215G board.

Table 3-6 lists all of the function commands and includes device applicability information. Each of the commands is described in detail in the following paragraphs. With the exception of the spin-down command (ØBH), all of the disk commands are similar to the commands for the iSBC 215A/B board and earlier versions. Some of the functions were enhanced at various times; however, all such enhancements default to compatibility with earlier board versions.

The functions available on the iSBC 215G board are divided into two general types: short-term functions and long-term functions. Short-term functions are those that are performed with the specified device directly on line with the iSBC 215G board. These functions terminate with the board sending a single interrupt to the host CPU (if the interrupt was not suppressed). The long-term functions are those that are initiated by the iSBC 215G board and completed off-line by the selected device. The on-line portion of the long-term function terminates with the board sending an interrupt to the host CPU (if the interrupt was not suppressed). When the selected device completes the off-line portion of the function, the iSBC 215G board sends a second interrupt to the host CPU. (The second interrupt cannot be suppressed.)

The following description of each of the commands includes a diagram of the I/O parameter block with the mandatory fields (other than reserved fields) shown for each function. In these descriptions, all of the long-term functions are so noted (short-term functions are not noted).

Table 3-6. Function Command Summary

Command	Hexadecimal Value	Hard Disk	Flexible Disk	Cartridge Tape
Initialize	ØØ	Yes	Yes	Yes
Transfer Status Buffer	Ø1	Yes	Yes	Yes
Format	Ø2	Yes	Yes	No
Read Sector ID	Ø3	Yes	Yes	No
Read Data	Ø4	Yes	Yes	Yes
Read Data to Buffer and Verify	Ø5	Yes	Yes	No
Write Data	Ø6	Yes	Yes	Yes
Write Buffer Data	Ø7	Yes	Yes	No
Initiate Track Seek	Ø8	Yes	Yes	No
Spin Down	ØB	Yes	No	No
iSBX Execute	ØC	No	No	No
iSBX Transfer	ØD	No	No	No
Buffer I/O	ØE	No	No	No
Diagnostic	ØF	Yes	Yes	No
Tape Initialize	1Ø	No	No	Yes
Rewind	11	No	No	Yes
Space Forward One File Mark	12	No	No	Yes
Write File Mark	14	No	No	Yes
Erase Tape	17	No	No	Yes
Load Tape	18	No	No	Yes
Tape Reset	1C	No	No	Yes
Retension Tape	1D	No	No	Yes
Read Tape Status	1E	No	No	Yes
Read/Write Terminate	1F	No	No	Yes

3.5.1 INITIALIZE (00H)

The initialize function transfers device-related parameters to the iSBC 215G board for subsequent use during execution of other functions. To perform this function, the host CPU establishes the following fields in the I/O parameter block:

Device Code	Bytes 8 and 9
Unit Number	Byte 1Ø
Function Code	Byte 11
Modifier	Bytes 12 and 13
Data Buffer Pointer	Bytes 18 through 21

The device parameters are specified in the data buffer area and fetched automatically by the iSBC 215G board (via the data buffer pointer) during function execution. Figure 3-8 illustrates the data buffer formats. The device parameters for tape are transferred in a single byte and require a single-byte data buffer for this purpose.

HARD DISK

	7 — 0	7 — 0	
1	NUMBER OF CYLINDERS		0
3	NOT USED	NUMBER OF HEADS	2
5	BPS (LOW)	SECTORS/TRACK	4
7	NO. ALT. CYLS.	BPS (HIGH)	6

TAPE

	7 — 0	7 — 0	
1	NOT USED	DEVICE AVAIL	0
3	NOT USED	NOT USED	2
5	NOT USED	NOT USED	4
7	NOT USED	NOT USED	6

FLEXIBLE DISK

	7 — 0	7 — 0	
1	NUMBER OF CYLINDERS		0
3	NUMBER OF HEADS	NOT USED	2
5	BPS (LOW)	SECTORS/TRACK	4
7	DRIVE VARIABLES	BPS (HIGH)	6

x-763

Figure 3-8. Initialize Function Data Buffer Format

Execution of the initialize function sets the iSBC 215G board logic for a mass storage device based on the device code and unit number specified in the I/O parameter block. Thus, to fully initialize the iSBC 215G board, the initialize function must be performed for each device attached directly to the board, to the board via the iSBX 218A board, and to the board via the iSBX 217B/C board. The iSBC 215G board firmware requires that the initialization procedures for all possible storage devices be performed, even when a particular device is not physically present; however, the iSBX 218A and 217B/C MULTIMODULE board initialization procedures need not be done if the boards are not installed.

The full initialization procedure must be performed following any interruption of power, system hardware reset, or invocation of the reset-board wake-up command. When the device specified is a disk drive (either hard or flexible), the heads are set to track Ø. When the device specified is hard-disk drive Ø, an on-board memory test is performed (unless inhibited by use of the function modifier) prior to execution of the disk drive initialization. When the on-board memory test is performed, the stored parameters for all of the storage devices are destroyed and the new parameters for hard-disk drive Ø are stored.

The following paragraphs describe the device parameters supplied to the iSBC 215G board during the initialization operation. Note that paragraphs 3.5.1.1 through 3.5.1.5 apply to hard-disk and flexible-disk drives, paragraph 3.5.1.6 applies to flexible-disk drives only, and paragraph 3.5.1.7 applies to cartridge-tape drives only.

3.5.1.1 Number of Cylinders

The number of cylinders is a two-byte hexadecimal value that specifies the total number of cylinders available on the disk drive. (Refer to the reference manual for the particular drive to determine the correct number for this parameter.) Setting the number of cylinders parameter to Ø removes the specified drive from use. A drive thus removed from service can be restored to service by performing the initialize function.

3.5.1.2 Number of Heads

The number of heads parameter is two one-byte hexadecimal values that specify the number of available recording surfaces. It is contained in byte 2 for hard-disk drives and in byte 3 for flexible disk drives. Byte 2 specifies the number of surfaces on the specified drive connected to the iSBC 215G board; byte 3 specifies the number of surfaces available on the specified drive connected to the board through the iSBX 218A board. In either instance, the unused byte should be set to all Ø's.

3.5.1.3 Sectors Per Track

The sectors per track parameter is a one-byte hexadecimal value that specifies the number of sectors available on each track on the specified drive and is contained in byte 4. For hard-sectored hard-disk drives and all flexible disk drives, this value can be obtained from the reference manual for the particular drive. For soft-sectored hard-disk drives, the number of sectors per track must be calculated from information provided in the disk drive manual. Some typical values are given in Tables 1-2 and 1-3. For ANSI hard-disk drives other than those listed in Table 1-3, refer to Appendix A to calculate the values.

3.5.1.4 Bytes Per Sector

The bytes per sector parameter is a two-byte hexadecimal value that specifies the number of bytes in a disk sector and is contained in bytes 5 and 6, with byte 6 as the most significant. The parameter value must match the formatted sector size for the specified drive. If the drive is not formatted, the sector size specified during formatting must match this value.

3.5.1.5 Number of Alternate Cylinders

The number of alternate cylinders parameter is a one-byte hexadecimal value that specifies the number of cylinders that are reserved as alternates on the drive. The parameter value must match the number of cylinders formatted as alternates for the specified drive. If the drive is not formatted, the number formatted as alternates during formatting must match this value.

3.5.1.6 Drive Variables

The drive variables parameter in byte 7 is a combination of values that specifies the recording format, head step rate, and head load delay time for flexible-disk drives only. For 5 1/4-inch disk drives, the default values are 22 ms for the head step rate and 36 ms for the head load delay time. For 8-inch drives, the default values are 11 ms for the head step rate and 6Ø ms for the head load delay time. In either instance, the default values are selected when bits 1 through 7 of this parameter are set to all Ø's. (These default values are the same as those for the iSBC 215A/B boards.) Figure 3-9 illustrates the format of the drive-variables parameter. Note that, If the default values are not selected by using all Ø's, both the step rate and head load delay must be selected using the values listed in Figure 3-9.

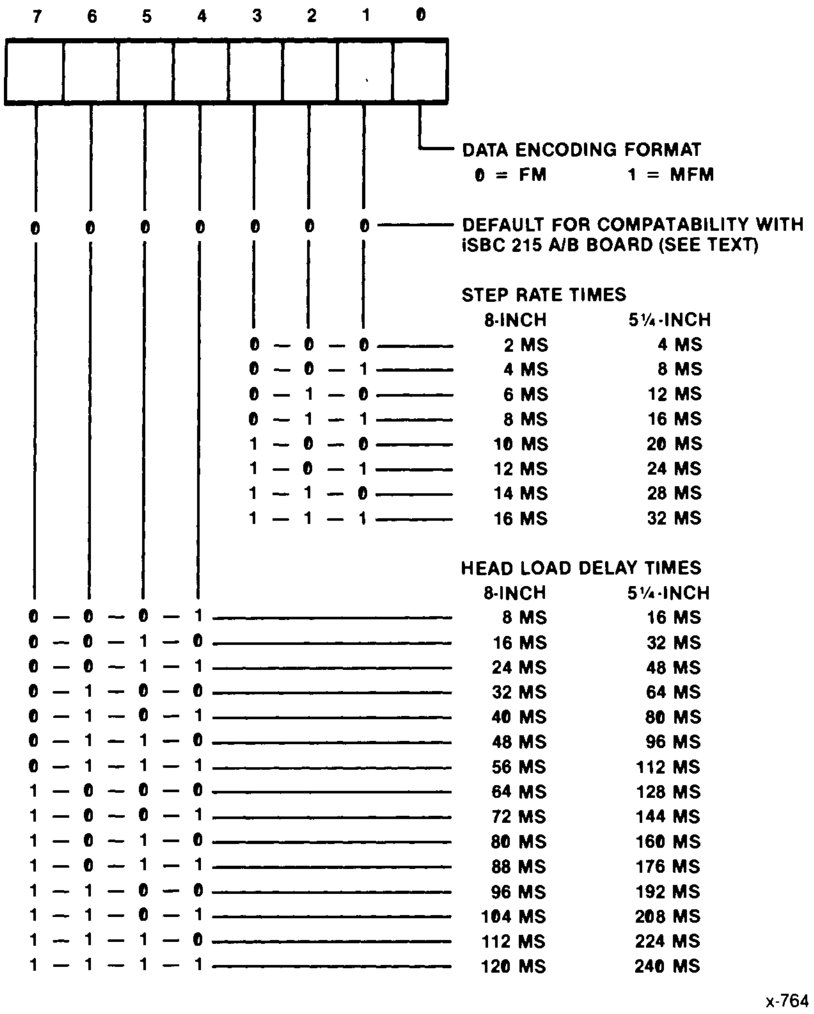


Figure 3-9. Drive Variables Byte Format

3.5.1.7 Tape Parameters

The tape available parameter is a one-byte value that specifies whether the tape device is available for use. The least significant bit indicates the device availability (Ø for device not available, 1 for device available). All other bits are reserved and should be set to Ø.

3.5.2 TRANSFER STATUS BUFFER (01H)

The transfer status buffer function transfers the contents of the iSBC 215G board 12-byte status buffer into system memory starting at the location specified by the data buffer pointer. To perform this function, the host CPU establishes the following fields in the I/O parameter block:

Unit Number	Byte 1Ø
Function Code	Byte 11
Modifier	Bytes 12 and 13
Data Buffer Pointer	Bytes 18 through 21

The host CPU can request the contents of either the short-term command status buffer or the long-term command status buffer (which is used only with some tape functions). When bit 6 of the modifier word is set to Ø, the short-term status buffer contents are transferred; when bit 6 is set to 1, the long-term status buffer contents are transferred. When the short-term buffer is specified, its contents are not affected by the transfer status buffer function. However, when the long-term buffer is specified, its contents are written into the short-term buffer over the existing contents. (The status buffer format and definitions of the status conditions are included separately later in this chapter.)

3.5.3 FORMAT (02H)

The format function partitions the addressed track for subsequent data recording (see Figure 3-1Ø). The partitioning is accomplished by writing sector headers and reserving recording space based on the initialization information for the specified disk drive. The sector headers contain information used in subsequent write or read operations to locate the correct sector data area. Each execution of the format function formats one track. To perform this function, the host CPU establishes the following fields in the I/O parameter block:

Device Code	Bytes 8 and 9
Unit Number	Byte 1Ø
Function Code	Byte 11
Modifier	Bytes 12 and 13
Cylinder Number	Bytes 14 and 15
Head Number	Byte 16
Sector One Offset	Byte 17
Data Buffer Pointer	Bytes 18 through 21

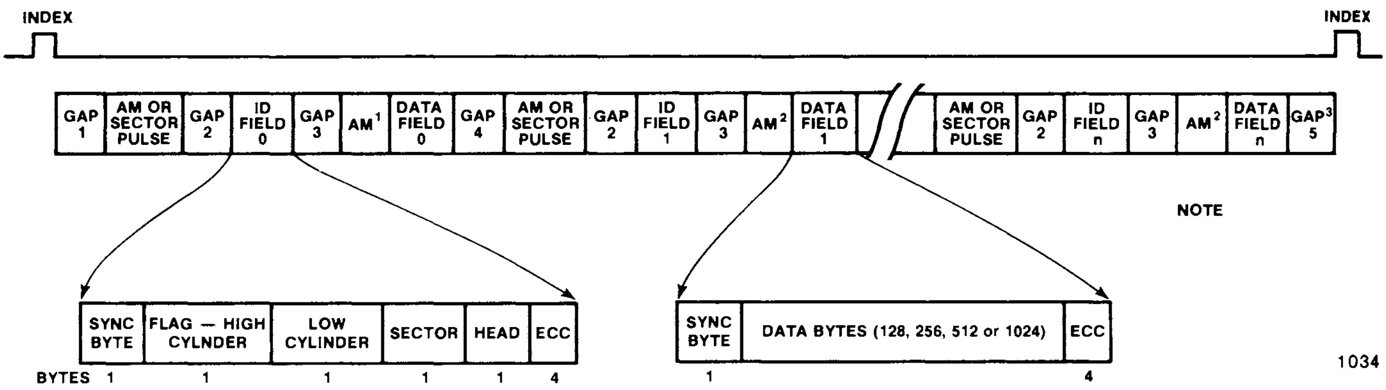


Figure 3-1Ø. Sector Data Format

Additional format parameters are specified in the data buffer area and fetched automatically by the iSBC 215G board during function execution. Figure 3-11 illustrates the data buffer content for the function.

DATA TRACK

	7 0	7 0	
1	USER PATTERN 1	00H	0
3	USER PATTERN 3	USER PATTERN 2	2
5	INTERLEAVE	USER PATTERN 4	4

DEFECTIVE TRACK

	7 0	7 0	
1	ALT CYL LSB	80H	0
3	ALTERNATE HEAD	ALT CYL MSB	2
5	INTERLEAVE	00H	4

ASSIGNED ALTERNATE TRACK

	7 0	7 0	
1	USER PATTERN 1	40H	0
3	USER PATTERN 3	USER PATTERN 2	2
5	INTERLEAVE	USER PATTERN 4	4

x-765

Figure 3-11. Format Function Data Buffer Format

There are three options for formatting a track. Byte Ø in the data buffer specifies the type of format function required. Most tracks are formatted as standard data types. When a track is determined to have a medium defect, it is formatted as a defective track and provides a pointer to the alternate track used in its place. At format time, a few tracks on the disk are reserved as alternate data tracks. When an alternate track is used, it is formatted as an assigned alternate track. Note that all iSBX 218A board format functions also must specify the format type. With 5 1/4-inch disk drives using a 512-byte sector size, either 8 or 9 sectors per track may be specified with iSBC 215G boards identified as PBA number 144263-Ø14 or PBA number 146484-ØØ1.

When formatting data tracks and assigned alternate tracks, bytes 1 through 4 provide a 4-byte user-specified pattern. This pattern is repeatedly recorded into each sector data area during track formatting, and can be any 4-byte pattern. Typically, some form of worst-case pattern is used as a test of the medium integrity. When formatting a defective track, bytes 1 and 2 specify the cylinder number and byte 3 specifies the head number for the alternate track to which it is pointed. As with the data and alternate tracks, the content of bytes 1 through 4 are repeatedly recorded into each sector data area during track formatting as a pointer to the assigned alternate track.

Byte 5 in the data buffer specifies the interleave factor for the track. The interleave factor controls the order of the sectors on the track, and is the minimum number of sector intervals between the start of one sector and the start of the next sequential sector. For example, when an interleave factor of one is specified, the sector numbers are sequential. Greater interleave factors allow increased disk rotation time between sequential numbers. This allows the host CPU to prepare for the next data transfer before the next sequential sector arrives at the read/write head. Host CPU processing time is an important consideration in determining the ideal interleave factor for the iSBC 215G board. Performance tests with typical applications programs are suggested to determine the ideal factor.

3.5.4 READ SECTOR ID (03H)

The read-sector ID function searches for the first error-free sector header on the presently selected cylinder and head and, when the header is located, transfers the contents of the sector ID field into system memory, starting at the location specified by the data buffer pointer.

To perform this function, the host CPU establishes the following fields in the I/O parameter block:

Device Code	Bytes 8 and 9
Unit Number	Byte 1Ø
Function Code	Byte 11
Modifier	Bytes 12 and 13
Data Buffer Pointer	Bytes 18 through 21

Because the read-sector-ID function is typically used to verify disk position, an implicit seek is not performed. The information from the sector ID field is stored in the data buffer automatically by the iSBC 215G board during function execution. Figure 3-12 illustrates the data buffer and flag byte format.

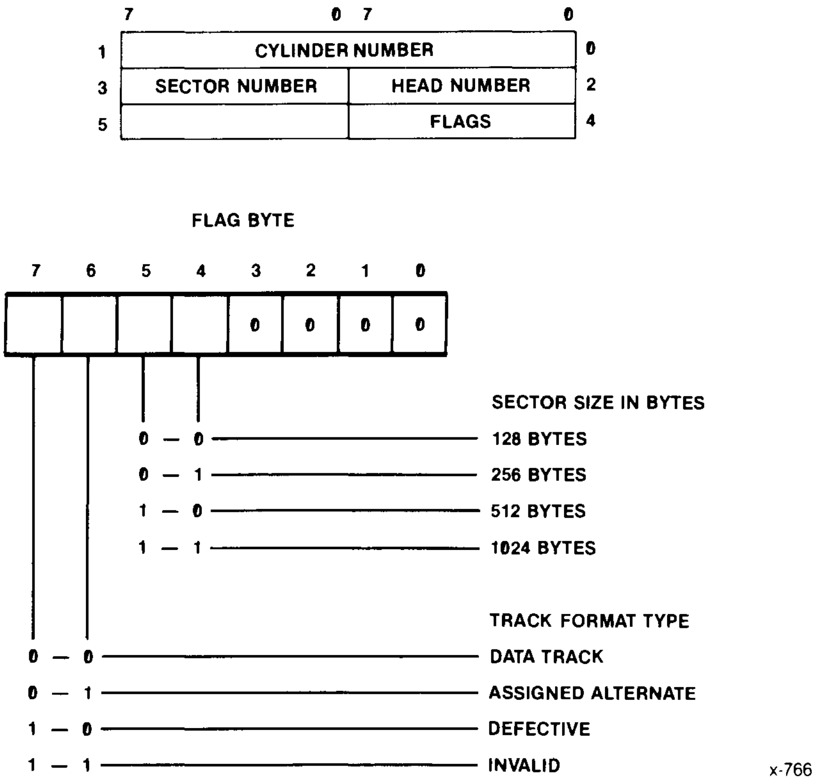


Figure 3-12. Read-Sector-ID Function Data Buffer and Flag Byte

3.5.5 READ DATA (04H)

The read data function transfers a block of data from the specified device into system memory, starting at the location specified by the data buffer pointer. To perform the read data function, the host CPU establishes the following fields in the I/O parameter block:

Device Code	Bytes 8 and 9
Unit Number	Byte 1Ø
Function Code	Byte 11
Modifier	Bytes 12 and 13
Cylinder Number	Bytes 14 and 15
Head Number	Byte 16
Starting Sector Number	Byte 17
Data Buffer Pointer	Bytes 18 through 21
Requested Transfer Count	Bytes 22 through 25

3.5.5.1 Disk Read Details

When a disk-read function is initiated, the iSBC 215G board compares the presently selected cylinder number with the requested number. If the two are not the same, the board initiates a seek function to the requested cylinder (implicit seek). When the requested cylinder is reached and the requested head selected, the board starts scanning sector headers for the requested sector. When the requested sector is located, the contents of the sector data field are written into the board sector buffer and error checking and correction (if required) are performed. The contents of the sector buffer are then transferred into system memory, starting at the data buffer pointer location.

If the first sector transferred satisfies the requested transfer count, the read-data function is terminated and the status is posted. If the requested transfer count is not satisfied, the next logical sector is transferred in the same manner as the first. This process continues until the requested transfer count is satisfied or end-of-medium is detected. (End-of-medium is defined as the highest cylinder, head, and sector numbers possible for a given volume, as specified in the initialization parameters.)

Additional implicit seeks are performed until the requested transfer count is satisfied. If the requested transfer count does not match an even sector boundary, only the amount of data required to satisfy the requested transfer count is transferred from the last sector accessed.

3.5.5.2 Tape Read Details

The tape read-data function uses the same function code as the disk read operation; however, the cylinder, head, and sector parameters and error correction are not used.

The tape read function is also always a complete operation. A complete tape read operation can consist of one or more tape read functions and must be both opened and terminated. The first tape read function in a sequence opens the tape read operation. Once the operation is opened, disk functions (either hard or flexible) can be interleaved with subsequent tape read functions. (Note that the only tape function permitted until the tape read operation is terminated is a tape read function.) The individual tape read functions within a tape read operation are closed when the requested transfer count is satisfied. The requested transfer count can be any value from 1 byte to 16 Mbytes.

The tape read-data operation remains open until one of the following conditions is satisfied:

FILE MARK TERMINATION -- The tape read operation termination results when a file mark is encountered. In such a case, the operation status byte indicates a summary error and operation-completed status (89H). To determine if the summary error resulted from a normal file-mark termination or an error condition, it is necessary to transfer and examine the 12-byte short-term status buffer. For a file-mark-induced termination, the status buffer contents indicate a length error (if the requested transfer count was not satisfied) and file mark detected.

COMMAND TERMINATION -- The host CPU can terminate a read-data operation by initiating the read/write-termination function. In terminating a normal read-data operation, the read/write-termination function initiates tape rewind to the beginning-of-tape marker. The operation status byte indicates a summary error and operation-completed status (89H), and the short-term status buffer indicates a buffer under-run/over-run error and beginning-of-tape marker detected.

BLANK TAPE TERMINATION -- If the read-data function is attempted on blank tape, the function is automatically terminated after a few inches of blank tape have passed the read head. The operation status byte indicates a hard error and a summary error. The short-term status buffer indicates a length error, soft data check, tape data check, and no data detected. The tape is not automatically rewound.

If the iSBC 215G board fails to maintain the data transfer rate required by the tape drive (typically 2ØØ K bytes per second), an over-run occurs, the iSBC 215G board closes the read-data function, and the tape drive automatically stops and repositions the tape. The next read-data function resumes without loss of data. The status for the tape read-data function closed by the over-run indicates that repositioning was required with a buffer under-run/over-run error posted in the short-term status buffer.

3.5.6 READ TO BUFFER AND VERIFY (05H)

The read to buffer and verify function transfers a block of data from the specified disk drive, one sector at a time, into the iSBC 215G board RAM buffer and checks each sector read for an error correcting code (ECC). To perform the read to buffer and verify function, the host CPU establishes the following fields in the I/O parameter block.

Device code	Bytes 8 and 9
Unit Number	Byte 1Ø
Function Code	Byte 11
Modifier	Bytes 12 and 13
Cylinder Number	Bytes 14 and 15
Head Number	Byte 16
Starting Sector Number	Byte 17
Requested Transfer Count	Bytes 22 through 25

The data transferred from the disk into the buffer memory can also be sent to a device attached to one of the iSBX connectors. To perform this operation, the read to buffer and verify function is followed by either the iSBX execute function or the write buffer data function.

By specifying one disk device for the read buffer data and verify function and a different disk device for the write buffer data function, device-to-device transfers can be accomplished without transferring the data into system memory. However, this must be done at the sector level of granularity. Also, the iSBX 217B/C board and an attached tape drive cannot be specified as one of the devices.

3.5.7 WRITE DATA (06H)

The write data function transfers a block of data from system memory, starting at the location specified by the data buffer pointer to the specified device. To perform the write-data function, the host CPU establishes the following fields in the I/O parameter block:

Device Code	Bytes 8 and 9
Unit Number	Byte 1Ø
Function Code	Byte 11
Modifier	Bytes 12 and 13
Cylinder Number	Bytes 14 and 15
Head Number	Byte 16
Starting Sector Number	Byte 17
Data Buffer Pointer	Bytes 18 through 21
Requested Transfer Count	Bytes 22 through 25

3.5.7.1 Disk Write Details

The disk write function is very similar to the disk-read function except for the direction of data movement. When a disk-write function is initiated, the iSBC 215G board compares the present cylinder number and the requested cylinder number. If the two are not the same, the board initiates a seek function to the requested cylinder (implicit seek). The board then fills the on-board sector buffer with data from system memory starting at the data buffer pointer location.

When the sector buffer contains enough data to fill one disk sector, the board prepares to write the data to the disk. It starts scanning sector headers in search of the requested sector. When the sector is located, the contents of the sector buffer are written into the sector data field. While the data are being written, a polynomial check number is calculated. It is written immediately following the last data byte.

If the first sector of data transferred satisfies the requested transfer count, the write-data function is terminated and status is posted. If the requested transfer count is not satisfied, the next logical sector is written in the same manner as the first. This process continues until the requested transfer count is satisfied or end-of-medium is detected. Additional implicit seeks are performed until the requested transfer count is satisfied. If the requested transfer count does not match an even sector boundary, only the amount of data required to satisfy the requested transfer count is transferred into the last sector accessed, and the remainder of the sector is filled with Ø's.

3.5.7.2 Tape Write Details

The tape write data function uses the same function code as the disk write operation; however, the cylinder, head, and sector parameters and error correction are not used.

The tape write function is also always a complete operation. A complete tape write operation can consist of one or more tape write functions and must be both opened and terminated. The first tape write function in a sequence opens the tape write operation. Once the operation is opened, disk functions (either hard or flexible) can be interleaved with subsequent tape write functions. (Note that the only tape function permitted until the tape write operation is terminated is a tape write function.) The individual tape write functions within a tape write operation are closed when the requested transfer count is satisfied. The requested transfer count can be any value from 1 byte to 16 Mbytes. However, the total transfer count must be divisible by 512.

The tape write-data operation remains open until the read/write-terminate function is executed to write the file mark and rewind the tape to the beginning-of-tape marker.

If the iSBC 215G board fails to maintain the data transfer rate required by the tape drive (typically 2ØØ K bytes per second), an over-run occurs, the iSBC 215G board closes the tape write function, and the tape drive automatically stops and repositions the tape. The next write-data function resumes without loss of data. The status for the tape write-data function closed by the over-run indicates that tape repositioning was required with a buffer under-run/over-run error posted in the short-term status buffer

3.5.8 WRITE BUFFER DATA (07H)

The write buffer data function writes the data present in the sector buffer to the specified disk drive. When the requested transfer count exceeds the sector size, the write buffer data function writes the same sector buffer contents into the data field of the next logical sector. To perform the write data buffer function, the host CPU establishes the following fields in the I/O parameter block:

Device Code	Bytes 8 and 9
Unit Number	Byte 1Ø
Function Code	Byte 11
Modifier	Bytes 12 and 13
Cylinder Number	Bytes 14 and 15
Head Number	Byte 16
Starting Sector Number	Byte 17
Requested Transfer Count	Bytes 22 through 25

The data transferred from the disk into the buffer memory can also be supplied to a device attached to one of the iSBX connectors. To perform this operation, the read buffer and verify function is followed by either the iSBX execute function or the write buffer data function. By specifying one disk device for the read to buffer and verify function and a different disk device for the write buffer data function, device-to-device transfers can be made without transferring the data into system memory. However, this must be done at the sector level of granularity. Also, the iSBX 217B/C board and an attached tape drive cannot be specified as one of the devices.

3.5.9 INITIATE TRACK SEEK (08H)

The initiate track seek function positions the read/write heads of the specified drive without initiating a data transfer. To perform the initiate track seek function, the host CPU establishes the following fields in the I/O parameter block:

Device Code	Bytes 8 and 9
Unit Number	Byte 1Ø
Function Code	Byte 11
Modifier	Bytes 12 and 13
Cylinder Number	Bytes 14 and 15
Head Number	Byte 16

Each data transfer function causes an implicit seek if the read/write heads are not located at the desired cylinder. However, if the implicit seek capability is used, any other device connected the iSBC 215G board is un-available to the host CPU until the selected operation is completed.

The iSBC 215G board accomplishes the initiate track seek function by directing the specified disk drive to perform an off-line seek. Once the off-line seek has been started, the board terminates the initiate track seek function, posts the operation-complete status, and sends an interrupt to the host CPU (if interrupts are not suppressed). When the disk drive completes the off-line seek, the iSBC 215G board posts the seek-complete status and sends a second interrupt to the host CPU. (The second interrupt cannot be suppressed.)

Note that, if it is necessary that the host CPU examine the 12-byte status buffer relative to the off-line seek operation, it must request the contents of the long-term status buffer. If the iSBC 215G board is busy with another device when an off-line seek operation is completed, the board completes the other function, posts the appropriate short-term status, and sends an interrupt to the host CPU (if interrupts are not suppressed). After the host CPU services the interrupt and resets the status semaphore, the iSBC 215G board posts the seek-complete status and again sends an interrupt to the host CPU, this time to indicate completion of the off-line seek operation.

It is possible for two or more disk drives to perform concurrent off-line seek operations; however, the iSBC 215G board is limited to one active short-term function while an off-line seek operation (or operations) is (are) in progress. If another function is attempted with a disk drive that is performing an off-line seek, the result is a seek-in-progress error. An initiate track seek function that specifies a destination cylinder with a number greater than the total number of cylinders available results in an automatic seek to track Ø, and an invalid-address error is posted in the short-term status buffer.

3.5.10 SPIN DOWN (0BH)

The spin down function directs the disk drive to prepare for an interruption of power. Typically, a disk drive that recognizes the spin down function moves the heads to a safe parking area on the disk surface. If the disk drive does not support the spin down function, the iSBC 215G board posts the invalid-command status in the short-term status buffer and terminates the function. To perform the spin down function, the host CPU establishes the following fields in the I/O parameter blocks:

Unit Number	Byte 1Ø
Function Code	Byte 11

3.5.11 iSBX™ EXECUTE (0CH)

The iSBX execute function transfers iSBC 215G firmware control to a program stored in the on-board RAM. To perform the iSBX execute function, the host CPU establishes the following fields in the I/O parameter block:

Function Code	Byte 11
General Address Pointer	Bytes 26 through 29

The iSBX execute function allows execution of programs associated with iSBX MULTIMODULE boards other than the iSBX 218A and 217B/C boards (for which the iSBC 215G board provides programs in on-board firmware). The execution program for another iSBX MULTIMODULE board must be entered in 8Ø89 assembler code and must be down-loaded to the board RAM using the buffer-I/O function prior to initiation of the iSBX execute function.

Execution of the down-loaded program begins at the memory address specified by the general address pointer. At completion of program execution, the program must exit to iSBC 215G board ROM address ØØC5H. The remainder of the bytes in the I/O parameter block are not required by the iSBC 215G board for this function and can be used to pass parameters to the down-loaded program. The iSBX execute function is classed as a short-term operation, and includes a single interrupt at the completion of the function.

3.5.12 iSBX™ TRANSFER (0DH)

The iSBX transfer function transfers a block of data between the specified iSBX device and system memory. To perform the iSBX transfer function, the host CPU establishes the following fields in the I/O parameter block:

Function Code	Byte 11
Modifier	Bytes 12 and 13
iSBX bus I/O Port Address	Bytes 14 and 15 (cylinder number)
Transfer Parameters	Byte 16 (head number)
Data Buffer Pointer	Bytes 18 through 21
Requested Transfer Count	Bytes 22 through 25

The iSBX transfer function allows use of the less complex iSBX MULTIMODULE boards on the iSBC 215G board without additional programming in 8Ø89 assembler code. This function re-defines the I/O parameter block slightly to accommodate passing the iSBX MULTIMODULE board parameters to the iSBC 215G board. The iSBX port address is specified in I/O parameter block bytes 14 and 15 and a set of transfer parameters are specified in byte 16. Table 3-7 lists the iSBX port addresses for the iSBC 215G board and Figure 3-13 illustrates the format of the transfer parameter byte. In common with the other data transfer functions, the data buffer pointer specifies the first address of the data buffer in system memory and the requested transfer count specifies the number of bytes to be transferred. The iSBX transfer function is classed as a short-term operation, and includes a single interrupt at the completion of the function.

Table 3-7. iSBX™ Bus I/O Port Addresses (Hexadecimal)

Port	J3 Channel Ø	J3 Channel 1	J4 Channel Ø	J4 Channel 1
Ø	CØ7Ø	CØBØ	CØDØ	CØEØ
1	CØ72	CØB2	CØD2	CØE2
2	CØ74	CØB4	CØD4	CØE4
3	CØ76	CØB6	CØD6	CØE6
4	CØ78	CØB8	CØD8	CØE8
5	CØ7A	CØBA	CØDA	CØEA
6	CØ7C	CØBC	CØDC	CØEC
7	CØ7E	CØBE	CØDE	CØEE

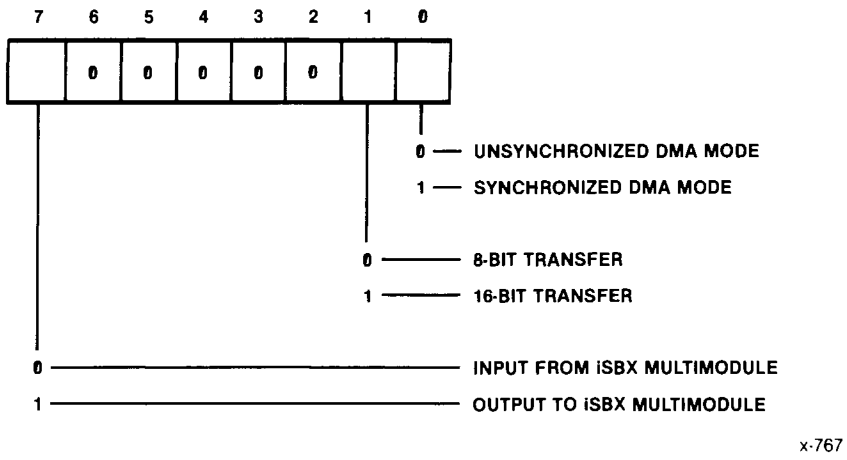


Figure 3-13. Transfer Parameter Byte Format (iSBX™ MULTIMODULE™ Board)

3.5.13 BUFFER I/O (0EH)

The buffer I/O function transfers a block of data from the system memory into the iSBC 215G board RAM buffer. To perform the buffer I/O function, the host CPU establishes the following fields in the I/O parameter block:

Unit Number	Byte 1Ø
Function Code	Byte 11
Modifier	Bytes 12 and 13
Board Memory Pointer	Bytes 14 and 15 (cylinder number)
I/O Specifier (Input = ØØH, Output = FFH)	Byte 16 (head number)
Data Buffer Pointer	Bytes 18 through 21
Requested Transfer Count	Bytes 22 through 25

The buffer I/O function allows the host CPU to transfer data between the iSBC 215G board RAM and a system memory buffer. This function is used for diagnostic purposes, to fill the iSBC 215G board sector buffer for a subsequent write buffer data function, and to down-load an iSBX program for later execution. The function redefines the I/O parameter block slightly to accommodate passing the starting on-board RAM address and the transfer direction to the board. (The board memory address pointer is specified in I/O parameter block bytes 14 and 15 and must be in the range of 4ØØØH to 45FFH; the transfer direction is specified in byte 16.) In common with other data transfer functions, the data buffer pointer specifies the first address of the data buffer in system memory and the requested transfer count specifies the number of bytes to be transferred. The buffer I/O function is classed as a short-term operation, and includes a single interrupt at the completion of the function.

3.5.14 DIAGNOSTIC (0FH)

The diagnostic function initiates a go/no-go self test, contained in the iSBC 215G firmware, that verifies the internal data and status logic in the disk drives. To perform the diagnostic function, the host CPU establishes the following fields in the I/O parameter block:

Device Code	Bytes 8 and 9
Unit Number	Byte 1Ø
Function Code	Byte 11
Modifier	Bytes 12 and 13

The diagnostic function reserves the use of the highest cylinder (track) number on head Ø for execution of the program. This track cannot be used for data storage. When the function is initiated, the cylinder and head are selected automatically; the unit number is specified in the I/O parameter block. The following three additional modifiers (specified in modifier byte 13) specifically apply to the diagnostic function.

ØØH -- Re-calibrates, then initiates a seek to the highest cylinder number of head Ø. At seek completion, the iSBC 215G board performs a read-sector-ID function to verify track location before performing a write and read test on sector Ø using a 55AAH data pattern.

Ø1H -- Initiates a ROM check-sum test to verify the contents of the iSBC 215G board ROM.

Ø2H -- Initiates a seek to cylinder Ø. At seek completion, the iSBC 215G board performs a read-sector-ID function to verify that the heads are located at cylinder Ø.

3.5.15 TAPE INITIALIZATION (10H)

The tape initialization function is the second function in a four-step initialization process for the specified QIC-Ø2 tape drive. The initialization process is started in the initialize function (ØØH) and is completed following the tape initialization function by performing the tape reset function and then the load tape function. To perform the tape initialization function, the host CPU establishes the following fields in the I/O parameter block:

Device Code	Bytes 8 and 9
Unit Number	Byte 1Ø
Function Code	Byte 11

Initializing the tape drives attached to the iSBC 215G board via the iSBX 217B/C board is a four-step process. In the first step, the host CPU performs the initialize function (ØØH) to initialize the iSBC 215G board. In this, the second step, the iSBC 215G board initializes the iSBX 217B/C board. In the third step, tape reset, the tape drive is initialized. Finally, in the fourth step, the load tape function is performed to position the tape for subsequent operations. The tape initialization function is classed as a short-term operation, and includes a single interrupt at the completion of the function.

3.5.16 REWIND (11H)

The rewind function returns the tape on the specified drive to the beginning of tape marker. To perform the rewind function, the host CPU establishes the following fields in the I/O parameter block:

Device Code	Bytes 8 and 9
Unit Number	Byte 1Ø
Function Code	Byte 11

Typically, the tape on a drive is rewound for one of two reasons: to return the tape to its starting point prior to removing the cartridge from the drive, or to position the tape to a known point before attempting a data transfer. The rewind operation is classed as a long-term function.

The iSBC 215G board accomplishes the rewind function by directing the iSBX 217B/C board to perform an off-line rewind on the specified tape drive. Once the operation is started, the iSBC 215G board terminates the rewind function, posts the operation complete status, and sends an interrupt to the host CPU (if interrupts are not suppressed). When the tape drive completes the off-line rewind, the iSBC 215G board sends a second interrupt to the host CPU.

Note that, if it is necessary that the host CPU examine the 12-byte status buffer relative to the off-line rewind operation, it must request the contents of the long-term status buffer. If the iSBC 215G board is busy with another device when an off-line rewind operation is completed, the board completes the other function, posts the appropriate status, and sends an interrupt to the host CPU (if interrupts are not suppressed). After the host CPU services the interrupt and resets the status semaphore, the iSBC 215G board posts the rewind complete status and again sends an interrupt to the host CPU, this time to indicate completion of the off-line rewind operation.

3.5.17 SPACE FORWARD ONE FILE MARK (12H)

The space forward one file function moves the tape forward until a file mark or end of medium is reached. To perform the function, the host CPU establishes the following fields in the I/O parameter block:

Device Code	Bytes 8 and 9
Unit Number	Byte 1Ø
Function Code	Byte 11

The iSBC 215G board accomplishes the space forward one file function by directing the iSBX 217B/C board to perform an off-line tape movement on the specified tape drive. Once the operation is started, the iSBC 215G board terminates the space forward one file function, posts the operation complete status, and sends an interrupt to the host CPU (if interrupts are not suppressed). When the tape drive reaches a file mark or detects end of medium and completes the function, the iSBC 215G board sends a second interrupt to the host CPU.

Note that, if it is necessary that the host CPU examine the 12-byte status buffer relative to the off-line space forward one file operation, it must request the contents of the long-term status buffer. If the iSBC 215G board is busy with another device when an off-line operation is completed, the board completes the other function, posts the appropriate status, and sends an interrupt to the host CPU (if interrupts are not suppressed). After the host CPU services the interrupt and resets the semaphore, the iSBC 215G board posts the operation complete status and again sends an interrupt to the host CPU, this time to indicate completion of the off-line space forward one file operation. This operation is classed as a long-term function.

3.5.18 WRITE FILE MARK (14H)

The write file mark function allows writing additional file marks at the end of a file written to the tape. (During write operations, the read/write terminate function automatically writes one file mark at the end of a file.) To perform the write file mark function, the host CPU establishes the following fields in the I/O parameter block:

Device Code	Bytes 8 and 9
Unit Number	Byte 1Ø
Function Code	Byte 11

The write file mark function writes a file mark on the tape at the position of the write head at the time of the command. This allows writing several file marks on the tape to denote special meaning to the separation between two files or to indicate end of tape. This function is classed as a short-term function.

3.5.19 ERASE TAPE (17H)

The erase tape function prepares the tape for subsequent recording by removing all existing recorded data. To perform the erase tape function, the host CPU establishes the following fields in the I/O parameter block:

Device Code	Bytes 8 and 9
Unit Number	Byte 1Ø
Function Code	Byte 11

Whenever a write data operation is performed, the tape is automatically erased just before the data is written onto the tape. However, there is some risk that this erase operation may leave some background noise in the area written onto. Also, any data from a previous recording in the area beyond the subject area are not erased. The erase tape function allows for the removal of all previous data by first rewinding the tape to the beginning of tape marker, erasing forward to the end of tape marker, and again rewinding to the beginning of tape marker.

The iSBC 215G board accomplishes the erase tape function by directing the iSBX 217B/C board to perform an off-line erase operation on the specified tape drive. Once the operation is started, the iSBC 215G board terminates the erase tape function, posts the operation complete status, and sends an interrupt to the host CPU (if interrupts are not suppressed). When the tape drive completes the function, the iSBC 215G board sends a second interrupt to the host CPU.

Note that, if it is necessary that the host CPU examine the 12-byte status buffer relative to the off-line erase tape operation, it must request the contents of the long-term status buffer. If the iSBC 215G board is busy with another device when an off-line operation is completed, the board completes the other function, posts the appropriate status, and sends an interrupt to the host CPU (if interrupts are not suppressed). After the host CPU services the interrupt and resets the status flag, the iSBC 215G board posts the operation complete status and again sends an interrupt to the host CPU, this time to indicate completion of the off-line erase-tape operation. This operation is classed as a long-term function.

3.5.20 LOAD TAPE (18H)

The load tape function is the fourth function in the tape initialization sequence and positions the tape to the beginning of tape marker. The load tape function also initiates the on-going iSBC 215G board check for tape medium change. To perform the load tape function, the host CPU establishes the following fields in the I/O parameter block:

Device Code	Bytes 8 and 9
Unit Number	Byte 1Ø
Function Code	Byte 11

3.5.21 TAPE RESET (1CH)

The tape reset function is the third function in the tape initialization sequence and is used to initialize the tape drive. To perform the tape reset function, the host CPU establishes the following fields in the I/O parameter block:

Device Code	Bytes 8 and 9
Unit Number	Byte 1Ø
Function Code	Byte 11

3.5.22 RETENSION TAPE (1DH)

The retension tape function prepares the tape for subsequent operations by moving the tape forward to the end of tape marker and then rewinding to the beginning of tape marker. This re-stacks the tape in the cartridge and assures free and easy tape movement. To perform the retension tape function, the host CPU establishes the following fields in the I/O parameter block:

Device Code	Bytes 8 and 9
Unit Number	Byte 1Ø
Function Code	Byte 11

The iSBC 215G board accomplishes the retension tape function by directing the iSBX 217B/C board to perform off-line retensioning on the specified tape drive. Once the operation is started, the iSBC 215G board terminates the function, posts the operation complete status, and sends an interrupt to the host CPU (if interrupts are not suppressed). When the tape drive completes the operation, the iSBC 215G board sends a second interrupt to the host CPU.

Note that, if it is necessary that the host CPU examine the 12-byte status buffer relative to the off-line retension tape operation, it must request the contents of the long-term status buffer. If the iSBC 215G board is busy with another device when an off-line operation is completed, the board completes the other function, posts the appropriate status, and sends an interrupt to the host CPU (if interrupts are not suppressed). After the host CPU services the interrupt and resets the status flag, the iSBC 215G board posts the operation complete status and again sends an interrupt to the host CPU, this time to indicate completion of the off-line retension tape operation. This operation is classed as a long-term function.

3.5.23 READ TAPE STATUS (1EH)

The read tape status function transfers the existing tape drive status from the iSBX 217B/C board to the iSBC 215G board short-term status buffer. The tape status is automatically transferred from the iSBX 217B/C board to the iSBC 215G board at the end of each tape operation. Typically, this function is not used unless there exists a relatively long period between tape operations. If it is required that the host CPU examine the tape drive status, it must transfer the contents of the short-term buffer to system memory using the transfer status buffer function. To perform the read tape status function, the host CPU establishes the following fields in the I/O parameter block:

Device Code	Bytes 8 and 9
Unit Number	Byte 1Ø
Function Code	Byte 11

3.5.24 READ/WRITE TERMINATE (1FH)

The read/write terminate function marks the end of a read or write operation. When the read/write terminate function is used to terminate a write tape operation, a file mark is automatically written and the tape is not rewound. The read/write terminate function is used to terminate a read operation only when the operation is being aborted (a read operation ordinarily terminates when a file mark is encountered). When a read operation is aborted using the read/write terminate function, the tape is rewound to the beginning of tape marker. This operation is classed as a short-term function. To perform the read/write terminate function, the host CPU establishes the following fields in the I/O parameter block:

Device Code	Bytes 8 and 9
Unit Number	Byte 10
Function Code	Byte 11

3.6 FUNCTION MODIFIERS

The function modifiers allow the user to change easily the default functions and tailor the iSBC 215G board to a particular application. These modifiers are specified in bytes 12 and 13 of the I/O parameter block. Each of the modifier actions is assigned to a single bit in the modifier word and each action is enabled by the presence of a 1 in that bit position. Figure 3-14 illustrates the format of the modifier word; the following paragraphs describe each of the relevant parts of that word.

SUPPRESS INTERRUPT -- The suppress interrupt modifier bit, when set to 1, directs the iSBC 215G board to suppress assertion of the interrupt at the end of a short-term function. When a long-term function is executed, the suppress interrupt modifier suppresses the first interrupt when the iSBC 215G board posts the operation complete status. The second interrupt (sent when the board signals that the off-line portion of the function has been completed) is not suppressed.

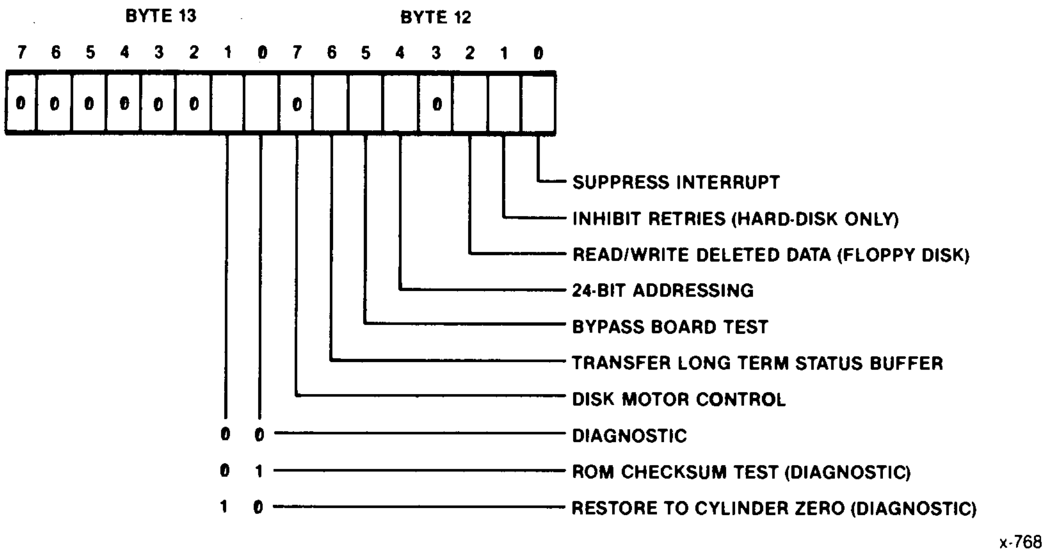


Figure 3-14. Modifier Word Format

INHIBIT RETRIES -- (Hard Disk Only) The inhibit retries modifier bit, when set to 1, directs the iSBC 215G board to attempt only once to complete a data transfer function.

READ/WRITE DELETED DATA -- The read/write deleted data modifier bit, when set to 1, allows access to the corresponding iSBX 218A board flexible disk controller command.

24-BIT ADDRESSING -- The 24-bit addressing modifier bit, when set to 1, converts the data buffer pointer format from the standard segment and offset addressing to 24-bit linear addressing. This allows placement of the data buffer anywhere in the 16-Mbyte space addressable on the MULTIBUS interface. When 24-bit addressing is used, byte 18 provides address bits ØH through 7H, byte 19 provides address bits 8H through FH, and byte 2Ø provides address bits 1ØH through 17H. Byte 21 is set to all Ø's.

BYPASS BOARD TEST -- The bypass board test modifier bit, when set to 1 during initialization of hard-disk drive Ø, causes the iSBC 215G board to skip the RAM test ordinarily executed when disk drive Ø is initialized. When the RAM test is executed, all sets of parameters in the on-board RAM are destroyed. Thus, if drive Ø is detached and then later re-attached with this modifier bit set to Ø, it is necessary to re-initialize all of the other drives to again pass the device parameters to the iSBC 215G board.

TRANSFER LONG-TERM STATUS BUFFER -- The transfer long-term status buffer modifier bit, when set to 1, converts the transfer status buffer function from a transfer of the short-term status buffer to transfer of the long-term status buffer. Because transfer of the long-term status buffer destroys the contents of the short-term status buffer, the short-term buffer contents should be transferred first.

DISK MOTOR CONTROL -- The disk motor control bit, when set to 1, invokes the motor on control for those 8-inch flexible-disk drives that require such control. Do not set this bit for 5 1/4-inch drives; motor control is assumed for those units.

ROM CHECKSUM TEST -- The ROM checksum modifier bit, when set to 1, converts the diagnostic function from the full diagnostic test to a checksum test of the iSBC 215G board ROM.

RESTORE TO CYLINDER Ø -- The restore to cylinder Ø modifier bit, when set to 1, limits the diagnostic function to restoring the read/write heads to cylinder Ø.

3.7 EXTENDED STATUS

At the end of each iSBC 215G board operation, including operations with any iSBX MULTIMODULE boards mounted on the iSBC 215G board, information pertaining to the operation is stored in one of the two status buffers provided in the on-board RAM. If the completed operation was a short-term function, only the short-term status is posted; if the completed operation was a long-term function, both the short-term and long-term status are posted. The short-term status is posted at the time the first interrupt is asserted, and the long-term status is posted when the specified device completes the off-line portion of the function. Prior to asserting the interrupt, the iSBC 215G board summarizes the status buffer into the operation status byte. Figure 3-15 illustrates the format of the operation status byte.

Bits Ø through 3 contain the code for a device specific status summary. Bits 4 and 5 are used to specify the device unit number. Bit 6 indicates (when set to 1) that a hard error occurred on the specified device and that a function could not be executed. Bit 7 indicates (when set to 1) that some kind of error occurred. If bit 7 is set to 1 and bit 6 is set to Ø, this indicates that a soft (recoverable) error occurred.

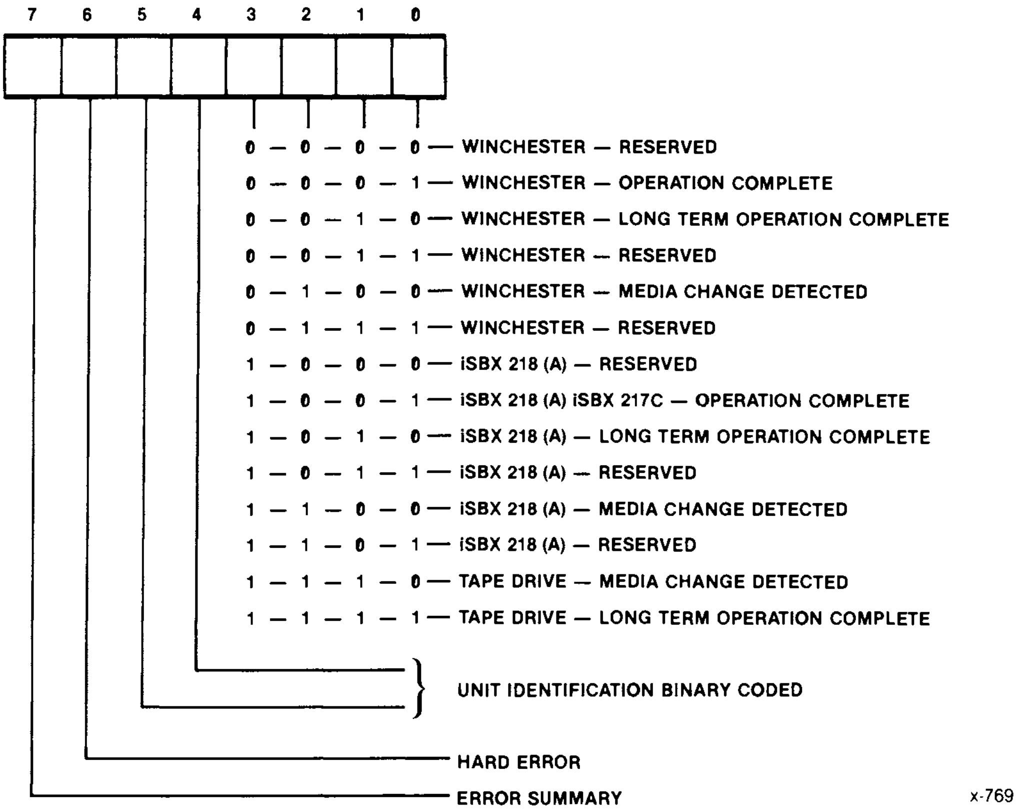


Figure 3-15. Operation Status Byte Format

3.8 STATUS OPERATIONS

Whenever the operation status byte indicates that an error occurred, the appropriate status block, short-term or long-term, can be requested by the host CPU to obtain additional information pertaining to the error condition. This information is stored in the status buffer, the contents of which are described in the following paragraphs.

3.8.1 STATUS BUFFER FORMAT

Although disk and tape operations use the same status buffers, the definitions of the bytes contained in the status buffer are different for those. Table 3-8 lists the format of the status buffer for both disk and tape operations.

Table 3-8. Error Status Buffer

Byte	Disk Function	Tape Function
Ø	Detailed Status Byte	Detailed Status Byte
1	Detailed Status Byte	Detailed Status Byte
2	Detailed Status Byte	Detailed Status Byte
3	Desired Cylinder (Low Byte)	Beginning of Tape Marker Detected
4	Desired Cylinder (High Byte)	Logical Load Point Detected
5	Desired Head and Volume	File Mark Detected
6	Desired Sector	Logical End of Tape Detected
7	Actual Cylinder (Low Byte)	Not Used
8	Flags, Actual Cylinder (High Byte)	No Data Detected (Blank Tape)
9	Actual Head and Volume	Not Used
1Ø	Actual Sector	Not Used
11	Number of Retries	Not Used

The contents for each of the status bytes are as follows:

Bytes Ø through 2 of the status buffer (for both disk and tape functions) contain the detailed error status for the last function completed.

Bytes 3 through 6 (for disk functions) list the cylinder, head, and sector address requested in the I/O parameter block for the function completed.

Byte 3 (for a tape function), when set to ØFFH, indicates that the beginning of tape marker was detected and that the tape is positioned at the start of the recording area on the tape.

Byte 4 (for a tape function), when set to ØFFH, indicates that the logical load point on the tape was detected. The logical load point is typically the fully rewound position on the tape; the beginning of tape marker is usually located a short distance before the fully rewound position.

Byte 5 (for a tape function), when set to ØFFH, indicates that the tape drive encountered a file mark during function execution.

Byte 6 (for a tape function), when set to ØFFH, indicates that the tape drive encountered the logical end of tape marker during the execution of the last function.

Bytes 7 through 1Ø (for disk functions) list the cylinder, head, and sector address actually accessed by the disk drive during the execution of the function. The status byte assigned to return the high byte of the actual cylinder number also returns additional information about the track and sector (flags). Figure 3-16 illustrates byte 8 of the status buffer for disk operations.

Bytes 7, 9, and 1Ø are not used for tape functions and are returned as all Ø's.

Byte 8 (for a tape function), when set to ØFFH, indicates that the tape drive was unable to detect any valid data after attempting to read several inches of blank tape.

Byte 11 (for disk functions) lists the number of retries attempted by the iSBC 215G board or the specified drive.

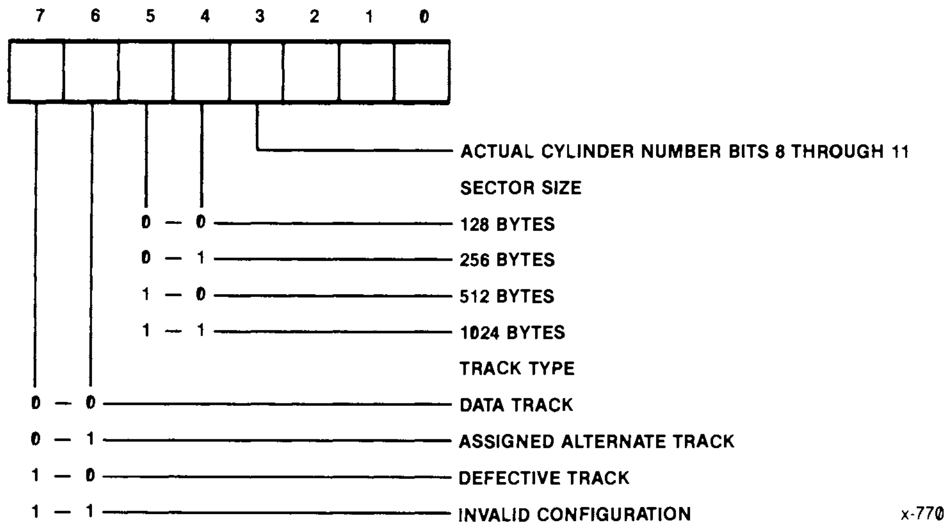


Figure 3-16. Status Buffer Byte 8 Format for Disk

3.8.2 DETAILED ERROR STATUS

The first three bytes of the status buffer provide detailed status pertaining to the last completed function. This is ordinarily called the error status. If any bit in the first three status bytes is set to 1, bit 7 in the operation status byte is set to report that an error occurred. If any bit in the first two status bytes is set to 1, bit 6 in the operation status byte reports the type: set to 1 for hard error; set to Ø for soft error. The definitions of the status bits for disk functions and tape functions are similar, but not identical. In the following descriptions of the status bits, both disk and tape definitions are included where required.

3.8.2.1 Status Byte 0

Figure 3-17 illustrates the format of status byte Ø; the paragraphs that follow the figure describe the bits of this status byte.

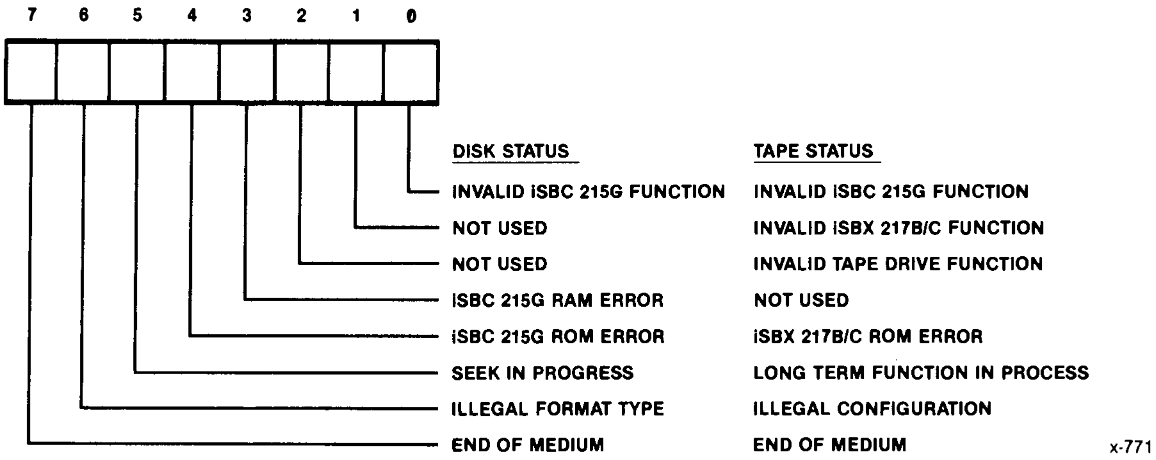


Figure 3-17. Status Buffer Byte Ø Format

Bit Ø -- INVALID iSBC 215G FUNCTION -- Bit Ø set to 1 indicates that the function code in byte 11 of the I/O parameter block was not one of the defined function codes. (The present version of the iSBC 215G board checks each function code to determine if it is defined. The iSBC 215A/B board and earlier versions of the iSBC 215G board did not make this verification check.)

Bit 1 -- INVALID iSBX 217B/C FUNCTION -- (Tape Function) Bit 1 set to 1 indicates that the function code (for the iSBX 217B/C board) in byte 11 of the I/O parameter block could not be executed by the iSBX 217B/C board. (Bit 1 is always set to Ø when disk functions are executed.)

Bit 2 -- INVALID TAPE DRIVE FUNCTION -- (Tape Function) Bit 2 set to 1 indicates that the function code (for the tape drive) in byte 11 of the I/O parameter block could not be executed by the tape drive. (Bit 2 is always set to Ø when disk functions are executed.)

Bit 3 -- iSBC 215G BOARD RAM ERROR -- (Disk Function) Bit 3 set to 1 indicates that the iSBC 215G board failed the RAM test portion of the internal diagnostic program. (Though not strictly a disk function, the RAM test function is classed as one; bit 3 is always set to Ø when tape functions are executed.)

Bit 4 -- iSBC 215G BOARD ROM ERROR -- (Disk Function) Bit 4 set to 1 indicates that the iSBC 215G board failed the ROM checksum portion of the internal diagnostic program.

Bit 4 -- iSBC 217B/C BOARD ROM ERROR -- (Tape Function) Bit 4 set to 1 indicates that the iSBX 217B/C board failed the ROM checksum test performed during the tape initialization function.

Bit 5 -- SEEK IN PROGRESS -- (Disk Function) Bit 5 set to 1 indicates that there was an off-line seek in progress in the disk drive when initiation of another function was attempted with the same drive.

Bit 5 -- LONG-TERM FUNCTION IN PROGRESS -- (Tape Function) Bit 5 set to 1 indicates that there was an off-line function in progress in the tape drive when initiation of another function was attempted with the same drive.

Bit 6 -- ILLEGAL FORMAT TYPE -- (Disk Function) Bit 6 set to 1 indicates that one of two illegal operations was attempted or that an illegal I/O parameter block format was detected. (The illegal operations are: 1) an attempt to assign an alternate to the assigned alternate track (that is, attempting to assign a second alternate track directly after finding that the assigned alternate track is defective); or 2) an attempt to access directly an iSBC 215G board unassigned alternate track as a primary data track. Also, bit 6 is set to 1 when a check of the device code, function code, or unit number in the I/O parameter block reveals that a illegal value was used.

Bit 6 -- ILLEGAL CONFIGURATION -- (Tape Function) -- Bit 6 set to 1 indicates that an attempt was made to access a tape drive that is not classed as present. As with disk functions, detection of an illegal device code, function code, or unit number in the I/O parameter block results in bit 6 being set to 1.

Bit 7 -- END OF MEDIUM -- Bit 7 set to 1 indicates that the end of medium marker was detected before the requested transfer count in the I/O parameter block was satisfied. For disk operations, the condition reported in bit 7 is classed as an error.

3.8.2.2 Status Byte 1

Figure 3-18 illustrates the format of status byte 1; the paragraphs that follow the figure describe the bits of this status byte.

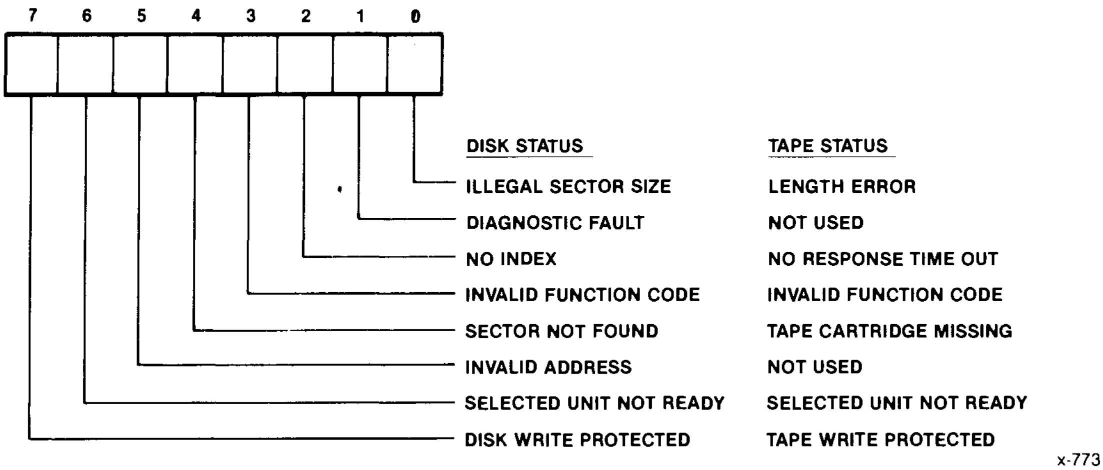


Figure 3-18. Status Buffer Byte 1 Format

Bit Ø -- ILLEGAL SECTOR SIZE -- (Disk Function) -- Bit Ø set to 1 indicates that the sector size information read from the sector header on the disk drive conflicts with the sector size specified when the initialization function was executed.

Bit Ø -- LENGTH ERROR -- (Tape Function) -- Bit Ø set to 1 indicates one of the following conditions:

The data transfer function specified a requested transfer count of Ø.

A file mark was detected with the requested transfer count un-satisfied.

The data transfer function was terminated by the iSBX 217B/C board.

Bit 1 -- DIAGNOSTIC FAULT -- (Disk Function) -- Bit 1 set to 1 indicates that the iSBC 215G board and the disk drive failed execution of the internal diagnostic program. (Though not strictly a disk function, the diagnostic fault function is classed as one; bit 1 is always set to Ø when tape functions are executed.)

Bit 2 -- NO INDEX -- (Disk Function) -- Bit 2 set to 1 indicates that the iSBC 215G board did not receive an index pulse from the disk drive. This error indicates typically that the specified disk drive is not attached to the iSBC 215G board or that power is not applied to the disk drive.

Bit 2 -- NO RESPONSE TIME OUT -- (Tape Function) Bit 2 set to 1 indicates that the iSBX 217B/C board failed to respond to an attempted access within the prescribed time.

Bit 3 -- INVALID FUNCTION CODE -- Bit 3 set to 1 (a summary error) indicates that one of the three function code error bits in status byte Ø was set to 1.

Bit 4 -- SECTOR NOT FOUND -- (Disk Function) -- Bit 4 set to 1 indicates that the iSBC 215G board failed to locate the sector number, specified in the I/O parameter block, in any of the sector ID fields in the track.

Bit 4 -- TAPE CARTRIDGE MISSING -- (Tape Function) -- Bit 4 set to 1 indicates that there is no tape cartridge installed in the specified tape drive.

Bit 5 -- INVALID ADDRESS -- (Disk Function) -- Bit 5 set to 1 indicates that an invalid cylinder, head, or sector was specified.

Bit 6 -- SELECTED UNIT NOT READY -- Bit 6 set to 1 indicates that the device specified in the I/O parameter block did not respond to an attempted access. This error typically indicates that the specified device is not attached to the iSBC 215G board, that power is not applied to the device, or that the device was manually switched off-line.

Bit 7 -- DISK/TAPE WRITE PROTECTED -- Bit 7 set to 1 indicates that an attempt was made to write to a medium, installed in the selected device, that was mechanically write-protected.

3.8.2.3 Status Byte 2

Figure 3-19 illustrates the format of status byte 2; the following paragraphs describe the bits of this status byte.

Bit Ø -- Not used. Always set to Ø.

Bit 1 -- TAPE SOFT ERROR -- (Tape Function) Bit 1 set to 1 indicates that the data transfer function with the tape drive connected to the iSBX 217B/C board was completed successfully, but that one or more retries was (were) necessary to complete the transfer. (The cause of the retry or retries can be actual data errors that were successfully written or read during retry.) Bit 1 is always set to Ø when disk functions are executed.

Bit 2 -- PARITY ERROR -- (Tape Function) Bit 2 set to 1 indicates that a data byte parity error was detected by the iSBX 217C board during a data transfer. Bit 2 is not used for disk functions or tape functions with the iSBX 217B board and is always set to Ø for such operations.

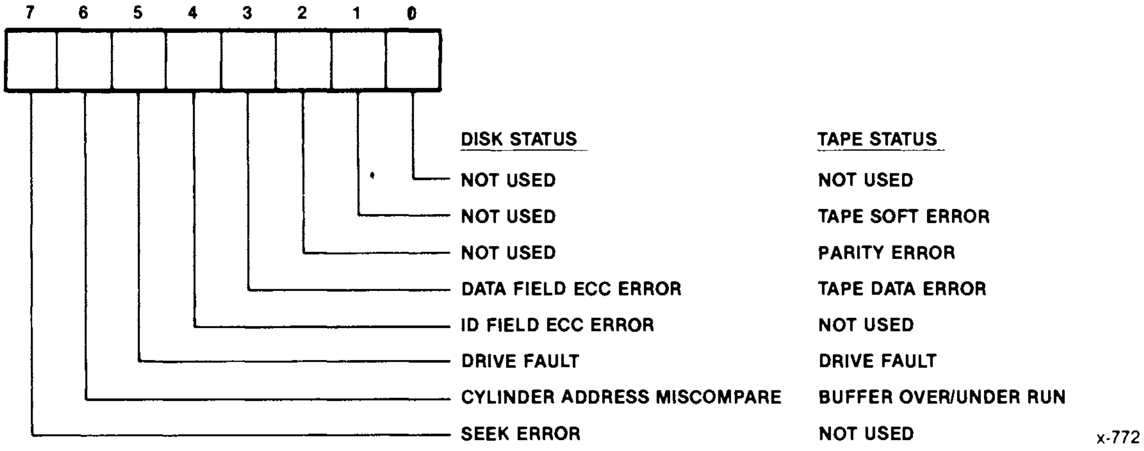


Figure 3-19. Status Buffer Byte 2 Format

Bit 3 -- DATA FIELD ECC ERROR -- (Disk Function) -- Bit 3 set to 1 indicates that the iSBC 215G board detected an error in the data field during an error correcting code (ECC) check in one of the sectors transferred. If bit 6 of the operation status byte is set to Ø, this indicates that the error was soft and is recoverable; if bit 6 is set to 1, this indicates that the error was hard and is not recoverable. Bit 3 is always set to Ø when tape functions are executed.

Bit 3 -- TAPE DATA ERROR -- (Tape Function) -- Bit 3 set to 1 indicates that the iSBC 215G board detected a data error in the file that was transferred and that the error was hard and is not recoverable. Bit 3 is always set to Ø when disk functions are executed.

Bit 4 -- ID FIELD ECC ERROR -- (Disk Function) Bit 4 set to 1 indicates that the iSBC 215G board detected an error in the ID field during an error correcting code (ECC) check in one of the sectors transferred. If bit 6 of the operation status byte is set to Ø, this indicates that the error was soft and is recoverable; if bit 6 is set to 1, this indicates that the error was hard and is not recoverable. Bit 4 is always set to Ø when tape functions are executed.

Bit 5 -- DRIVE FAULT -- Bit 5 set to 1 indicates that there is a hardware problem in the selected device.

Bit 6 -- CYLINDER ADDRESS MISCOMPARE -- (Disk Function) Bit 6 set to 1 indicates that the heads were positioned to the incorrect cylinder. The recovery process for this error is to perform a seek to track Ø and then re-attempt the seek to the desired track.

Bit 6 -- BUFFER OVER-RUN/UNDER-RUN -- (Tape Function) -- Bit 6 set to 1 indicates that the data transfers from the iSBC 215G board did not keep up with the tape drive. This is not an error condition.

Bit 7 -- SEEK ERROR -- (Disk Function) -- Bit 7 set to 1 indicates that the read/write heads were not positioned to the correct track during the seek (implicit or explicit) function. When this error is detected, the internal firmware automatically commands a re-calibrate procedure and will initiate a re-seek unless retries are inhibited. Bit 7 is always set to Ø when tape functions are executed.

3.9 INTERRUPTS

The iSBC 215G board generates interrupts to alert the host CPU of significant changes in mass storage system status by asserting any of the eight MULTIBUS interrupt lines (INTØ* through INT7*). The iSBC 215G board ordinarily posts interrupts to the host CPU for three conditions:

1. Operation complete.

2. Seek complete.

3. Medium change.

The interrupt on operation-complete can be disabled by entering a 1 in bit Ø of the modifier word in I/O parameter block bytes 12 and 13. The seek-complete and medium-change interrupts cannot be disabled. Once an interrupt is asserted, it can be cancelled by a clear-I/O command from the host CPU to the board (ØØH to the wake-up port), a power-on reset, or assertion of the MULTIBUS INIT* signal.

Interrupt priority level selection in the range of Ø through 7 is done through jumper stake pin connections on the board. Two pins must be connected by wire wrapping to select the priority. (Refer to the description of interrupt priority level selection in Chapter 2.)

3.10 iSBX™ BUS EXPANSION

Connectors J3 and J4 on the iSBC 215G board allow access to the iSBX bus (refer to the Intel MULTIBUS Handbook for detailed information). The iSBX bus includes 16 data lines and 3 address lines, providing a total of eight 16-bit I/O ports per connector. Using both J3 and J4, the iSBC 215G board can communicate through the iSBX bus with as many as 16 separate peripheral ports.

The iSBX 218A Flexible Diskette Controller Board connects to iSBX connector J4 and allows communication with as many as four flexible-disk drives; the iSBX 217B/C Magnetic Cartridge Tape Interface Board connects to iSBX connector J3 and allows communication with as many as four QIC-Ø2 1/4-inch tape drives. In addition, users can design I/O controller devices that interface with the iSBX bus and use the 8Ø89 I/O processor (IOP) to control data transfer.

Two methods are available to control the transfer of data between the iSBC 215G board and a device connected to the iSBX interface:

1. Commands from the iSBC 215G board ROM-based I/O program.

2. User written I/O programs.

Both the iSBX 218A Flexible Diskette Controller Board and the iSBX 217B/C Magnetic Cartridge Tape Interface Board use the ROM-based I/O program to control data transfers to and from the flexible-disk drives and QIC-Ø2 cartridge-tape drives, respectively. The following paragraphs describe data transfer between the iSBC 215G board and a user-designed I/O controller connected to the iSBX bus, using either the ROM-based I/O program or a user-written I/O driver program.

3.10.1 FIRMWARE DRIVERS

As described at the beginning of this chapter, the iSBC 215G board includes a ROM-based I/O transfer program that is designed to control hard-disk drives through the on-board drive interface, or flexible-disk and/or QIC-Ø2 cartridge-tape drives through an iSBX 218A board and/or iSBX 217B/C board attached to iSBX connector J4 and/or J3. The I/O-transfer-through-iSBX command in this program can also be used for general data transfer between the host system memory and a user-designed I/O controller connected to the iSBX bus.

The I/O-transfer-through-iSBX command allows transfer of data between the host memory and the iSBX bus in the same manner as with the write-data or read-data commands. With this command, however, the user must provide the necessary interface hardware between the iSBX connector(s) and the I/O device with which the iSBC 215G board is to communicate. This interface can be very simple, involving data buffers and limited handshaking capability, or as sophisticated as the disk-drive interface circuitry used in the iSBX 218A and iSBC 215G boards. The complexity of the interface will depend on the type of I/O device and the desired data transfer rate.

3.10.2 USER-PROVIDED DRIVERS

A second method of initiating and controlling data transfer between the host CPU and the iSBX interface is through a user-designed driver program written in 8Ø89 IOP assembler code. This method is more difficult to implement, but also more flexible. Such programs can be executed either from host memory or from the iSBC 215G board RAM. In writing a program in 8Ø89 IOP assembler code, reference to the 8Ø89 Assembler User's Guide and the 8Ø86 Family User's Manual is essential. The IOP offers a number of techniques for implementing handshaking with the iSBX bus, the use of wait states, and DMA transfers of whole blocks of data. These and other interfacing techniques are described in the user's guide.

There are two groups of interface control lines between the IOP and iSBX bus. The first group includes handshake and control lines; the second group includes program lines. Table 3-9 lists the first group of lines. The IOP uses these lines directly to control data transfer through the iSBX bus. The second group of lines is used for control and status. The IOP accesses these lines through a read to memory-mapped I/O address 8ØØØH for connector J3 and 8ØØ8H for connector J4. Table 3-1Ø lists these lines, pin assignments, and bit assignments.

Table 3-9. IOP Handshake and Control Lines on iSBX™ Bus

J3 or J4 Pin	Description	iSBX™ Bus Mnemonic
34	Request DMA Transfer	MDRQT
32	Acknowledge DMA Transfer	MDACK*
16	Initiate Wait State	MWAIT*
6	MULTIBUS Clock	MCLK
15	I/O Read	IORD*
13	I/O Write	IOWRT*
26	Terminate DMA Activity	TDMA

Table 3-1Ø. Control and Status Lines on iSBX™ Interface

Connector J3	Address 8ØØØH	Connector J4	Address 8ØØ8H	Pin No.	Description	iSBX™ Bus Mnemonic
OPØØ	Bit B	OPØ1	Bit 3	3Ø	Option Ø	OPTØ
OP1Ø	Bit C	OP11	Bit 4	28	Option 1	OPT1
INTRØØ	Bit 9	INTRØ1	Bit 1	14	Interrupt Ø	MINTRØ
INTR1Ø	Bit A	INTR11	Bit 2	12	Interrupt 1	MINTR1
MØPST*	Bit 8	M1PST*	Bit Ø	8	iSBX Board Present	MPST*

Jumpers can be connected on the iSBC 215G board to allow the IOP to also write onto the option lines (as shown in Table 3-11). The option lines on only one of the interface connectors may be driven at a time. To drive the lines, the IOP writes to memory mapped I/O port 8Ø18H. Bit 1 drives OPØØ and OPØ1, but not both at one time; bit 2 drives OP1Ø and OP11, but not both at one time. All other bit positions in the data word must be set to Ø when driving the option lines.

Table 3-11. Jumper Connections Allowing Option Lines to be Driven

Line	iSBX™ Connector	Mnemonic	Jumper Connection
OPØØ	J3	OPØ	W11-1 -- 2
OP11	J4	OPØ	W11-1 -- 3
OP1Ø	J3	OP1	W12-1 -- 2
OP11	J4	OP1	W12-1 -- 3

NOTE

If an iSBX controller is not installed on the iSBC 215G board, or if an iSBX controller that has been installed on a particular iSBX connector does not drive its respective terminate-DMA-activity line (J3-26, J4-26), the corresponding connector jumper (W3-1 -- 2 or W4-1 -- 2) must be installed.

3.10.2.1 Random-Access Memory Map

The iSBC 215G board RAM is used for a variety of purposes. Thus, only a portion of it is available for storage of an iSBX bus I/O program and its parameters. The available RAM space is shown in Table 3-12. Note that enough space has been reserved in the data buffer to store an entire 1Ø24 byte disk sector of data. If the sectors are to be smaller or if, for some other reason, less data buffer space is required, some of this space can be used for program storage. If the devices connected to the iSBX bus do not require data from other devices, all of data buffer space can be used.

Table 3-12. On-Board Program and Parameter Storage RAM Space

Description	Address Range (Hexadecimal)
Data Buffer (Note)	4ØØØ through 44ØF
Program Storage	441Ø through 45FF
	46CØ through 46FF
Scratch Pad (Note)	46ØØ through 46BF
Pointers, Error Status, and Save Area	47ØØ through 47FF

Note: These areas may be modified by iSBC 215G board command use. If data in the data buffer need not be saved, the space may be used as a scratch-pad area between transfers.

3.10.2.2 Execution From On-Board RAM

Executing the program from on-board RAM presents space limitations, but allows data transfers to be performed at the IOP highest program execution speed. To overcome some of the RAM space problems, the program can be divided into shorter routines that are stored in the host memory and read into RAM as needed. Separate routines might thus be written for disk formatting, checking status, writing, and reading. The iSBX-execute command allows an I/O transfer routine or program that is stored in iSBC 215G board RAM to be started from a host program.

3.10.2.3 Execution From System Memory

Executing the program from host memory is inherently slower than executing the program from on-board RAM, because it requires constant access to the MULTIBUS interface. This method, however, allows the size of the program to be virtually unlimited. The procedure for executing a program from host memory is much the same as for executing a program stored in local memory. To execute a program, the user must:

1. Establish I/O communications blocks in host system memory.

2. Set the wake-up address jumpers on the iSBC 215G board for the address of the first byte of the wake-up block.

3. Program the host CPU to initiate program execution by writing Ø1H to the wake-up I/O port.

There are two important differences in the set up of the I/O communications blocks when executing I/O programs from host system memory. These are:

1. Byte Ø of the channel control block must be set to Ø3H to indicate to the iSBC 215G board that the I/O program is located in host memory.

2. The controller invocation block becomes the I/O parameter block. Refer to the 8Ø86 Family User's Manual for detailed information on setting up an I/O parameter block when the I/O program is to be executed from host system memory.

3.10.2.4 Program Execution

When loading and executing a user written I/O transfer program or routine, the following procedure is used:

1. Load the program or routine into RAM using the buffer-I/O command from the iSBC 215G board firmware.

2. Execute the execute-iSBX-I/O command to start the program. Note that the general address pointer in the I/O parameter block for this command must point to the address of the start of the

program in on-board RAM (see Figure 3-2Ø). Also, upon entering the program, the following IOP registers are defined as:

GA: 7EØØH Scratch Pad Stack
IX: Ø to 3 Unit Number

Exit from the program must always be to ROM location ØØC5H, and the IOP BC register must be set to FFH.

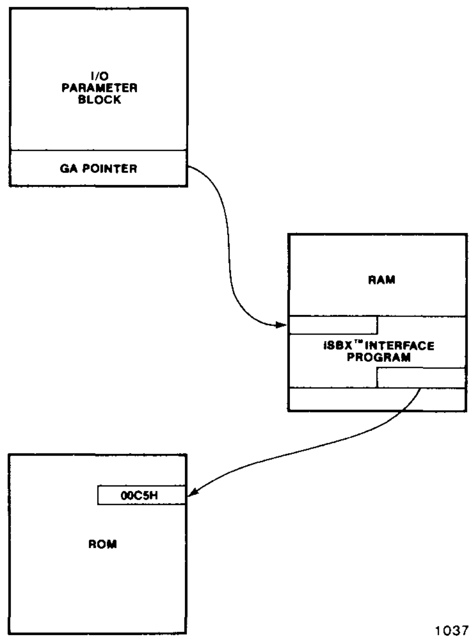


Figure 3-2Ø. Execution of iSBX™ Bus I/O Program from RAM

3.10.2.5 EXAMPLE I/O PROGRAM

Appendix B provides an example of a host CPU program to initiate data transfers between the host system memory and disk drives through the iSBC 215G board.

***

4.1 INTRODUCTION

This chapter provides a functional description of the iSBC 215G board circuit operation. The description 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 8Ø89 8/16-Bit HMOS I/O Processor and the MULTIBUS interface. Familiarity with the disk drive operation and interface specifications will also prove beneficial in understanding the board operation.

4.2 SCHEMATIC INTERPRETATION

An installation component location diagram (Figure 5-2), a block diagram (Figure 5-3), and a schematic diagram (Figure 5-4) for the iSBC 215G board are included in Chapter 5 of this manual.

The schematic is drawn to standard drafting conventions with input signals entering from the left. Signal connections between individual sheets of the schematic include a location coordinate code immediately preceding (input signals) or following (output signals) the signal mnemonic. This code defines the location of the origin or destination of the signal within the schematic diagram. 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 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 column or 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. A circle on the input of a logic element indicates that a relative low level is required to activate the element. The absence of a circle indicates that a relative high level is required to activate the element. Output levels are indicated in the same manner. Logic gating symbols are drawn according to circuit function rather than the manufacturer's definition. For example, the gate defined by the truth table in Figure 4-1 can be drawn in one of the two configurations shown, depending on its 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 mnemonic is followed by a star (for example, XACK*); when a signal is active in its high state, the star is omitted from the signal mnemonic, (for example, XACK). This convention corresponds to placing a bar over a signal mnemonic to indicate that it is active in its low state (for example, XACK).

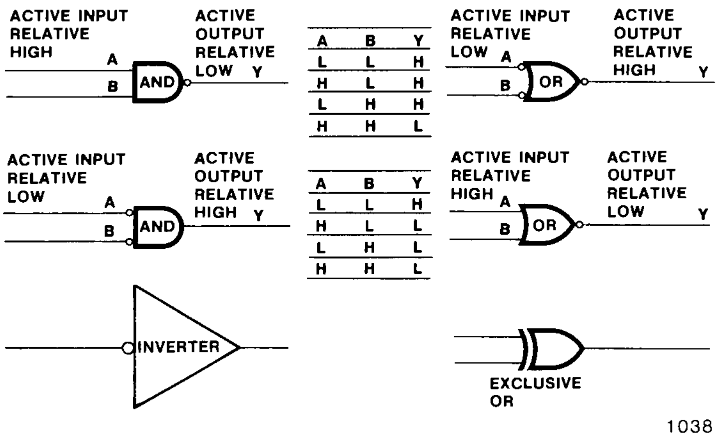


Figure 4-1. Logic Conventions

4.3 GENERAL DESCRIPTION

The function of the iSBC 215G board is to allow the host system to access any location on a specific disk or tape of a selected drive and either:

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

2. Transfer data from that location to system memory (read operation).

To accomplish this task, the board circuitry is divided into three sections (see Figure 4-2):

1. Logic that performs communications and data transfers between the host central processor unit (CPU) and the iSBC 215G board through the MULTIBUS interface.

2. Logic that performs data transfers between the iSBC 215G board and the disk drive(s) through the hard-disk interface, and between the board and the flexible-disk or cartridge-tape drive(s) through the iSBX bus interface.

3. Logic that controls both of the above functions.

As shown in Figure 4-2, the iSBC 215G board contains an Intel 8Ø89 8/16-Bit HMOS I/O Processor (IOP), which controls the data transfer process, using a program stored in on-board ROM. It receives instructions from the host CPU through four I/O communications blocks in system memory. Once the host CPU instructs the board to begin a data transfer, the IOP makes a DMA transfer (independent of the host CPU) to or from system memory.

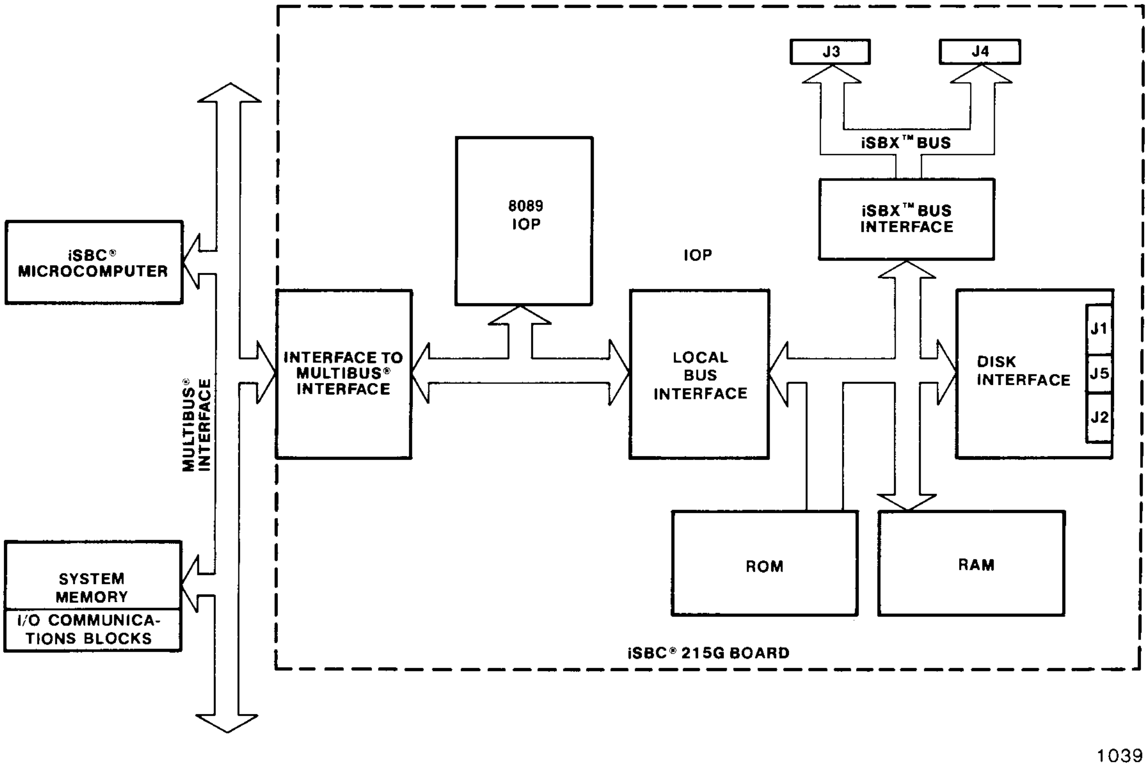


Figure 4-2. Board Simplified Block Diagram

On-board RAM space (2 K bytes) is included for intermediate storage of data and to allow on-board error checking. This RAM operates as a data buffer and allows DMA transfer between the board and host system memory, which minimizes MULTIBUS overhead and eliminates disk drive overruns.

In support of the following general description of the iSBC 215G board functional logic groups, a detailed block diagram and a schematic diagram are included in Chapter 5 (see Figures 5-3 and 5-4).

4.3.1 COMMUNICATIONS WITH HOST

As shown in the block diagram (Figure 5-3), the board includes the logic to operate within a multi-master system and contend with other masters 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 drive communication circuitry.

The MULTIBUS interface address latches and PAL U64 transmit 24-bit addresses to system memory via the MULTIBUS interface. The MULTIBUS interface data transceiver transmits data to or from system memory via the MULTIBUS interface. The data transceivers use a byte-swap technique to allow data transfer with either an 8- or 16-bit system memory. Intercommunication among board logic groups is accomplished via the board internal data bus, which is 16 bits wide.

The wake-up address comparator assigns the iSBC 215G board a host system I/O port address and sets up a communications link between the IOP and the I/O communication blocks in system memory.

4.3.2 COMMUNICATIONS WITH HARD DISK

The IOP treats the ROM, RAM, iSBX I/O ports, and disk communications portions of the board circuitry as local memory. The internal address latches transmit 16-bit addresses to local memory. The internal data transceivers transmit data either to or from local memory. (Some of the addresses in local memory provide access to local I/O ports). The address decoder decodes these addresses and generates chip-select or enable signals that control the transfer of data to and from the disk. For example, address 8Ø28H 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 serializer/de-serializer (SER/DES) performs serial-to-parallel and parallel-to-serial conversion operations 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 word between the RAM and the SER/DES. In 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. In a read operation, a 16-bit serial word is transferred from the disk through the read data receiver to the SER/DES. The SER/DES then performs a serial-to-parallel conversion and stores the resulting word in the read buffer. The write data drivers and the read data receivers are designed to generate and read the differential NRZ drive signals.

The 32-bit ID comparator determines when the selected sector on the disk is located 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 into the 32-bit ID comparator. Sector ID's from the disk are then read and compared with the selected sector ID. When the selected sector is located, 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 the end of each data field (see Figure 3-2). This ECC is used for error checking and correction of data errors. It corrects all errors in a burst of as many as 11 bits, and detects all errors in a burst of as many as 32 bits.

The gap control logic controls the spacing of data within a sector. Three programmable counters count disk clock pulses to provide timing for the gap control logic. Counter programmability allows disk formatting for a number of different record sizes and gap lengths.

The disk control logic transmits disk control information to the drives through control line drivers. The input control logic receives status information from the disk drive units and controls the sequencing of the board read and write operations.

4.3.3 COMMUNICATIONS VIA iSBX™ BUS

The iSBX bus interface provides the capability to connect Intel iSBX MULTIMODULE devices to the iSBC 215G board in order to control other peripheral devices such as flexible-disk or QIC-Ø2 cartridge-tape drives. For detailed information on the iSBX bus, refer to the Intel MULTIBUS Handbook.

4.4 DETAILED DESCRIPTION

The detailed functional description of the iSBC 215G board circuitry is divided into three sections: board-to-host communications, board-to-drive communications, and local memory organization.

4.4.1 BOARD/HOST COMMUNICATIONS

The following paragraphs provide a detailed functional description of the iSBC 215G board logic that communicates with the host CPU through the MULTIBUS interface.

4.4.1.1 MULTIBUS® Interface Signals

The IOP communicates with the host CPU and the system memory through the MULTIBUS interface. The MULTIBUS interface signal description and pin configurations are included in Chapter 2. For a detailed description of the MULTIBUS interface operation, refer to the Intel MULTIBUS Handbook.

4.4.1.2 I/O Processor

The 8Ø89 I/O Processor (IOP), (U84, 4X4), is a microprocessor device that is 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 without host CPU action allows it to carry out most system memory-to-drive or drive-to-memory transfers of data simultaneously with other host CPU operations. For detailed information on the 8Ø89 8/16-Bit HMOS I/O Processor, refer to the Intel Microsystem Components Handbook, Volume II.

A number of IOP control lines are important to the board design. The RST* line (4D1), when driven low, resets the IOP to the beginning of its internal firmware control program, and resets the interrupt latch and the read/write logic. Channel attention line CA (4B4) allows the host CPU to gain the attention of the IOP. On the first channel attention following a reset, the IOP fetches the contents of address FFFF6H and begins an internal initialization procedure. On subsequent channel attentions, the IOP reads the I/O communications blocks in system memory for further instructions.

The bus interface unit (BIU) in the IOP controls the board internal data bus cycles, transferring instructions and data between the IOP and external memory or the disk. Every bus access is associated with a register 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 SØ*, S1*, and S2*. The 8288 Bus Controller decodes these lines and provides signals that selectively enable one bus or the other.

Although the IOP is a 16-bit processor, it is capable of making both single-byte fetches (8-bit system memory) or two-byte fetches (16-bit system memory). The address Ø line, IADR-Ø (5B7), controls the byte swapping facility of the bus controller when communicating with an 8-bit system memory.

The clock circuit consists of an 8284A Clock Generator/Driver (U55, 4C6), and a 15-MHz crystal. The clock generator/driver divides the crystal output by three to produce the 5-MHz clock signal necessary to drive the IOP, and also produces an IOP reset signal (RST), which is used at power-up, after an initialization, or after a board reset. In addition to the reset signal, the clock/driver also produces a synchronized ready input (RDY) to the IOP. A high on the RDY line from the addressed device (XACK* from external memory or the iSBX interface, or RDY from the on-board read/write port), indicates that the memory or read/write port has accepted data during a write operation or that data are ready to be read during a read operation.

The 8289 Bus Arbiter, (U9Ø, 3D6), controls the IOP access to the MULTIBUS interface (see Figure 4-3). The bus arbiter monitors the IOP status lines (SØ*, S1*, and S2*). When the lines indicate that the IOP does not presently control the bus, the bus arbiter activates a bus request (BREQ*). The low BREQ* is transmitted to the bus priority resolving circuitry in the host CPU, which returns a low on bus priority in line BPRN*, giving the IOP access to the MULTIBUS interface, and the bus arbiter activates its busy signal (BUSY*), indicating to the other system masters that the MULTIBUS interface is in use. The bus arbiter then activates the address enable signal (AEN*), which is transmitted to the 8288 Bus Controller (U91, 3C4), to enable its command outputs; to the clock generator/driver (U55, 4C6), to enable its bus ready logic; and to the system address latches (U81, U82, and U83, 4X2), to allow an address to be gated to the MULTIBUS interface. Jumper stake pins W18-1, 2, and 3 allow the user to select the any-request option. Jumper W18-1 -- 2 installation causes the board to relinquish control of the MULTIBUS interface following a request from a higher priority device only. Jumper W18-1 -- 3 installation causes the board to relinquish control of the MULTIBUS interface following a request from any device of either higher or lower priority.

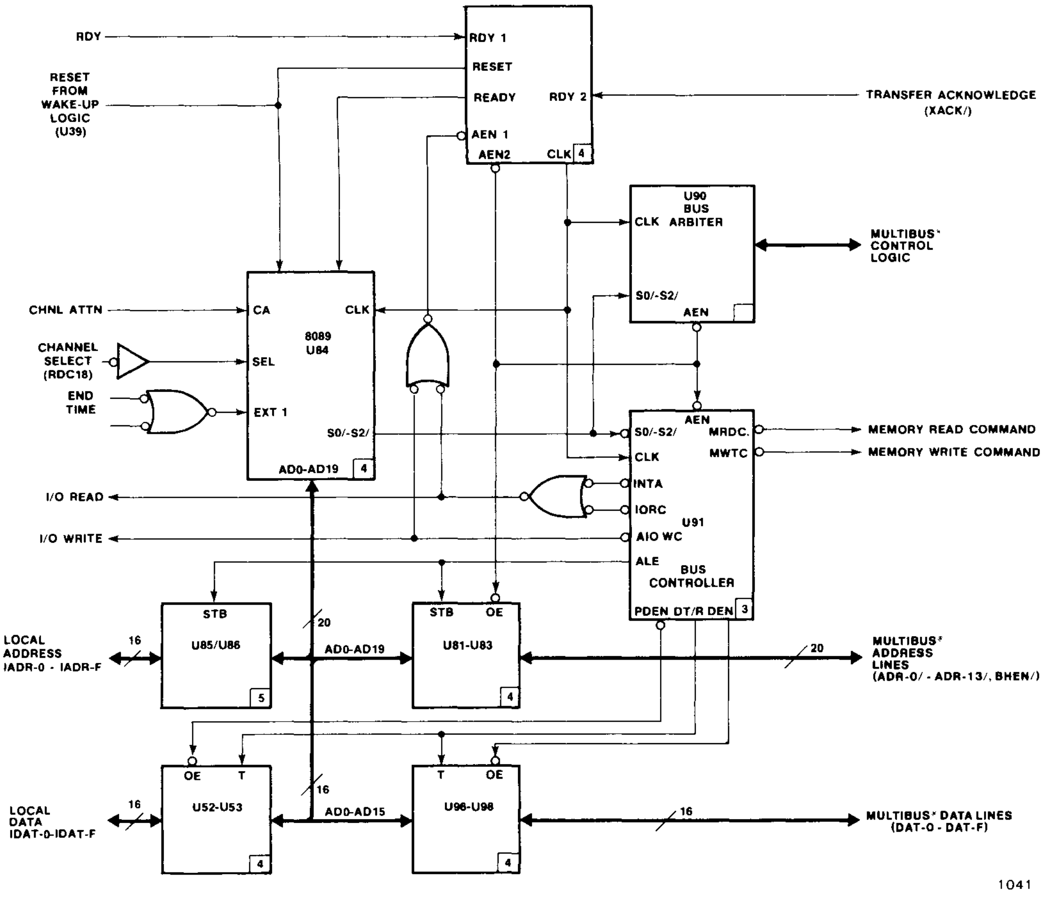


Figure 4-3. Bus Arbiter and Controller Logic

4.4.1.3 Bus Controller

The 8288 Bus Controller (U91, 3C4), decodes the status line outputs (SØ*, S1*, and S2*) from the IOP and generates the appropriate bus cycle signal. Table 4-1 lists the different signals generated for each configuration of the IOP status lines.

Table 4-1. IOP Status Line Decodes

Status Input			CPU Cycle	Bus Controller Command
SØ*	S1*	S2*		
Ø	Ø	Ø	Instruction Fetch, Local	INTA*
Ø	Ø	1	Read Memory, Local	IORC*
Ø	1	Ø	Write Memory, Local	AIOWC*
Ø	1	1	Halt	None
1	Ø	Ø	Instruction Fetch, System	MRDC*
1	Ø	1	Read Memory, System	MRDC*
1	1	Ø	Write Memory, System	MWTC*
1	1	1	Passive	None

These bus cycle signals can be divided into two groups: those that allow the IOP to access system memory (MWTC* and MRDC*) and those that allow the IOP to access local (internal) memory (I-AIOWC* and IORC*). The IOP uses the I/O read (IORC*) and I/O write (I-AIOWC*) signals to read information from the local ROM (U87 and U88, 6X7), or to read from or write to the local RAM (U99 through U1Ø2, 6X4). The IOP also uses I-IORC* and I-AIOWC* to enable the read and write function decoders (U35 and U36, 5B2, and 5A2).

The bus controller also generates a group of signals that control address and data flow throughout the iSBC 215G board. The address latch enable line (ALE) is used to strobe addresses from the IOP into both the system address latches (U81 through U83, 4X2), and the local address latches (U85 through U86, 5X7).

Data transmit/receive (DT/R), data enable (DEN), and peripheral data enable (PDEN*) signals control the data flow through the iSBC 215G board. The DT/R signal controls the direction of data transmission through the MULTIBUS interface and local transceivers. If DT/R is high, data are transmitted either to the MULTIBUS interface through transceivers U96, U97, and U98 (4X7) or to the local bus through transceivers U52 and U53 (4X6). If DT/R is low, the data transfer is in the opposite direction, into the IOP through one of the two sets of transceivers. The DEN and PDEN* signals control the selection of the transceivers. If DEN is high, MULTIBUS interface transceivers U96, U97, and U98 are enabled; if PDEN* is low (indicating a peripheral cycle), local transceivers U52 and U53 are enabled.

4.4.1.3.1 MULTIBUS® DATA TRANSFER. The iSBC 215G board has three sets of MULTIBUS interface data transceivers: low-byte transceiver U97, which buffers DAT-Ø* through DAT-7*; high-byte transceiver U96, which buffers DAT-8* through DAT-F*; and swap-byte transceiver U98, which switches the data from DAT-Ø* through DAT-7* on the MULTIBUS interface to high-byte data bus lines AD8 through AD15 on the iSBC 215G board (see Figure 4-4). This byte-swap is performed only when the iSBC 215G board interfaces with a 16-bit system memory in byte mode. In such an instance, every odd address read from system memory is transmitted to the high-byte data lines of the board. The procedure is reversed when writing to an 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 and is derived from ADRO*; and ENBL SWAP BYTE* (5C1), which controls the swap byte transceiver. Table 4-2 shows when each of the control signals is active.

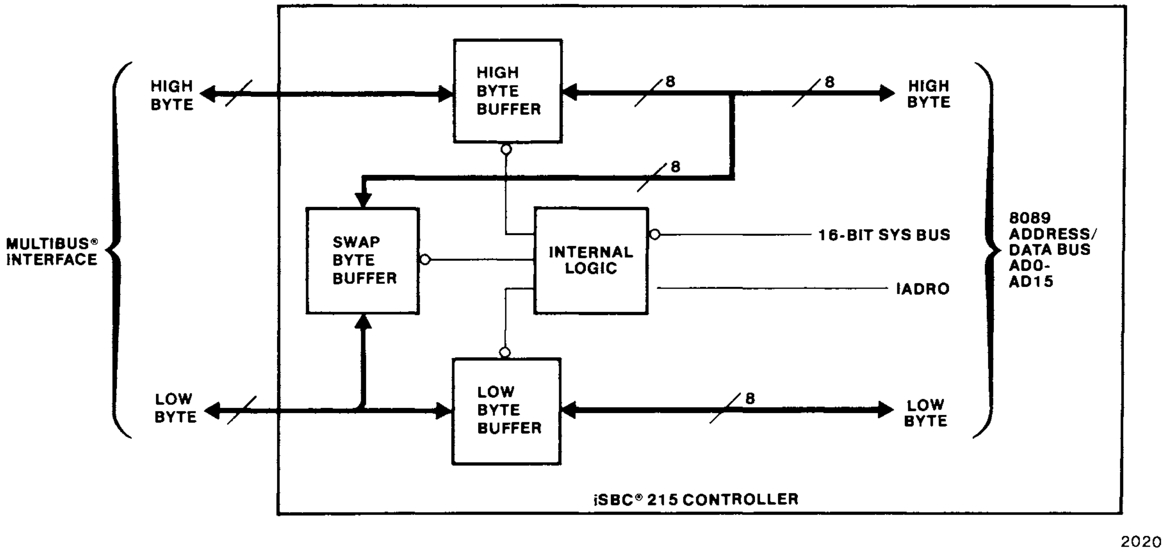


Figure 4-4. Data Transfer from MULTIBUS® to iSBC® 215G Board

Table 4-2. Data Transfer Signal Relationships

Signal	8-Bit System Memory		16-Bit System Memory	
	I/ADRØ LOW	I/ADRØ HIGH	I/ADRØ HIGH	I/ADRØ LOW
ENBL LO BYTE*	L	H	NA	L
ENBL SWAP BYTE*	H	L	NA	H
ENBLE HI BYTE*	H	H	NA	L

Note: NA denotes not applicable.

4.4.1.3.2 INITIALIZATION. Before data can be transferred between system memory and the iSBC 215G board, the board must be initialized. The initialization procedure, which is described in Chapter 3, involves:

1. Resetting the IOP.
2. Clearing the reset.
3. Establishing a communication link between the IOP and the I/O communications blocks in system memory.
4. Reading the drive parameters from system memory to the iSBC 215G on-board RAM.

The following paragraphs describe the hardware operations that take place during this initialization procedure. (See Figure 4-5.)

4.4.1.3.2.1 Wake-Up Address Comparator. For the purpose of resetting the iSBC 215G board, clearing the reset, or getting the attention of the IOP (driving the CA signal true), the host CPU addresses the board 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 (the wake-up I/O port). Table 4-3 shows the three possible commands. The user determines the address of the I/O port through which board/CPU communications are to take place (the wake-up address) and installs the appropriate jumpers on the iSBC 215G board. When the host CPU issues a write command (IOWC*) to the wake-up address in system I/O space, the wake-up address comparator (U77 through U8Ø, 2X5) compares the address with the jumper configuration. If the address and configuration agree, the WAKEUP* signal is driven low, enabling the board to decode the command on the MULTIBUS interface data lines and determine the action to be taken.

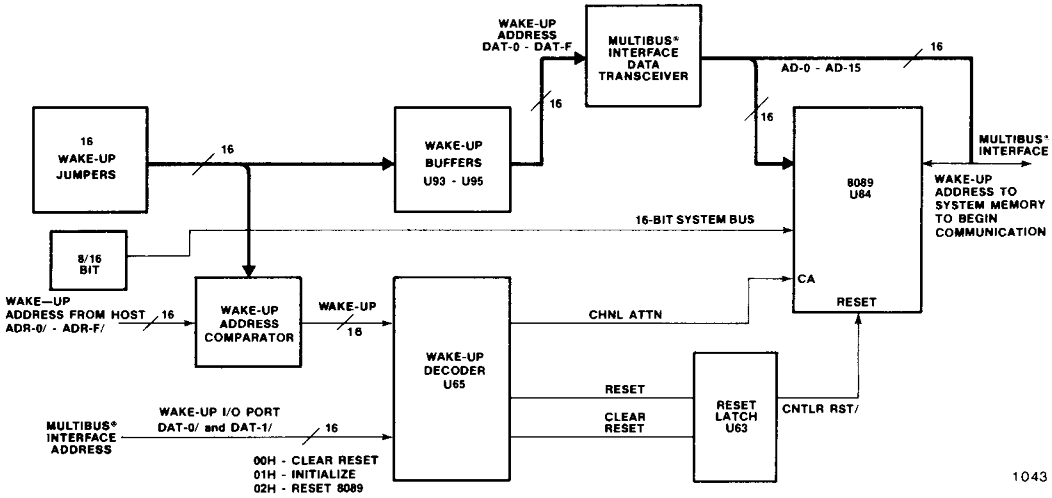


Figure 4-5. Wake-Up Address Logic

Table 4-3. Host Wake-Up Commands

Command	Description
ØØH	Clear interrupt and clear reset.
Ø1H	Channel attention (start IOP operations).
Ø2H	Reset IOP.

The host CPU may use 8- or 16-bit I/O port addressing. A user-installed jumper indicates to the board the type of addressing that is being used. When jumper W3Ø-2 -- 19 is not installed, (8-bit I/O address), pin 9 of U75 is held high, creating a "don't care" status for the outputs of high-byte wake-up address comparators U77 and U78.

As it is described in Chapter 3, the iSBC 215G board also uses the configuration of the wake-up address jumpers to calculate the address of the first byte of the wake-up block, which is the first I/O communications block in system memory.

4.4.1.3.2.2 Reset and Clear. The first operation that must be performed during the initialization of the iSBC 215G board is the IOP reset. To reset the IOP, the host CPU writes Ø2H to the wake-up address. The WAKE-UP* line becomes low and gates the Ø2H (DAT-Ø* high and DAT-1* low) to the wake-up decoder (U65, 3B7), producing a low on the controller reset (CNTLR RST*) line. A low CNTLR RST* signal resets the IOP (4X4), resets read/write control logic IC U42 (8B1 through 5), clears control register U3 (12B5), and also sets 24-bit addressing PAL U64 (9B3) to its initialized state. When jumper W36-1 -- 2 is not installed, all I/O communication blocks are in the first page of system memory; when jumper W36-1 -- 2 is installed, all I/O communication blocks are in the last page. Once the board has been reset, the host CPU writes ØØH (clear interrupt) to the wake-up address, which clears the reset. Wake-up decoder U65 decodes the highs on DAT-Ø* and DAT-1* to drive the CNTLR RST* line high.

4.4.1.3.2.3 I/O Communications Blocks Links. Following a power-up event or a software reset (Ø2H written to the wake-up I/O port), the link between the iSBC 215G board and the I/O communications blocks in system memory must be established. To establish this link, a clear reset (ØØH) is written to the wake-up I/O port followed by a channel attention (Ø1H). The Ø1H is gated to wake-up decoder U65, producing a high on the channel attention (CHNL ATTN) line, which, in turn, drives the CA input to the IOP (4C4) high.

Since this is the first channel attention following reset, the IOP starts an internal initialization process. The first step of this process is to do an on-board fetch from address FFFF6H. The board actually gains control of the bus and this address is transmitted on the IOP address/data lines (ADØ-AD15) to latches U85 and U86 (5B7). Gates U66, U72, and U76 (5D4) decode the output of these latches. The output of U76 enables U89 (5D3), gating the status configured by system data bus width jumper W3Ø-1 -- 2Ø through data bit Ø line (DAT-Ø*) to the IOP. (Jumper W3Ø-1 -- 2Ø installed indicates that the host memory system supports 16-bit data transfers, jumper W3Ø-1 --2Ø not installed indicates 8-bit data transfers.) Inverter U89 also generates the transfer acknowledge signal (XACK*), which is sent to the IOP (through the clock generator/driver), indicating that the operation has been completed. After determining the width of the system bus (8- or 16-bits), the IOP also performs on-board fetches from the addresses shown in Figure 4-6 as part of the initialization sequence. The XACK signal is generated after each fetch. (Thus, although it appears to the IOP that it is reading from the MULTIBUS interface, the read operation is from the on-board bus.)

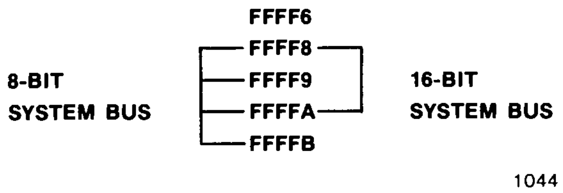


Figure 4-6. Initialization Sequence Address Fetches

Fetching addresses FFFF8/9H gates Ø's into the IOP. Fetching addresses FFFFA/BH causes the GATE SWS* line (5C1) to become low. The GATE SWS* signal gates the settings of the wake-up address jumpers through buffers U93, U94, and U95 (2X3) and into the IOP, which multiplies the configuration settings by 2^4 to determine the 2Ø-bit address of the wake-up block. The IOP then uses this address to fetch the channel control block address and establish a link with the other I/O communications blocks. On subsequent channel attention operations (when the host CPU writes Ø1H to the wake-up I/O port), the IOP skips the wake-up block and proceeds directly to the channel control block, which had been stored previously in an internal IOP non-programmable register. The IOP uses the channel control block to obtain the starting address of the board ROM-resident I/O transfer program (also called the channel control program). From this point on, this firmware program directs the board 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.

4.4.1.3.3 INTERRUPT PRIORITY. Jumper stake pins W19-C and W19-Ø through W19-7 (3B2) allow the user to select the controller interrupt priority with respect to other system peripherals. To issue an interrupt to the host CPU, the IOP writes Ø1ØØH to local I/O port 8Ø1ØH. This generates a high on data line BDAT-8 and a low on write decoder line WDC1Ø*, causing the interrupt latch (U56, 3B5) output to be high and the selected interrupt line to the MULTIBUS interface to be low. A ØØH written to the system I/O port wake-up address by the host CPU clears the interrupt.

4.4.2 BOARD/DRIVE COMMUNICATIONS

The following paragraphs provide a detailed functional description of the sections of the iSBC 215G board that communicate with hard-disk drives through the Winchester drive interface and with flexible-disk and QIC-Ø2 cartridge-tape drives through the iSBX bus interface (via connectors J3 and J4).

4.4.2.1 Board/Hard-Disk Drive Interface

All of the signals that are transmitted between the iSBC 215G board and the hard-disk (Winchester) drives are transmitted through the control cable (J1), the read/write cable (J2), or the ANSI cable (J5). (The physical configuration of these cables is described and illustrated in Chapter 2.) All signals transmitted between the board and the drives (except the read, write, and clock signals) are TTL level. The read, write, and clock signals are transmitted as differential signals.

The interface signals supported by the iSBC 215G board are described in the following paragraphs. Each of the drive interfaces uses the available lines in a unique manner. For the specific use of the lines, refer to Figures 2-1 through 2-5 and the specific drive user's manual.

4.4.2.1.1 CONTROL SIGNALS. Control and status information are exchanged between the iSBC 215G board and the drive through the control cable (J1 for non-ANSI interfaces; J5 for ANSI interfaces). Output signals are defined as those signals that the board transmits; input signals are defined as those that the board receives. The control cable is connected from J1 (or J5) on the iSBC 215G board to the first drive and as many as three subsequent drives in a daisy-chain fashion as shown in Figures 2-13 through 2-17. The functions of the 37 control cable lines can be divided into five classes:

1. Device Select (Output)
2. Head Select (Output)
3. General-Purpose Data Bus (Bidirectional)
4. Command Data (Output)
5. Status Data (Input)

Table 2-13 describes the function of the signals in each of these classes as transmitted through the control cable.

4.4.2.1.2 READ/WRITE SIGNALS. Read data, write data, clocks, and two status lines constitute the information exchanged via the read/write cable. Output signals are defined as those signals that the iSBC 215G board transmits to the disk drives; input signals are defined as those that the board receives. For multiple Shugart SA4ØØØ drives, the read/write cables are connected from the iSBC 215G board to the disk drives in radial fashion; that is, one cable from the board to each drive. Connector J2 provides read, write, and clock signals for two drives; for example, RDØ (+ and -), and RD1 (+ and -). One of these signals selects physical address Ø; the other selects physical address 1. For SA1ØØØ drives, only the signals associated with physical address Ø are used. These signals are then daisy-chained between drives, allowing the board to communicate with as many as four drives. Chapter 2 describes the cabling requirements and the physical configuration of the cables for the various drive types supported by the iSBC 215G board. Table 2-14 describes the function of each of the signals transmitted through the read/write cable. Note that the read, write, and clock signals are differential signals, requiring two lines in the cable; the status lines are TTL-level signals.

4.4.2.2 Board/Flexible-Disk and QIC-02 Cartridge-Tape Drive Interface

All signal and control lines transmitted between the iSBC 215G board and the flexible-disk and QIC-Ø2 cartridge-tape drives via the iSBX bus are transmitted through connectors J3 and J4. These lines are described only in general terms in this manual and only as the lines pertain to the remainder of the description of the board interface with the storage drives. For more detailed information on these lines, refer to the Intel MULTIBUS Handbook.

It should be noted that the iSBC 215G board does not support any parallel-to-serial or serial-to-parallel conversion of data for transmission through the iSBX connectors. It interfaces with any device connected to these connectors through an 8- or 16-bit parallel bus and a number of control and handshake lines. The iSBX interface thus resembles the read/write port, made up of the write buffer and the read buffer, that is used in the iSBC 215G board interface to the hard-disk drives.

The schematic diagram mnemonics for the signal and control lines (from the iSBC 215G board) that are connected to iSBX connectors J3 and J4 often differ from the respective line mnemonic from the iSBX bus specifications. Table 2-11 lists both the iSBX bus mnemonic and the iSBC 215G board mnemonic for each signal, in the iSBX bus, that the board supports.

4.4.2.3 Interface Timing

The following paragraphs provide a detailed description of the inter-circuit timing of formatting a disk, writing to a disk, or reading from a disk. The timing logic is shown on sheet 8 of the schematic diagram; the disk drive interface receivers and drivers are shown on sheets 9 through 12.

4.4.2.3.1 <u>DIRECT MEMORY ACCESS TRANSFERS</u>. In general, when the iSBC 215G board performs a read or a write function, it locates the area of the disk where the read or write is to be performed, then enters the direct memory access (DMA) mode to perform the actual transfer. In the DMA mode, the 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 board controls the transfer of data between the read/write port and the disk.

The ready line (RDY, 8D1) is used for hand shaking between the IOP and the data transfer circuitry. When RDY is low, the IOP is quiescent; when

RDY is high, the IOP performs a DMA transfer of data either from the local RAM to the write buffer (block-to-port), or from the read buffer to local RAM (port-to-block). Gates U4Ø, U41, and U12 (8D3) control the RDY line.

To perform a write or a read, the IOP executes firmware routines 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 TIME OUT line (8D8) is low; the MWAIT* line (13D1 and 8D8) is high; the R/W GATE line (8D1) is high (see Figure 4-7), U21-8 (8D3) is high (held so by the low on the ENBL XFER line, 8D1), and the R/WDC 28 line (the output of U11-11, 8D7) is low. The low on R/WDC 28 thus keeps RDY activated.

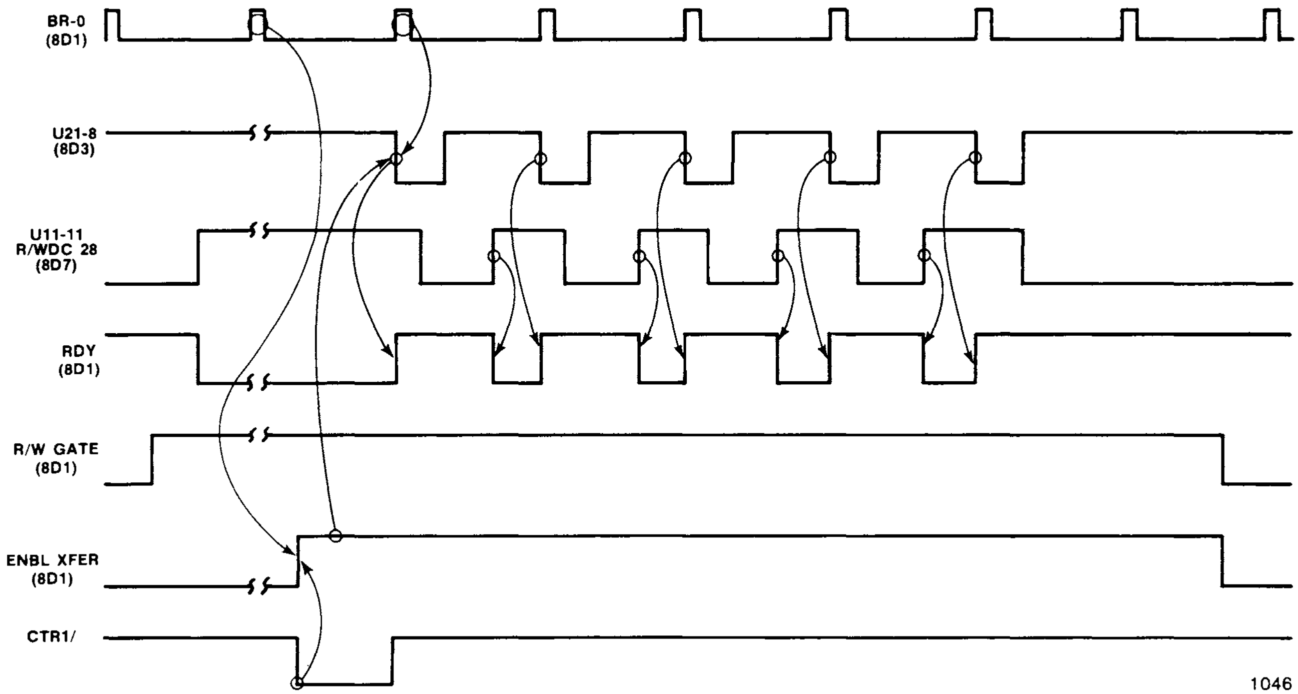


Figure 4-7. Ready Signal Timing Diagram

On its first attempt to transfer data in the DMA mode, the IOP activates either RDC 28* or WDC 28* (8D8), depending on whether a read or a write is being performed, respectively. When RDC 28* or WDC28* is activated, the R/WDC 28 line is activated, lowering RDY and switching the IOP to the quiescent (wait) state. When the board data transfer circuitry locates the area on the disk where the read or write is to begin, it activates ENBL XFER (8D1). On the next occurrence of a bit ring-Ø pulse (bit Ø of each word, BR-Ø, 8D1) following the activation of ENBL XFER, U21-8 (8D3) is activated, activating RDY. The IOP 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 U21-11, U21-8 is driven high and, on the next IOP attempt to perform a data transfer, R/WDC 28 is also driven high, lowering RDY. The data transfer does not occur and the IOP switches to the wait state.

During this time, the SER/DES either transfers the word from the write buffer to the disk or reads another word from the disk into the read buffer. Then, on the next BR-Ø pulse, RDY is again activated and the next DMA data transfer occurs. The IOP continues in this DMA mode until the R/W GATE line is lowered.

Note that two other lines have potential control over the RDY line. The TIME OUT line (8D8) allows the IOP to be activated if a sector cannot be located on a cylinder. While the drive is searching for a sector, the RDY line is held low. If, after two revolutions, the drive does not locate a sync byte, the TIME OUT line is raised. IC U41 (8D3) gates the TIME OUT signal to U12 (8D1) and activates RDY.

The MWAIT* line (8D8) is an iSBX interface control line and is derived from MWAITØ* and MWAIT1* (13D8). Signal MWAIT* exercises the same control over the RDY line as U4Ø (8D3) and can thus be used to set up a handshaking arrangement between an I/O controller connected to one of the iSBX interface connectors (J3 or J4) and the IOP. For more detailed information, refer to the Intel Microsystem Components Handbook.

4.4.2.3.2 <u>DISK FORMATTING</u>. Before the surfaces of a disk can be used for writing and reading data, the disk must be formatted. Formatting is the operation of writing all address fields, gaps, ID headers, etc. for the complete disk. The iSBC 215G board performs this operation under software control. The software routine that controls formatting allows for formatting a single track with each format command to the board until the entire disk is formatted.

Implementation of the format command is divided into two operations. During the first operation, address marks (soft-sectored disks only), gaps, and ID fields are written during a single disk revolution. During the second operation, user-supplied data are written into data fields. The second operation requires two disk revolutions, one to write the odd physical data fields (1, 3, 5, etc.), and one to write the even physical fields (Ø, 2, 4, etc.). Three disk revolutions are thus required to format a single track. The hardware execution operation described in the following paragraphs pertains to the formatting of a soft-sectored disk. The iSBC 215G board supports both soft- and hard-sectored disks.

NOTE

> A soft-sectored disk (as used in Shugart/Quantum drives) requires that an address mark be written at the beginning of each sector during the formatting operation. Hard-sectored disks (as used in Memorex and Priam drives) provide a sector pulse at the beginning of each sector. Thus, address marks need not be written.

The formatting procedure, however is essentially the same. The differences are described at the end of this section, along with the slight differences in the sector format used with Shugart/Quantum drives. When the format command is issued to the iSBC 215G board, the IOP performs a seek to the desired track (cylinder) to begin the format operation.

When the heads are positioned over the selected track, the IOP writes a CØH (for drive Ø), a C8H (for drive 1), a DØH (for drive 2), and a D8H (for drive 3) to I/O port 8Ø18 (decoded as WDC 18*). The activation of WDC18* enables U3 (12A5) and activates the WRT GAT-F and FORMAT lines (12B1) and the WRT GATE line (12C1). (See Figure 4-8.) The WRT GAT-F and FORMAT signals enable the board format control circuitry. With WRT XFER inactive and all Ø's applied to the SER/DES, the board then then writes all Ø's to the drive while the IOP waits for the receipt of the first INDEX* pulse (11D8).

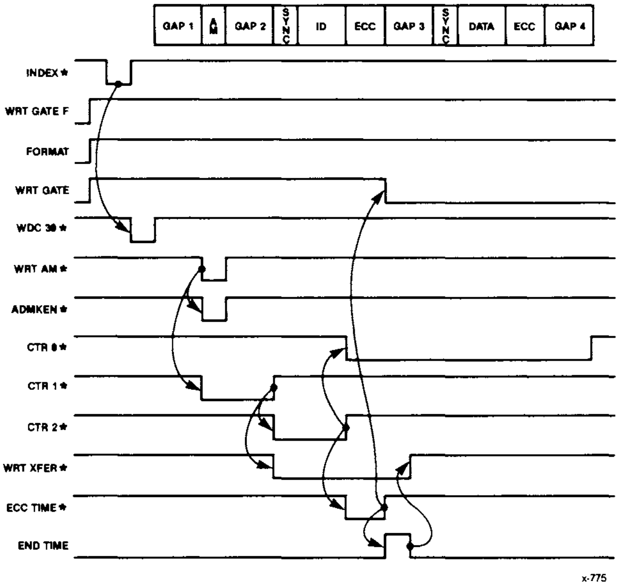


Figure 4-8. Disk Formatting Sequence Timing Diagram

The receipt of INDEX* sets latch U34 (11D6), which, in turn, sets bit F of the status register. This causes the IOP to poll (read) I/O port 8ØØØH bit F (decoded as RDC ØØ*). Upon detecting the index, the IOP writes XXXXH to I/O port 8Ø3ØH (decoded as WDC 3Ø), which triggers U63 (8B7), activating the WRT AM* line (8B1) and causing the first address mark to be written on the disk through the ADMKEN* line (12D1).

The time allowed by the IOP between the detection of index and the activation of U63 (8B7) is approximately 11 byte times, which is the time that the iSBC 215G board requires to perform a number of firmware steps in preparation for writing the first address mark and ID field (see Figure 3-2 for a pictorial representation of the track format). During this time, the IOP writes the sync byte (ØØ19H) to the write buffer (U46 and U49, 7C7 and 7D7), by writing to I/O port 8Ø28H (decoded as WDC 28*). It performs this operation in preparation for writing the ID field on the track.

The activation of WRT AM* also starts counter 1 (U69, 8A7). (The IOP preset the counters in U69 at the beginning of the format operation.) When counter 1 times out at the end of 11 byte times, it activates the WRT XFER* line through U63-7 (8C3), and starts counter 2. The activation of WRT XFER* initiates the IOP DMA mode, during which the sync byte and the sector ID are written onto the disk. Counter 2 times out at the end of the ID field, starting counter Ø and activating the ECC TIME line (8B1). During the ECC TIME, the ECC code from the ECC generator is written following the ID field. At the end of ECC TIME, the END TIME line is enabled, which lowers the WRT XFER* line and switches the IOP out of the DMA mode. After the last ID field is written, the FORMAT line is deactivated, which inhibits the writing of any additional address marks.

Counter Ø is set for a time equal to ECC+G3+DATA+G4, which the IOP sets according to the sector size selected for the drive. When counter Ø times out, it activates WRT AM* and counter 1, which begins the formatting of the second sector. This procedure is repeated until the IOP determines that the last ID field has been formatted. The IOP then begins searching for the index pulse. Upon receipt of index, the RST FRMT* line is activated, resetting WRT GAT-F and FORMAT, and inhibiting the writing of the next address mark. The IOP then continues through the format routine to the second operation, which is the writing of user-supplied data into the data fields.

For hard-sectored disks, jumper W16-1 -- 3 (8B8) is installed. The formatting of the first sector thus begins when the first SECTOR* pulse from the disk (following index) is received, rather than when WDC 3Ø* is activated. When the SECTOR* line (11B8) is activated, it activates the INDEX-SECTOR* line (11C1), which starts counter 1 counting. Formatting then continues in the same manner as with soft-sectored disks, except that the beginning of the next sector occurs at the receipt of the next SECTOR* pulse rather than at the timing out of counter Ø.

The 8-inch Shugart/Quantum drive sector format differs in two ways from that of the other drive types. In these drives, an address mark is placed before both the ID field and the data field, with no gap between the address mark and the sync byte. In addition, D9H is used for the sync byte in the data field rather than 19H. When the iSBC 215G board sync byte detector circuit (U54, U68, and U73, 7B5), detects a sync byte (19 or D9) following an address mark, and the SR-6 (7B1) line is activated (D9 only detected), and the DATA SYNC and IDNCMPRL lines are activated through latch U37 (9A6). DATA SYNC and IDNCMPRL then set bits 3 and 6, respectively, of status register U1Ø (11C5), indicating to the board the presence of a data field instead of an ID field. In Memorex, 14-inch Shugart, Pertec, and Priam drives, a data field is assumed to follow an ID field without an intervening address mark.

A second difference between the 8-inch Shugart/Quantum drive and the others is that, with the Shugart/Quantum drives, a 4EH pattern is written in the gaps rather than Ø's. Inverters U58 and U17 (8D6) and gate U19 (8D5) create the 4EH pattern. Gates U4Ø and U6Ø (8A3) apply the pattern to the SER/DES when the SHUGART and WRT GAT-F lines are activated during a format.

4.4.2.3.3 WRITE DATA TRANSFERS. The write operation is divided into two steps: read sector ID and write data. When a write is initiated, the IOP writes ØØØ6H to I/O port 8ØØØH (decoded as WDCØØ). Latch U24 (12C5) then activates the AM SEARCH*, ADMKEN*, and RD GATE* lines, which enables the drive to search for the address mark and enables the board read circuitry (see Figure 4-9).

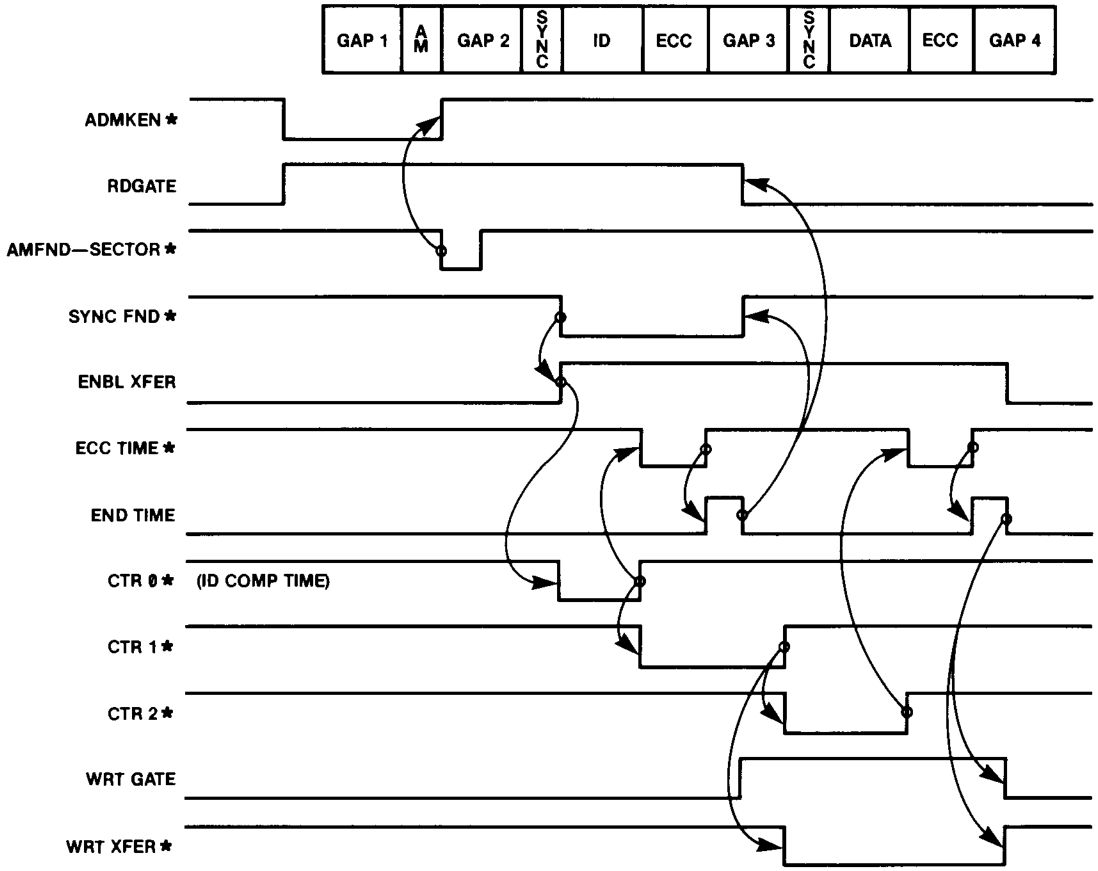


Figure 4-9. Write Data Transfer Timing Diagram

The IOP also writes to I/O ports 8Ø3ØH and 8Ø38H (decoded as WDC3Ø* and WDC38*), loading the ID of the sector to be written to into the 32-bit ID comparator logic (U2, U1, U23, and U22). Note that it has previously written to I/O port 8Ø2ØH (decoded as WDC2Ø*) to load counters Ø, 1, and 2 of U69 (8A7). When the address mark (or sector pulse) is detected, SECTOR* is activated, which activates the AMFND-SECTOR* line (11B1). The low on AMFND-SECTOR* resets U34 (8C7) and de-activates the ID FIELD line, which de-activates the ADMKEN* line and activates the ALW SYNC SRCH, initiating the search for the sync byte. Note that with the Shugart drives, the sync byte follows the address mark directly. Activating AM FND-SECTOR* thus activates ALW SYNC SRCH directly, through jumper W14-1 -- 2 (12C3).

In searching for the sync byte, serial data from the disk are read into the SER-DES. Sync byte comparator U73 and U54 (7B5) monitors the outputs of the SER-DES and drives the SYNC BYTE* line (7B1) low when 19H (the sync byte) is detected. The enabling of SYNC BYTE* enables the SYNC FND* lines (9C1), which, in turn, activates the ID comparator (U1, U2, U22, and U23, 9DX) and word clock U2Ø (8D7). The SYNC FND* signal also drives the ENBL XFER line (8C1) high, which enables the ECC generator logic (7AX) and ready latch U21 (8D4) and enables counter Ø of U69 (8A7).

The 32-bit comparator compares the ID read from the disk with the ID of the selected sector. At the end of the ID time, counter Ø times out, driving the ECC TIME* line (7A8) low and initiating the ECC compare. If the ID and the ECC are valid, bit 6 of the board status register (U1Ø, 11C5) is reset. At the end of ECC time, U42-1Ø (8B2) activates the END TIME line, which resets RD GATE. The IOP then checks bit 6 of control status register U1Ø (11C5). If the bit is inactive, the IOP continues with the write operation. If the ID or ECC is not valid (bit 6 active), the AM ENABLE and RD GATE lines are then asserted and the board searches for the next address mark.

To begin the second step of the write operation, the IOP writes Ø1H to I/O port 8ØØØH (decoded WDCØØ*) and enables the write gate (WRT GATE), through U24 (12B5), enabling the drive write circuitry. When counter Ø 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 U63-7 (8C3) is set, activating WRT XFER*, which enables write buffers U46 and U49 (7C7) and the ECC comparator logic (7AX), and raises the RDY line, indicating to the IOP that the write buffer is ready to receive data.

The IOP then enters DMA mode to write data from local RAM to the disk. The board 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. The END TIME signal then terminates the write operation.

4.4.2.3.4 <u>READ DATA TRANSFERS</u>. The read operation is divided into two steps: read sector ID and read data. Reading the sector ID is performed in the same manner as for the write operation (see Figure 4-1Ø).

When the desired sector is located, the RD GATE is raised to search for the sync byte of the data field. When SYNC FND* is activated, counter 2 is started through U61-8 (8C4) and U59 (8B6), the ECC generator is enabled, and the RDY line is activated, initiating the DMA read data transfer mode. Data are then transferred from the disk to local RAM for the duration of the count of counter 2. When counter 2 times out, ECC TIME is activated. Following ECC TIME, the END TIME line is driven high, terminating the read operation.

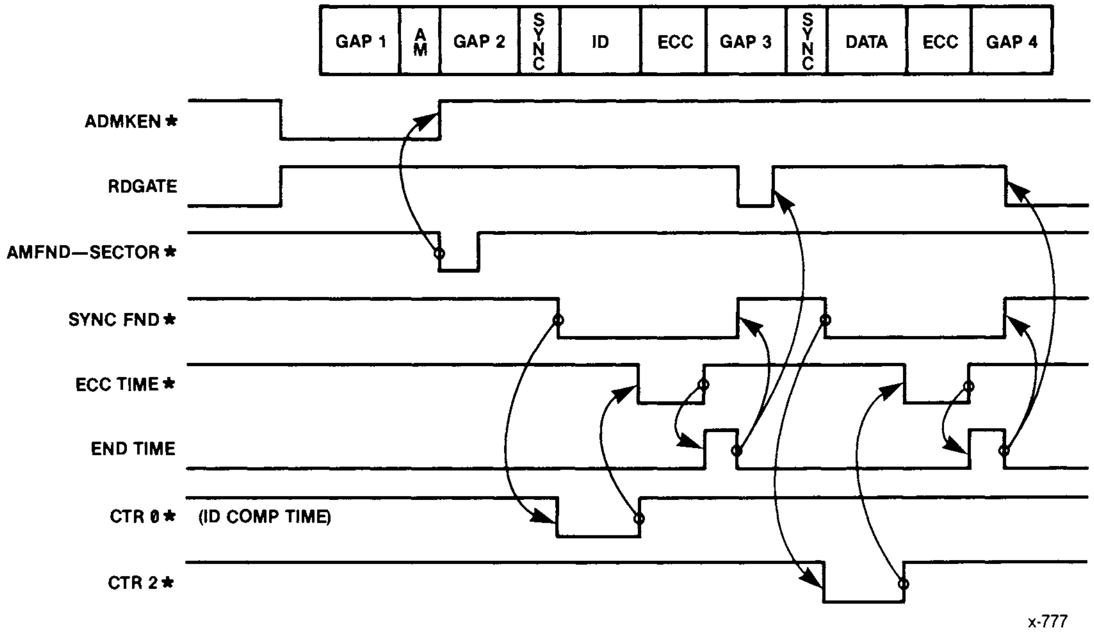


Figure 4-1Ø. Read Data Transfer Timing Diagram

4.4.2.4 Serializer/De-Serializer

The serializer/de-serializer (SER/DES) logic performs two functions: it converts parallel data words into a serial bit string to be sent to the disk drive during a write operation, and it converts a serial bit string into 16-bit words during a read operation. It consists of the write buffer (U46 and U49, 7C7), the serializer/de-serializer (U47 and U5Ø, 7C5), the read buffer (U48 and U51, 7C4), and the selector (U7Ø, 7A7).

During a write operation (WRT XFER* low), the IOP writes to I/O port address 8Ø28H. Write I/O port address decoder U35 (5A2) decodes this address and drives WDC28* low, clocking the data to be written to the disk (BDAT-Ø through BDAT-F) into write buffer U46 and U49 (7C7). A high on load-serial-register line LDSR (7C6), derived from word clock U2Ø (8C7) loads the contents of the write buffer (SR-Ø through SR-F) into the SER/DES (7C5). Read/write clock R/W CLK-B (7B8) then clocks the data bit-by-bit through the QH' output of U5Ø (7D5), and through selector U7Ø (7A7) to the WRT DATA line. The R/W CLK-A signal clocks the serial data string on WRT DATA through U18 (1ØC3) to the selected drive.

During a read operation, R/W CLK-B (1ØB1) gates the serial data string (RD DATA) from the disk drive through U18 (1ØB4) and selector U7Ø (7A7) and into the SI input of U47 (7C5), creating a 16-bit parallel word. The bit ring-Ø line (BR-Ø, 7B8), which is derived from word clock U2Ø (8C7), then clocks this word into read buffer U48 and U51 (7C4). With the read buffer loaded, the IOP initiates a read to I/O port address 8Ø28H. Read I/O port address decoder U36 (5B2) decodes this address and drives RDC 28* low, which clocks the data word from the read buffer onto internal data bus lines IDAT-Ø through IDAT-F.

4.4.2.5 Sync Byte Comparator

The sync byte comparator detects the presence of a sync byte during a read operation and synchronizes word clock U2Ø (8C7) with the data. A sync byte is written before each sector ID and each data field to indicate to the board that data to be read are forthcoming (see Figure 3-2). The sync byte value is always 19H, except for the Shugart/Quantum drives, which use D9H for data fields.

During a read operation, sync byte decoder U54 and U73 (7B5) monitors the output of the SER/DES (U47 and U5Ø, 7C5). When 19H is detected, the SYNC BYTE* signal is driven low, indicating the presence of the sync byte. The SYNC BYTE* signal and the next output of R/W CLK-B set the SYNC FND flip-flop (U57, 9C6), which activates word clock U2Ø (8C6), and activates the read/write logic (sheet 8).

4.4.2.6 Sector Identity Comparator

The 32-bit sector identity (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 flags, cylinder number, sector number, and head address.

To load the sector ID of the record being searched for into 32-bit ID comparator U1, U2, U22, and U23 (9DX), the IOP writes to I/O ports 8Ø3ØH and 8Ø38H, enabling the WDC3Ø* and WDC38* lines, respectively. These lines initiate loading 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). The R/W CLK-B signal clocks the ID information, which is stored in the ID comparator, out of U22 (pins 7 and 9) bit-by-bit. Comparator U26 (9D2) compares the serial string of bits with the sector ID from the disk drive (RD-DATA). If the two sector ID's differ, ID no-compare line ID NCMPR* is activated; if the sectors are the same, ID NCMPR* is driven high. Selector U7Ø (7A7) OR's the ID NCMPR* and the ECC NCMPR* lines. The resulting ID-ECC NCMPR* line is latched into U37 (9B6). The Q output of U37, ID NCMPR-L, is transmitted to bit 6 of status register U1Ø (11C5). The IOP then reads the contents of the status register and checks the condition of bit 6. Bit 6 being high indicates that the record read from the disk was either not the record being searched for or that it had an ECC error; conversely, bit 6 being low indicates that the ID field compared and that there was no ECC error. The IOP then reads or writes the data portion of the record.

4.4.2.7 Error Checking Code Generator

The error checking code (ECC) logic: 1) generates (during a write operation) a four-byte ECC polynomial check sum that is appended to the ID field (format write) and the data field (normal write) of a record (see Figure 3-2), and 2) re-generates (during a read operation) the ECC polynomial check sum and compares it with the ECC field read from the disk record to ensure that correct data were read from the drive.

During the write operation, serial data (either an ID field or a data field) are transmitted from the SER/DES (7C5) through selector U7Ø (7A7) and into the ECC generator through pins 1 and 2 of U1Ø3 (7A6), where the ECC polynomial check sum is generated. At the same time, a high on the WRT XFER DLYD line (7B8), transmitted through gate U68 (7B4), enables the serial data to be transmitted through U71 (7A2) and selector U7Ø (7A7) to the WRT DATA line for transmission to the disk. At ECC time (end of data field), the WRT XFER DLYD signal becomes low, inhibiting write data from being transferred through gate U68 (7B4). The ECC TIME* line then becomes low, causing the ECC polynomial check sum to be written onto the disk through U71 (7A3), U7Ø (7A7) and the WRT DATA line.

During a read operation, serial data (again either a sector ID or a data field) are read into the ECC generator through selector U7Ø (7A7) and into the SER/DES through U71 (7A3) and U7Ø. At ECC time, U71 compares the ECC polynomial from the ECC generator bit-by-bit with the ECC polynomial from the disk and transmits the difference through U7Ø to the SER/DES for storage in RAM. If the difference is Ø, the ID-ECC NCMPR* line is driven high, indicating correct data or sector ID. If the result of the comparison is not Ø, the difference, called the "error syndrome", is used by the IOP to correct errors in a sector ID or data field (if correctable).

4.4.2.8 Status Register

The status register (U1Ø and U44, 11X5; and U9, 11B3) transmit status information from the selected disk drive, the iSBX interface, and various logic within the board drive interface circuitry to other logic of the board. When the IOP issues a read status command, or checks status as an internal operation, read decode enable lines RDC ØØ* and RDC Ø8* are activated, causing the contents of status registers U1Ø, U44, and U9, respectively, to be transferred onto the internal bus (IDAT-Ø through IDAT-F). The IOP then analyzes the status information for an internal operation or communicates the status of the data transfer operation to the host CPU through system memory (controller invocation block). Table 4-4 lists the status register bits. Refer to Chapter 3 for information on the status information transmitted to the host CPU.

Table 4-4. Status Register Bits

Bit Number	Function		
	8ØØØH (Upper Byte) U44 (11D5)	8ØØØH (Lower Byte) U1Ø (11C5)	8ØØ8H (Lower Byte) U9 (11B3)
F	Index		
E	Drive Request		
D	Illegal Request		
C	Option Bit 1Ø		
B	Option Bit ØØ		
A	Interrupt 1Ø		
9	Interrupt ØØ		
8	iSBX Board on J3		
7		Time Out	Write Protected
6		ID No Compare	Track Zero
5		Bus Acknowledge	Vendor
4		Fault	Option Bit 11
3		Data Sync	Option Bit Ø1
2		Seek Complete	Interrupt 11
1		Ready	Interrupt Ø1
Ø			iSBX Board at J4
Note:	Bit numbers C, B, A, 9, and 8 and 4, 3, 2, 1, and Ø are iSBX bus lines.		

4.4.2.9 Line Drivers and Receivers

All serial data and high-speed clock signals transmitted between the iSBC 215G board and the drives use differential pair line drivers and receivers. The polarity on these lines is positive-true logic; that is, when the + side of the line is more positive than the - side of line, a positive logic 1 is being transmitted.

The board differential drivers (U16, 1ØX3) are referenced to Ø and +5 V. The board receivers that accept differential signals from other than Shugart SA1ØØØ drives (U13, 1ØX6), are also referenced to Ø and +5 V. The receivers for 8-inch Shugart SA1ØØØ drives (U15, 1ØX5) accept differential signals that are referenced to -5 and +5 V.

4.4.3 LOCAL MEMORY ORGANIZATION

As was described in the functional overview, the IOP addresses the read-only memory (ROM), random-access memory (RAM), iSBX I/O ports, and the hard-disk communications side of the board circuitry as local memory. Figure 4-11 is a map of this local memory. The following paragraphs describe the ROM, RAM, and I/O ports.

4.4.3.1 Read-Only Memory

The iSBC 215G board ROM, which contains the IOP disk control program, consists of two (8k x 8-bit) ROM devices (U87 and U88, 6X7). On any read from local memory in the range of ØØØØH to 3FFFH, chip-select decoder U65 (5B4) decodes address lines IADR-E and IADR-F and drives ROM chip-select line CSROM* low, enabling the ROM devices.

4.4.3.2 Random-Access Memory

The iSBC 215G board RAM consists of four (1k x 4-bit) RAM devices (U99 through U1Ø2, 6X4). On any read or write to local memory in the range of 4ØØØH to 47FFH, chip-select decoder U65 (5B4) drives RAM chip-select line CSRAM* low, enabling the RAM devices.

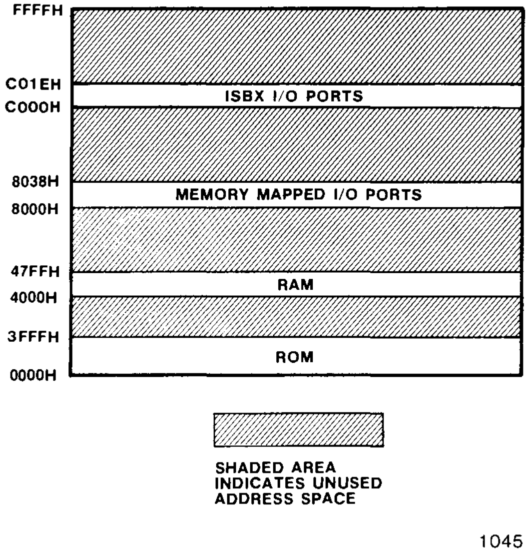


Figure 4-11. Local Memory Map

4.4.3.3 I/O Port Decoding

The IOP views the control devices in the disk control circuitry (such as ID comparators, counters, write buffer, read buffer, etc.) and the iSBX bus ports as local I/O ports, each with an address in local memory space. To enable one of the drive control devices, the IOP executes a read or a write to the device address. On any read or write to local memory in the range 8ØØØH through 8Ø38H, chip-select decoder U65 (5B4) pin 1Ø output is driven low.

When this low on pin 1Ø of U65 is accompanied by a low on I/O read line I-IORC*, read I/O port address decoder U36 (5B2) is enabled. When the low on pin 1Ø of U65 is accompanied by a low on I/O write line I-AIOWC*, write I/O port address decoder U35 (5A2) is enabled. Decoder U35 or U36 then decodes local memory address lines IADR-3 through IADR-5 to select the desired drive control device. Table 4-5 shows the address of each local I/O port and its function.

The two iSBX bus connectors on the iSBC 215G board, J3 and J4, provide access to the iSBX bus 16 data lines and 3 address lines. The two iSBX channels provide a total of sixteen 16-bit I/O ports per connector. Each of these I/O ports has an address in local memory space (see Table 4-6).

When the IOP executes a read or write to one of these ports, chip-select decoder U65-9 (5B4) activates the CSMMIO* line. Gate U3Ø (13C3) and inverter U31 (13C4) decode the CSMMIO* and IADR-4 lines to select either J3 or J4. Address lines IADR-1, IADR-2, and IADR-3 are transmitted to connectors J3 and J4, pins 11, 9, and 7, respectively (5C1), to select the I/O port on the selected connector.

Table 4-5. Local I/O Ports

	Read (U33 Enabled)		Write (U32 Enabled)	
Address	Enable Line	Function	Enable Line	Function
8ØØØH	RDCØØ*	Read status	WDCØØ*	Write control data to disk drive and enable AM SEARCH*, RDGATE, AND WRTGATE
8ØØ8H	RDCØ8	Read status	WDCØ88	Clear index and ID not-compare latches
8Ø1ØH	RDC1Ø	Read disk data bus	WDC1Ø*	Write to disk data bus
8Ø18H	RDC18*	Raise IOP Channel 2 Attention	WDC18*	Write to unit select and control register
8Ø2ØH	RDC2Ø*	Read contents of counter Ø	WDC2Ø*	Load counter Ø
8Ø22H	RDC2Ø*	Read contents of counter 1	WDC2Ø*	Load counter 1
8Ø24H	RDC2Ø*	Read contents of counter 2	WDC2Ø*	Load counter 2
8Ø26H		Not used	WDC2Ø*	Write mode word
8Ø28H	RDC28*	Read contents of read buffer	WDC28*	Write data to write buffer

Table 4-5. Local I/O Ports (continued)

	Read (U33 Enabled)		Write (U32 Enabled)	
Address	Enable Line	Function	Enable Line	Function
8ØØ3ØH	RDC3Ø*	Read vendor bits 3 and 4	WDC3Ø*	Write sector ID to high comparator, start track format operation
8Ø38H		Not used	WDC38*	Write sector ID to low comparator

Table 4-6. iSBX™ Bus I/O Port Addresses

Port	J3 Channel Ø	J3 Channel 1	J4 Channel Ø	J4 Channel 1
Ø	CØ7Ø	CØBØ	CØDØ	CØEØ
1	CØ72	CØB2	CØD2	CØE2
2	CØ74	CØB4	CØD4	CØE4
3	CØ76	CØB6	CØD6	CØE6
4	CØ78	CØB8	CØD8	CØE8
5	CØ7A	CØBA	CØDA	CØEA
6	CØ7C	CØBC	CØDC	CØEC
7	CØ7E	CØBE	CØDE	CØEE

***

5.1 INTRODUCTION

This chapter contains the various required diagrams and service and repair instructions for the iSBC 215G Winchester Disk Controller Board.

5.2 SERVICE DIAGRAMS

Figure 5-1 is the parts location diagram; Figure 5-2 shows jumper locations; Figure 5-3 is the schematic diagram. In Figure 5-3 a signal mnemonic that is followed by an asterisk or slash (for example, BHEN*) is active low; a signal mnemonic without a star is active high.

5.3 SERVICE AND REPAIR ASSISTANCE

Customers within the United States can obtain service and repair assistance by contacting the Intel Product Service Center 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 Center, you should have the following information available:

1. Date that you received the product.
2. Complete part number of the product (including the dash number).
3. Serial number of the product (usually stamped on the component side of the board).
4. Shipping and billing address.
5. Purchase order number (for billing purposes if your Intel product warranty has expired).
6. Extended warranty agreement information (if applicable).

Regional Telephone Numbers:

☎

Western Region:	6Ø2-869-4951	International:	6Ø2-869-4862
Midwest Region:	6Ø2-869-4392	TWX Number:	91Ø-951-133Ø
Eastern Region:	6Ø2-869-4Ø45		

SERVICE INFORMATION

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

Use the original factory packing material (if possible) when preparing the product for shipment to the Repair Center. If the original material is not available, wrap the product in cushioning material such as Air Cap TH-240 (manufactured by the Sealed Air Corporation, Hawthorne, NJ), enclose it in a heavy-duty corrugated shipping carton, and label it FRAGILE to ensure careful handling. Ship only to the address specified by the Service Center personnel.

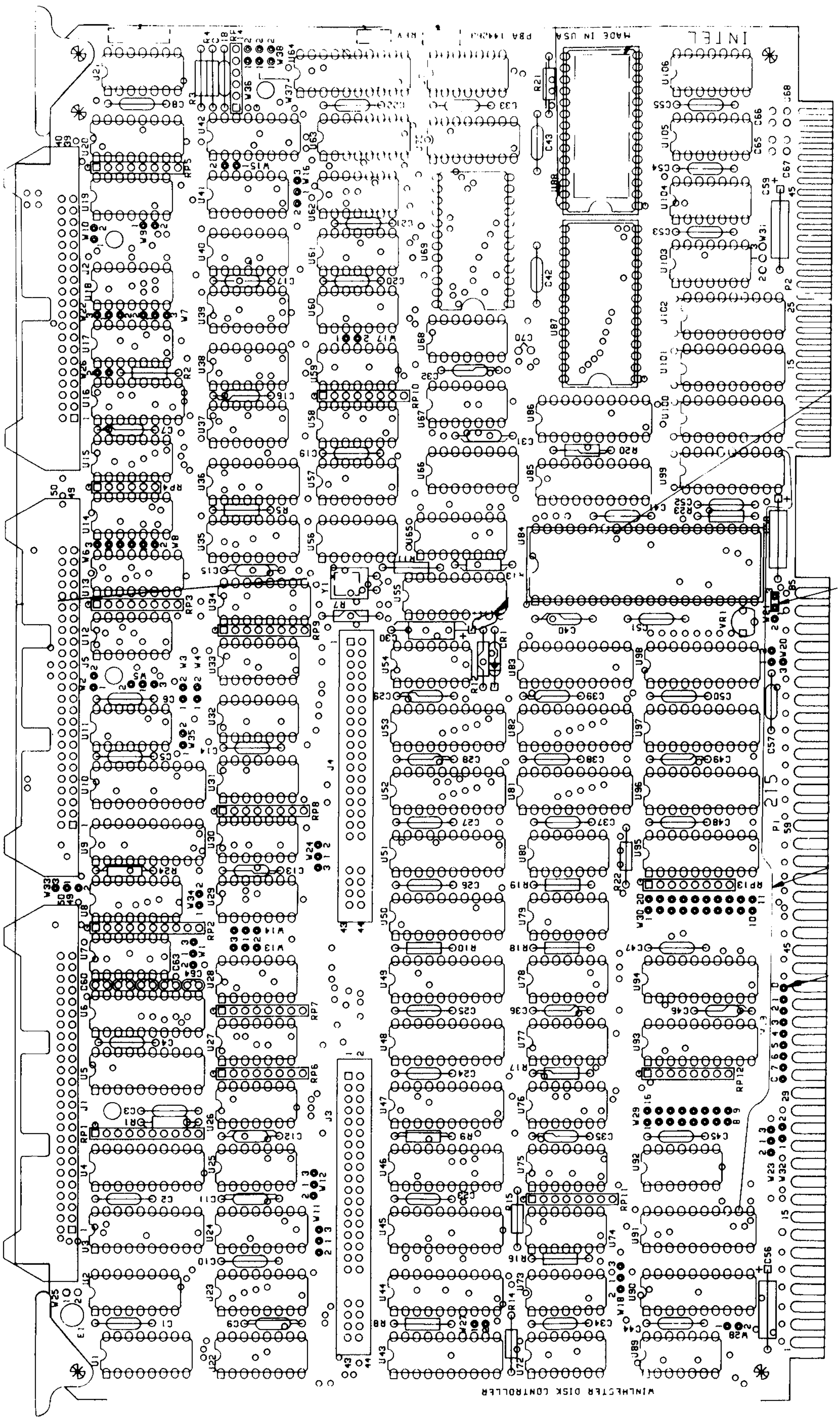


Figure 5-1. Board Parts Location Diagram

Table 5-1. Jumper Connections

Jumper Number	Schematic Sheet Number	Default Connection[1]	Signal Mnemonic
W1	12	W1-1 -- 2	CMD BUS ENB*
W2	13	W2-1 -- 2	VENDOR Ø
W3	11	W3-1 -- 2	EXTRØ
W4	11	W4-1 -- 2	EXTR1
W5	1Ø	W5-1 -- 2	RDØ-
W6	1Ø	W6-1 -- 2	RDØ+
W7	1Ø	W7-1 -- 2	RDCLØ-
W8	1Ø	W8-1 -- 2	RDCLØ+
W9	13	--	TRIPOLAR*
W1Ø	13	W1Ø-1 -- 2	RADIAL SELECT
W11	12	W11-1 -- 3	OPØ1
W12	12	--	OP1Ø/OP11
W13	12	W13-1 -- 3	RDGATE
W14	12	W14-1 -- 3	AM CNTRL
W15	13	W15-1 -- 2	SHUGART
W16	8	W16-1 -- 3	INDEX SECTOR*
W17	11	W17-1 -- 2	INDEX B*
W18	3	W18-1 -- 2	ANYRQST
W19	3	W19-C -- 5	INT5
W2Ø	1	--	-12 V
W21	1	W21-1 -- 3	-5 V
W22	1Ø	W22-1 -- 3	RDCL
W23	3	W23-1 -- 2	CBRQ*
W24	13	--	DREQØ
W25	1	W25-1 -- 2	GND
W26	12	--	VENDOR 1
W27	11	--	VENDOR 2
W28	3	W28-1 -- 2	BPRO*
W29	2	W29-5 -- 12	ADRB*
W29	2	W29-8 -- 9	ADR8*
W3Ø	2	W3Ø-1 -- 2Ø	16 BIT SYS BUS
W31[2]	6	W31-1 -- 2	VCC
W32	4	--	LOCK*
W33	11	W33-1 -- 3	INDEX*
W34	11	W34-1 -- 2	SKCOM*
W35	11	W35-1 -- 2	RDY*
W36	9	--	RGØ/F
W37	9	--	VENDOR 3
W38	9	--	VENDOR 4

Note: 1. Default connections pertain to the iSBC 215G board used in conjunction with an X3T9/1226 ANSI drive.

2. Default connection for iSBC 215G board.

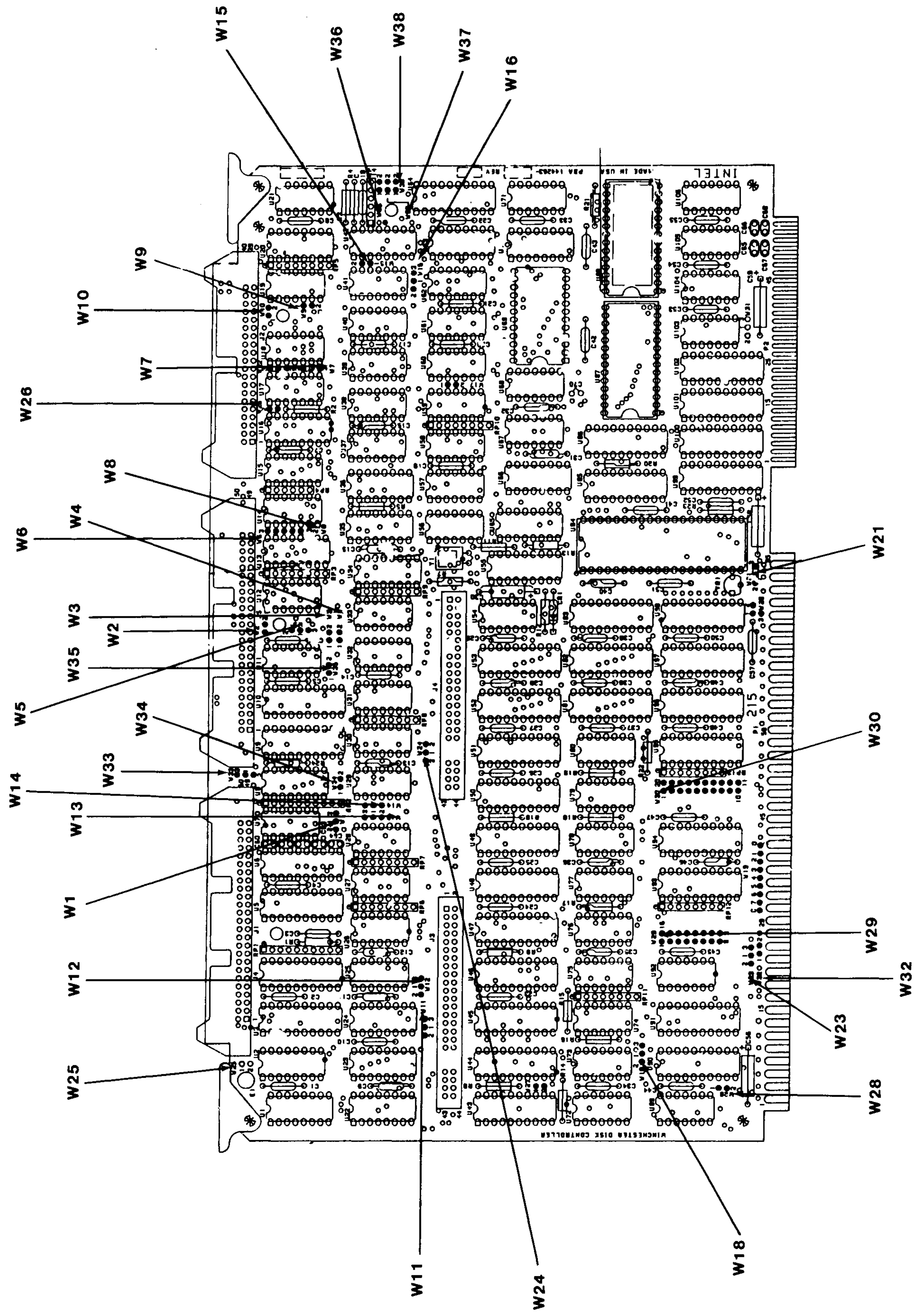
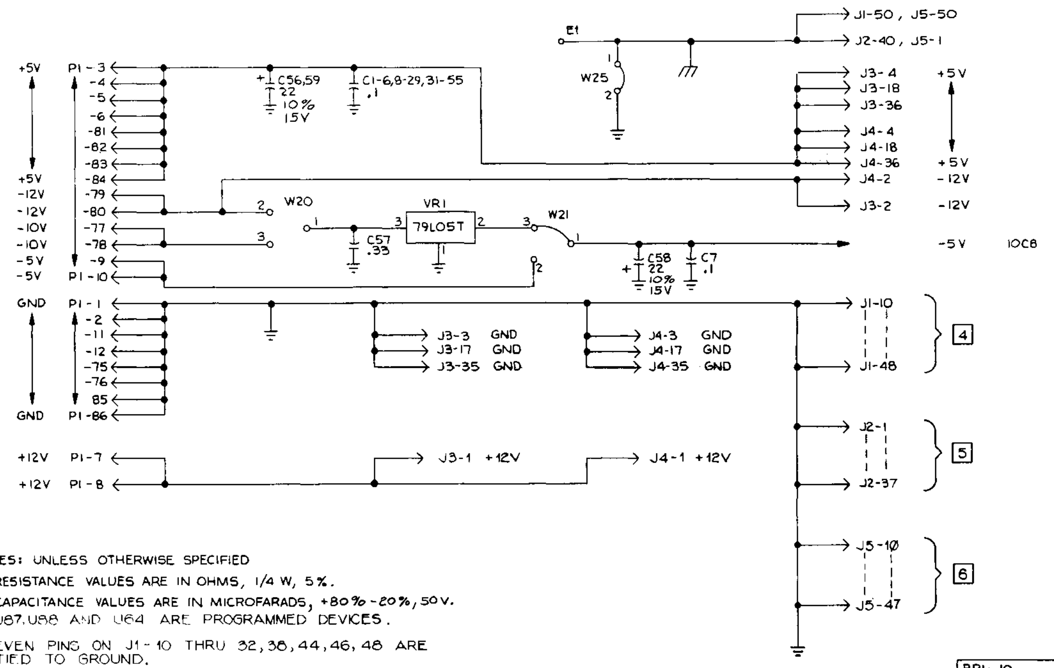


Figure 5-2. Jumper Location Diagram

WAKE-UP ADDRESS 8/16 SYSTEM DATA BUS AND 8/16 I/O PORT BUS JUMPER TABLE

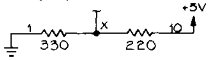
JUMPER From	To	FUNCTION
W29-1	W29-16	WUA Bit F
W29-2	W29-15	WUA Bit E
W29-3	W29-14	WUA Bit D
W29-4	W29-13	WUA Bit C
W29-5	W29-12	WUA Bit B
W29-6	W29-11	WUA Bit A
W29-7	W29-10	WUA Bit 9
W29-8	W29-9	WUA Bit 8
W30-1	W30-20	8/16 System Data Bus
W30-2	W30-19	8/16 I/O Port Bus
W30-3	W30-18	WUA Bit 7
W30-4	W30-17	WUA Bit 6
W30-5	W30-16	WUA Bit 5
W30-6	W30-15	WUA Bit 4
W30-7	W30-14	WUA Bit 3
W30-8	W30-13	WUA Bit 2
W30-9	W30-12	WUA Bit 1
W30-10	W30-11	WUA Bit 0

REV	DESCRIPTION	DFT	DATE	CHK	DATE	APVD	DATE
A	ECO 40-1635	R. Throng	12/3/84	KL	12/3/84	Kawai	12-3-84

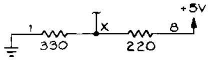


NOTES: UNLESS OTHERWISE SPECIFIED

1. RESISTANCE VALUES ARE IN OHMS, 1/4 W, 5%.
2. CAPACITANCE VALUES ARE IN MICROFARADS, +80%-20%, 50V.
3. U87, U88 AND U64 ARE PROGRAMMED DEVICES.
4. EVEN PINS ON J1-10 THRU 32, 38, 44, 46, 48 ARE TIED TO GROUND.
5. ODD PINS ON J2-1, 7, 13 THRU 19, 25, 31 THRU 33, 35, 37, & EVEN PINS ON J2-4, 10, 22, 28, 32, ARE TIED TO GROUND.
6. EVEN PINS ON J5-10 THRU 38 & ODD PINS ON J5-41, 44, 47 ARE TIED TO GROUND.
7. RP2-X DENOTES



8. RP6-X DENOTES



9. NOT INSTALLED.
10. AN ASTERISK FOLLOWING A SIGNAL NAME DENOTES NEGATION.

POWER & GROUND LOCATION CHART

REFERENCE DESIGNATIONS	DEVICE TYPE	POWER PINS				
		GND	+5V	-5V	-10V	-12V
U84	8089A-3	1,20	40			
U58,74	7406	7	14			
U7, 27	7438	7	14			
U60	74S02	7	14			
U17	74S04	7	14			
U18	74S74	7	14			
U66	74S133	8	16			
U71	74S280	7	14			
U11,61,68	74LS00	7	14			
U33,73	74LS02	7	14			
U25, 28	74LS04	7	14			
U12,29,59,72	74LS08	7	14			
U40,75	74LS10	7	14			
U31	74LS14	7	14			
U54	74LS20	7	14			
U30,32,38,41,67,76,92	74LS32	7	14			
U19,26	74LS51	7	14			
U21,37,39,56,57,62	74LS74	7	14			
U15,89	74LS125	7	14			
U35,36	74LS138	8	16			
U65	74LS139	8	16			
U8	74LS153	8	16			
U20	74LS161	8	16			
U103,104,105,106	74LS164	7	14			
U1, 2, 22, 23	74LS165	8	16			
U42	74LS174	8	16			
U24	74LS175	8	16			
U43,45,93,94,95	74LS244	10	20			
U70	74LS257	8	16			
U77, 78, 79, 80	74LS266	7	14			
U3	74LS273	10	20			
U34, 63	74LS279	8	16			
U47, 50	74LS299	10	20			
U9,85,86	74LS373	10	20			
U10,44,46,48,49,51	74LS374	10	20			
U14	75107A	7	14	13		
U13	MC3486	8	16			
U16	MC3487	8	16			
U99,100,101,102	2114A-5	9	18			
U69	8254	12	24			
U4,5,6,81,82,83	8283	10	20			
U55	8284-A	9	18			
U52,53	8286	10	20			
U96,97,98	8287	10	20			
U91	8288	10	20			
U90	8289	10	20			
3 U87,88	2764	14	28			
U64	PAL16R4	10	20			

RP1-10
RP2-9
RP4-6
RP6-6
RP7-2,3,4,5,6
RP8-6,7,8
RP9-2,4
RP10-7
RP11-2
UNUSED RP RESISTORS

RP14	
J5	
P1	
E1	
W38	
VR1	
CR1	
Y1	
R24	R6
C71	C65-C69
U106	
LAST USED	NOT USED

TYPE	REF DES	QTY
74LS00	U11	1
74LS10	U75	2
74LS08	U72	2
74LS04	U28	1

SPARE GATES

QUANTITY PER DASH NO. | ITEM NO. | DESCRIPTION | PARTS LIST

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

	SIGNATURE	DATE
DRN BY	R. Throng	12/01/84
CHK BY	Kagi ick	12/3/84
ENGR APVD	Kawai	12/3/84
APVD		
APVD		

intel® 3065 BOWERS AVE SANTA CLARA CALIF. 95051

TITLE: SCHEMATIC iSBC 215G

SIZE	DCLC	CL CODE	RLSE LVL	DOCUMENT NUMBER	REV
D	40	0100		147631	A

SCALE NONE SHEET 1 OF 14

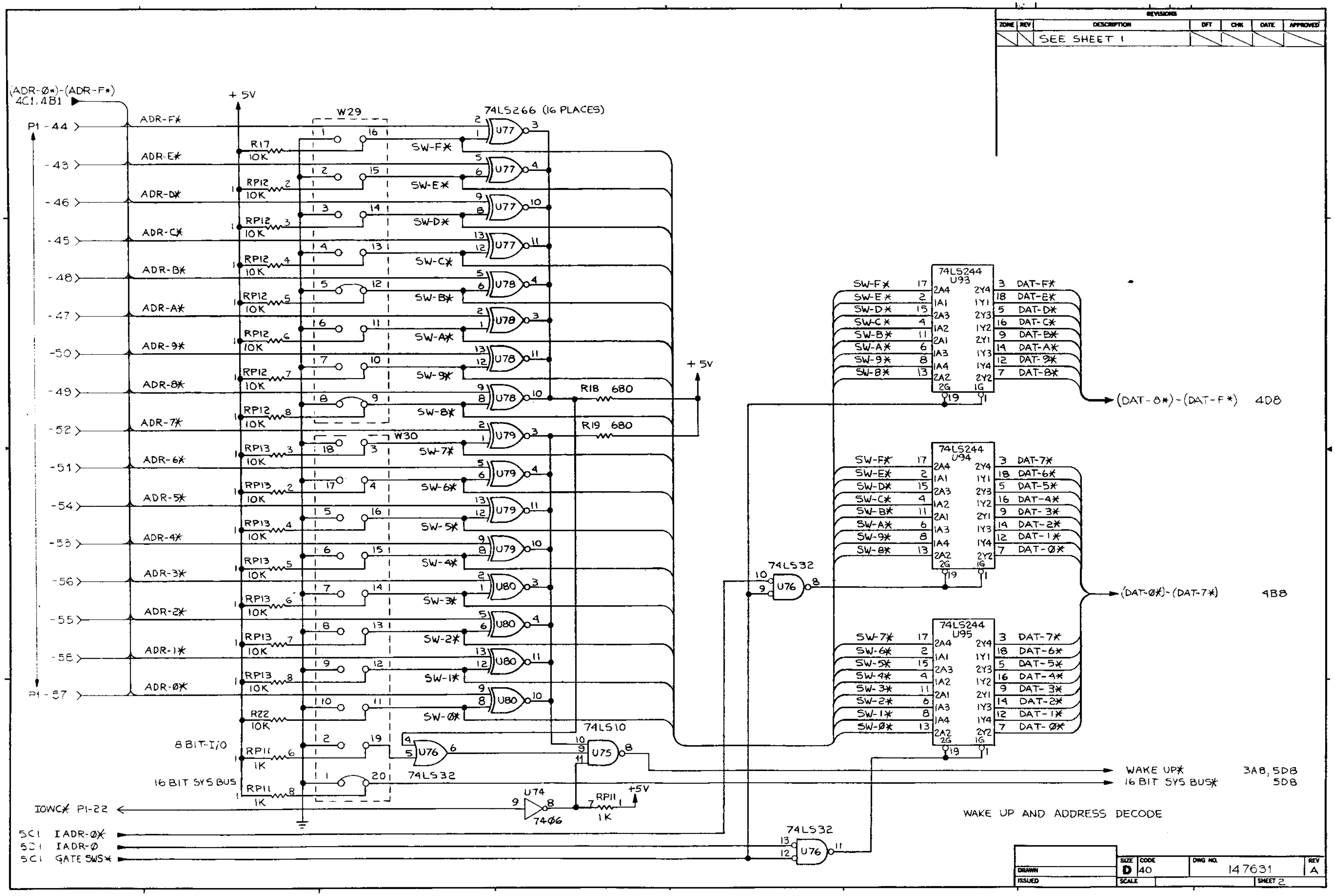
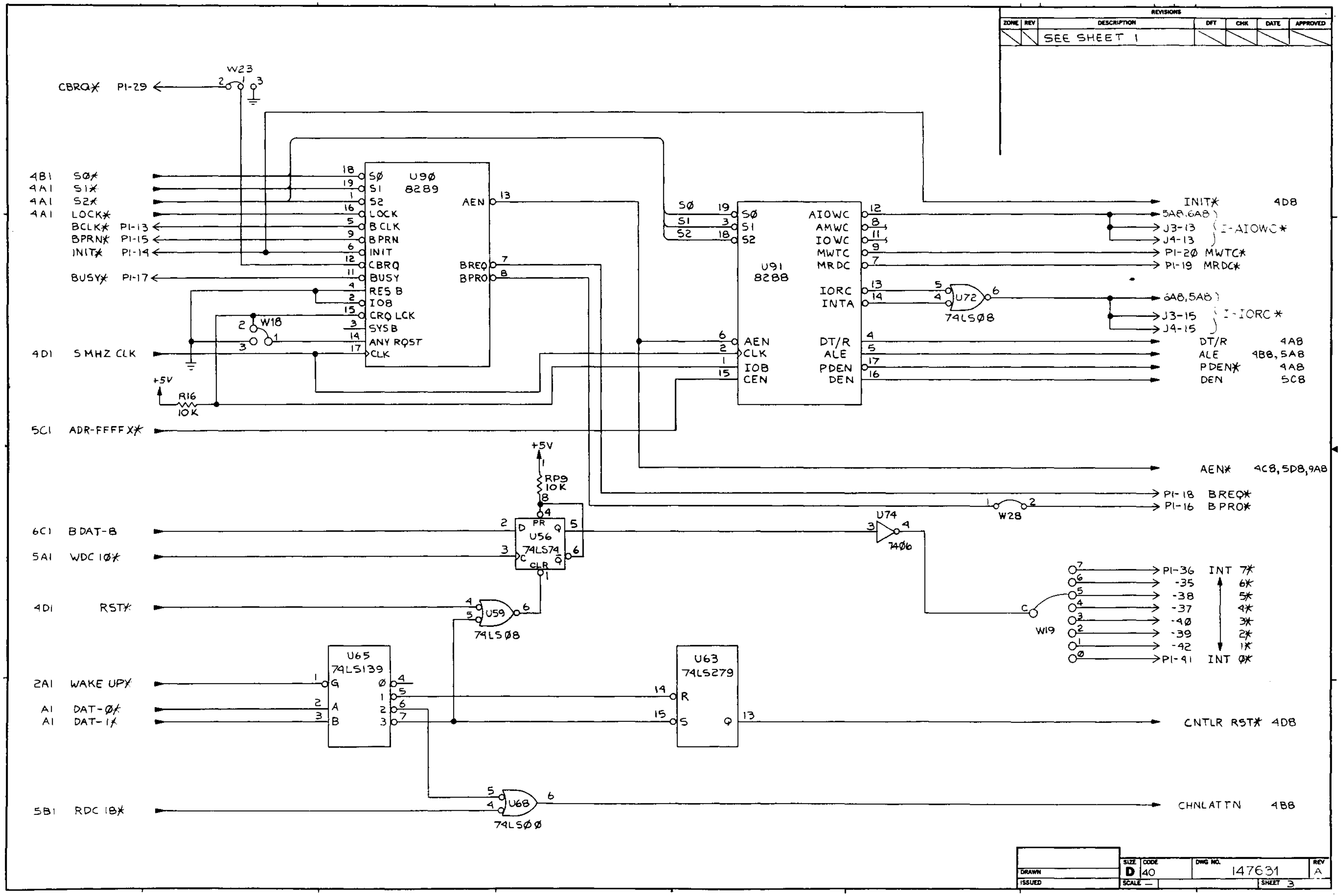
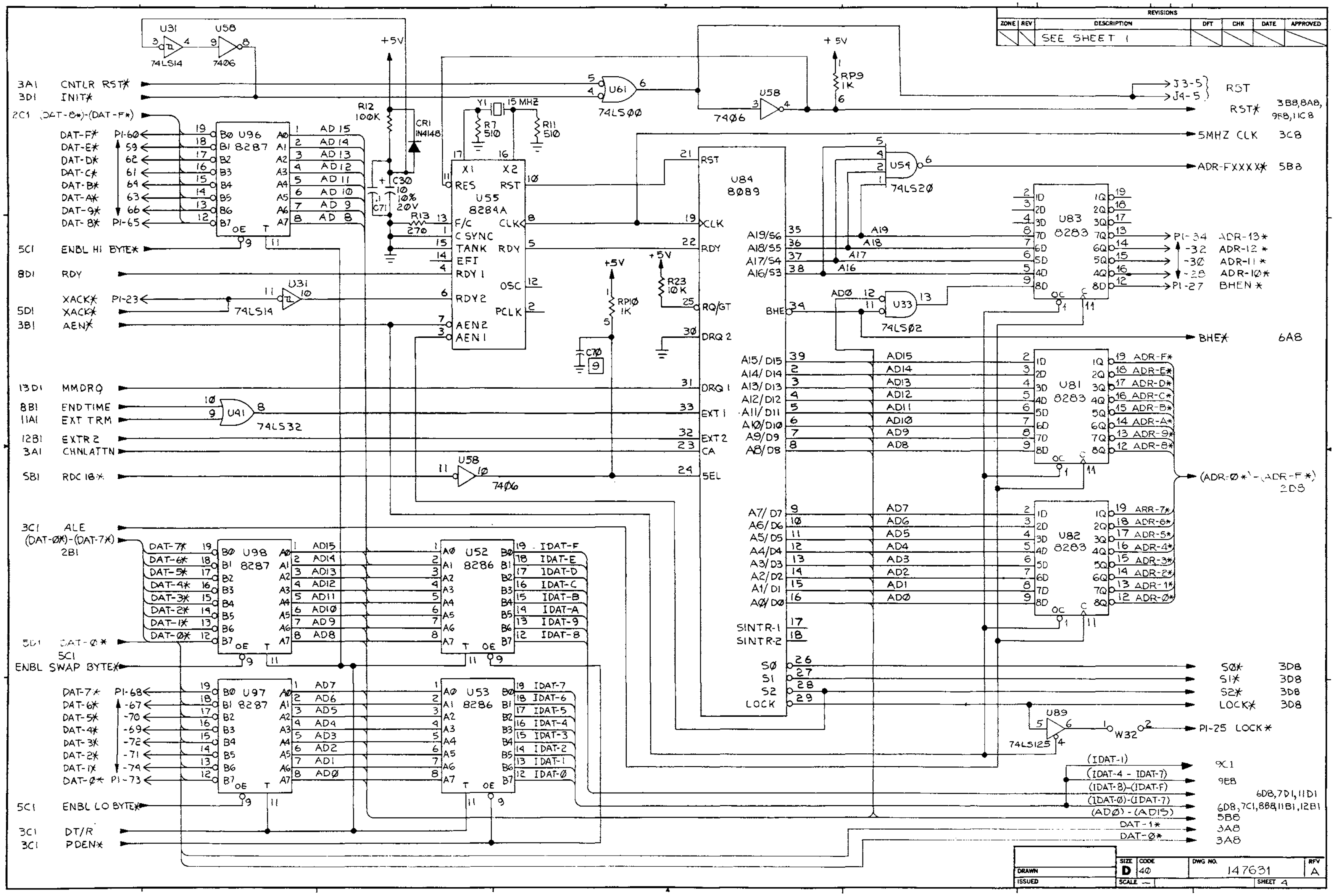
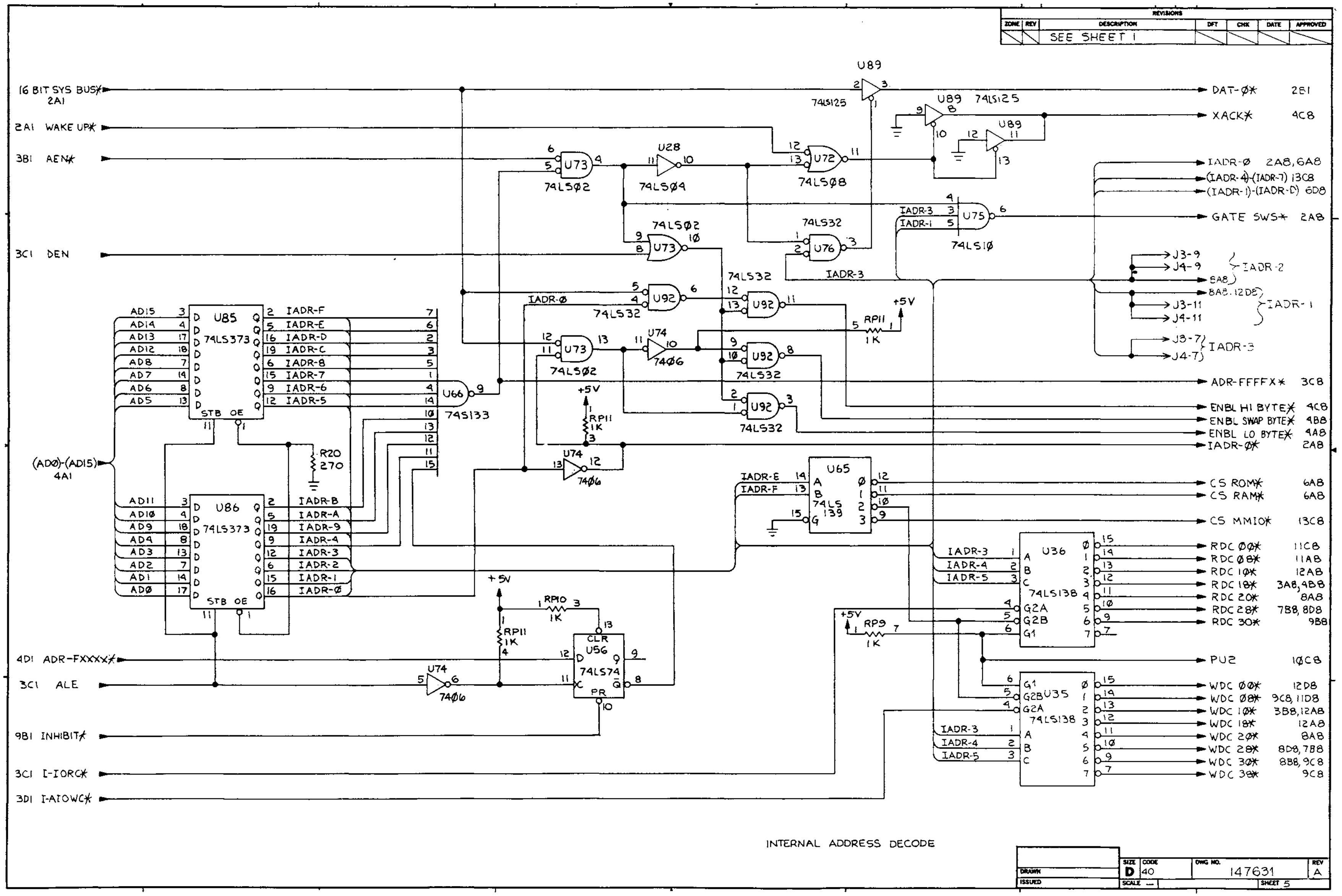


Figure 5-4. Board Schematic Diagram (Sheet 2 of 14)

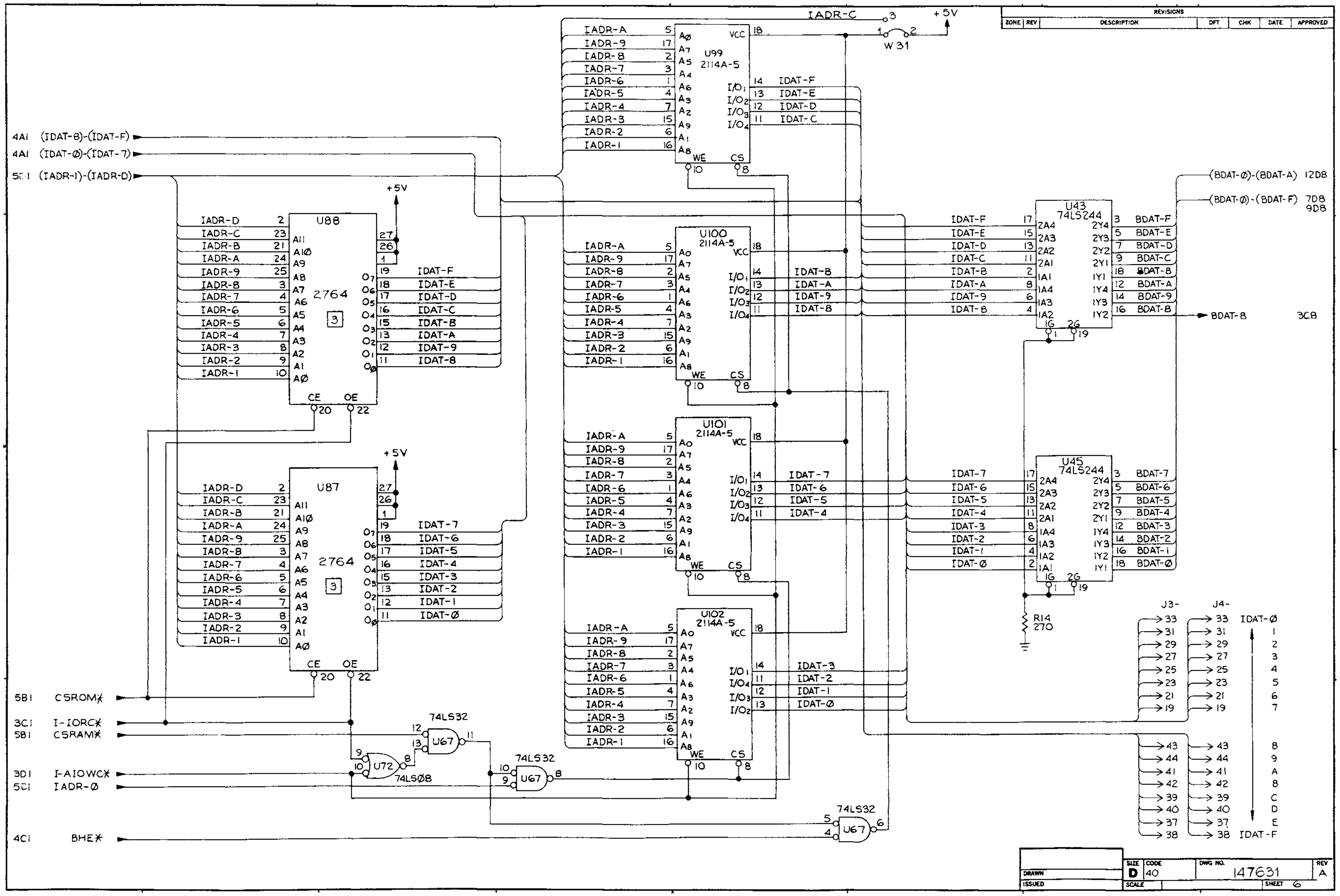

REVISIONS
ZONE REV DESCRIPTION DFT CHK DATE APPROVED
SEE SHEET 1
W23
CBRQ* P1-29
4B1 S0*
4A1 S1*
4A1 S2*
4A1 LOCK*
BCLK* P1-13
BPRN* P1-15
INIT* P1-14
BUSY* P1-17
U90 8289
S0 S1 S2 LOCK BCLK BPRN INIT CBRQ BUSY RES B IOB CRQ LCK SYSB ANY RQST CLK
AEN BREQ BPRO
W18
4D1 5MHZ CLK
+5V
R16 10K
5C1 ADR-FFFFX*
U91 8288
S0 S1 S2 AEN CLK IOB CEN
AIOWC AMWC IOWC MWTC MRDC IORC INTA DT/R ALE PDEN DEN
U72 74LS08
INIT* 4D8
5A8,6A8
J3-13
J4-13
I-AIOWC*
P1-20 MWTC*
P1-19 MRDC*
6A8,5A8
J3-15
J4-15
I-IORC*
DT/R 4A8
ALE 4B8,5A8
PDEN* 4A8
DEN 5C8
AEN* 4C8,5D8,9A8
P1-18 BREQ*
P1-16 BPRO*
W28
+5V
RP9 10K
6C1 BDAT-8
5A1 WDC 10*
U56 74LS74
PR D Q C Q̄ CLR
U74 7406
4D1 RST*
U59 74LS08
W19
P1-36 INT 7*
-35 6*
-38 5*
-37 4*
-40 3*
-39 2*
-42 1*
P1-41 INT 0*
2A1 WAKE UP*
A1 DAT-0*
A1 DAT-1*
U65 74LS139
U63 74LS279
R S Q
CNTLR RST* 4D8
5B1 RDC 18*
U68 74LS00
CHNLATTN 4B8
DRAWN
ISSUED
SIZE D CODE 40
DWG NO. 147631
REV A
SCALE —
SHEET 3



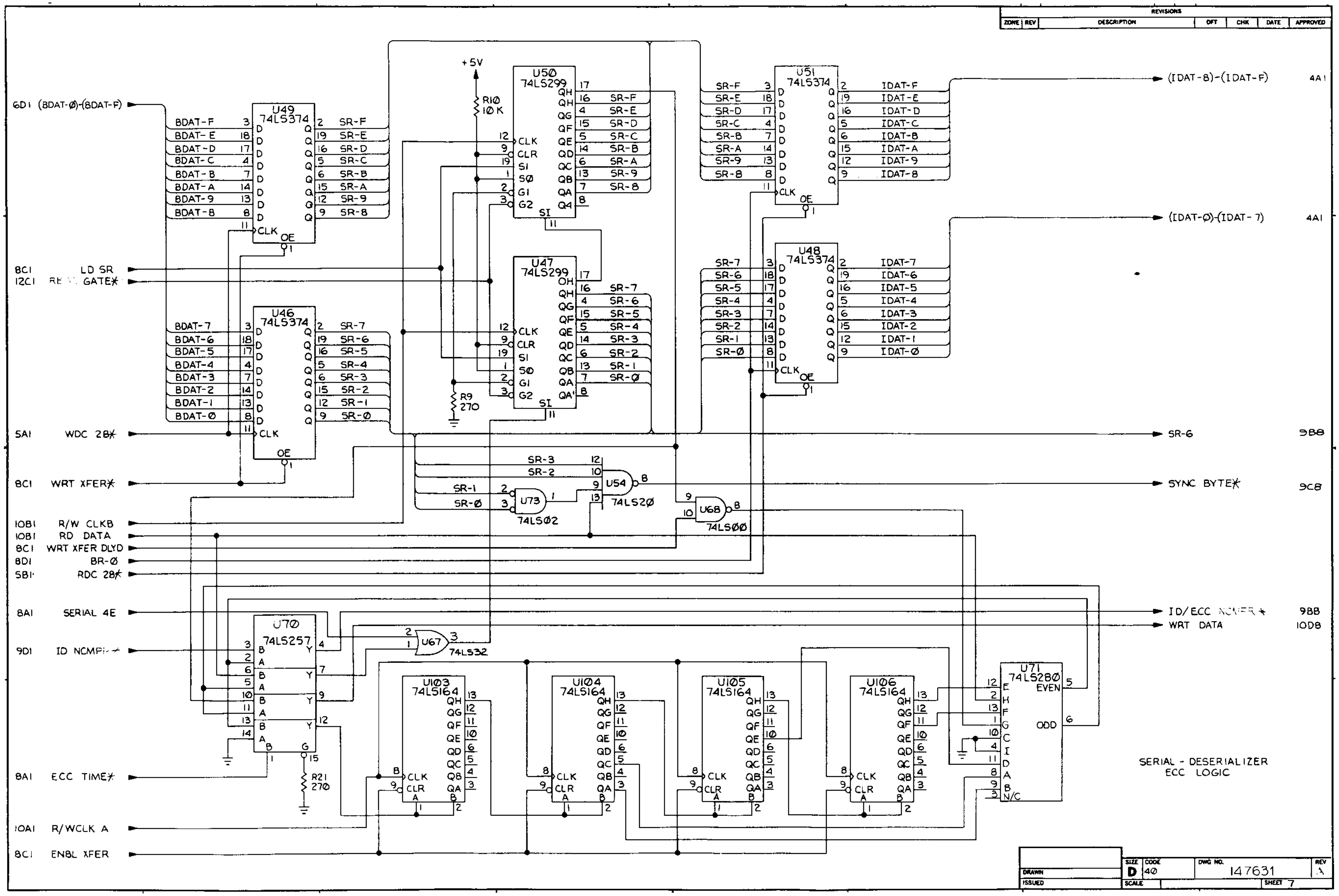
REVISIONS
ZONE REV DESCRIPTION DFT CHK DATE APPROVED
SEE SHEET 1
3A1 CNTLR RST*
3D1 INIT*
2C1 (DAT-8*)-(DAT-F*)
U31 74LS14
U58 7406
U61 74LS00
U96 8287
U98 8287
U97 8287
U52 8286
U53 8286
U55 8284A
U84 8089
U54 74LS20
U83 8283
U81 8283
U82 8283
U33 74LS02
U41 74LS32
U89 74LS125
Y1 15 MHZ
R12 100K
CR1 IN4148
C30 10 10% 20V
C71 .1
R13 270
R7 510
R11 510
RP9 1K
RP10 1K
R23 10K
C70
+5V
5C1 ENBL HI BYTE*
8D1 RDY
XACK* P1-23
5D1 XACK*
3B1 AEN*
13D1 MMDRQ
8B1 END TIME
11A1 EXT TRM
12B1 EXTR 2
3A1 CHNLATTN
5B1 RDC 18*
3C1 ALE
(DAT-Ø*)-(DAT-7*) 2B1
5D1 DAT-Ø*
5C1 ENBL SWAP BYTE*
5C1 ENBL LO BYTE*
3C1 DT/R
3C1 PDEN*
J3-5 J4-5 RST
RST* 3B8,8A8, 9B8,11C8
5MHZ CLK 3C8
ADR-FXXXX* 5B8
P1-34 ADR-13*
-32 ADR-12*
-30 ADR-11*
-28 ADR-10*
P1-27 BHEN*
BHE* 6A8
(ADR-Ø*)-(ADR-F*) 2D8
SØ* 3D8
S1* 3D8
S2* 3D8
LOCK* 3D8
P1-25 LOCK*
W32
(IDAT-1) 9C1
(IDAT-4 - IDAT-7) 9E8
(IDAT-8)-(IDAT-F) 6D8,7D1,11D1
(IDAT-Ø)-(IDAT-7) 6D8,7C1,8B8,11B1,12B1
(ADØ)-(AD15) 5B8
DAT-1* 3A8
DAT-Ø* 3A8
SIZE D CODE 40 DWG NO. 147631 REV A
DRAWN ISSUED SCALE — SHEET 4



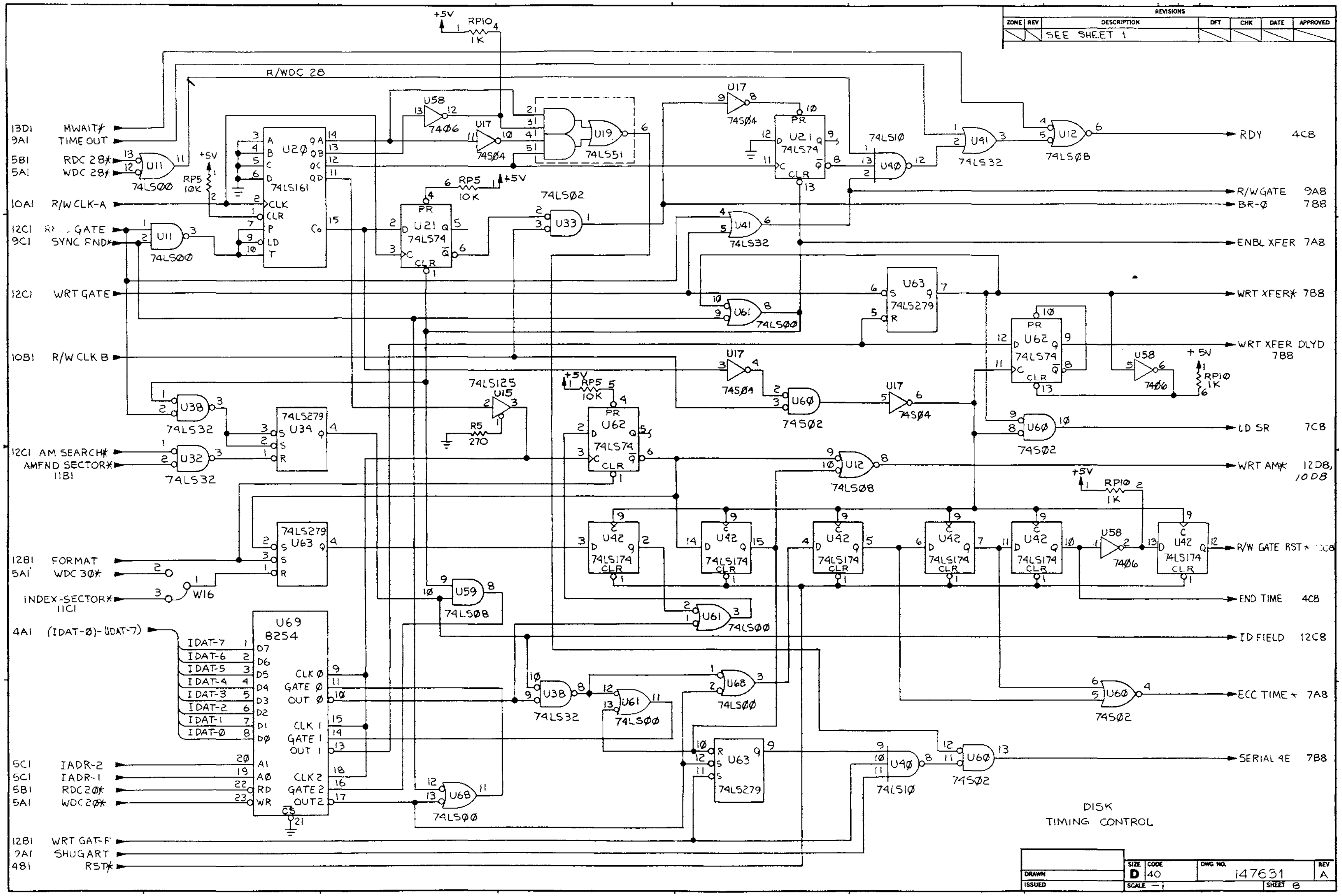
INTERNAL ADDRESS DECODE



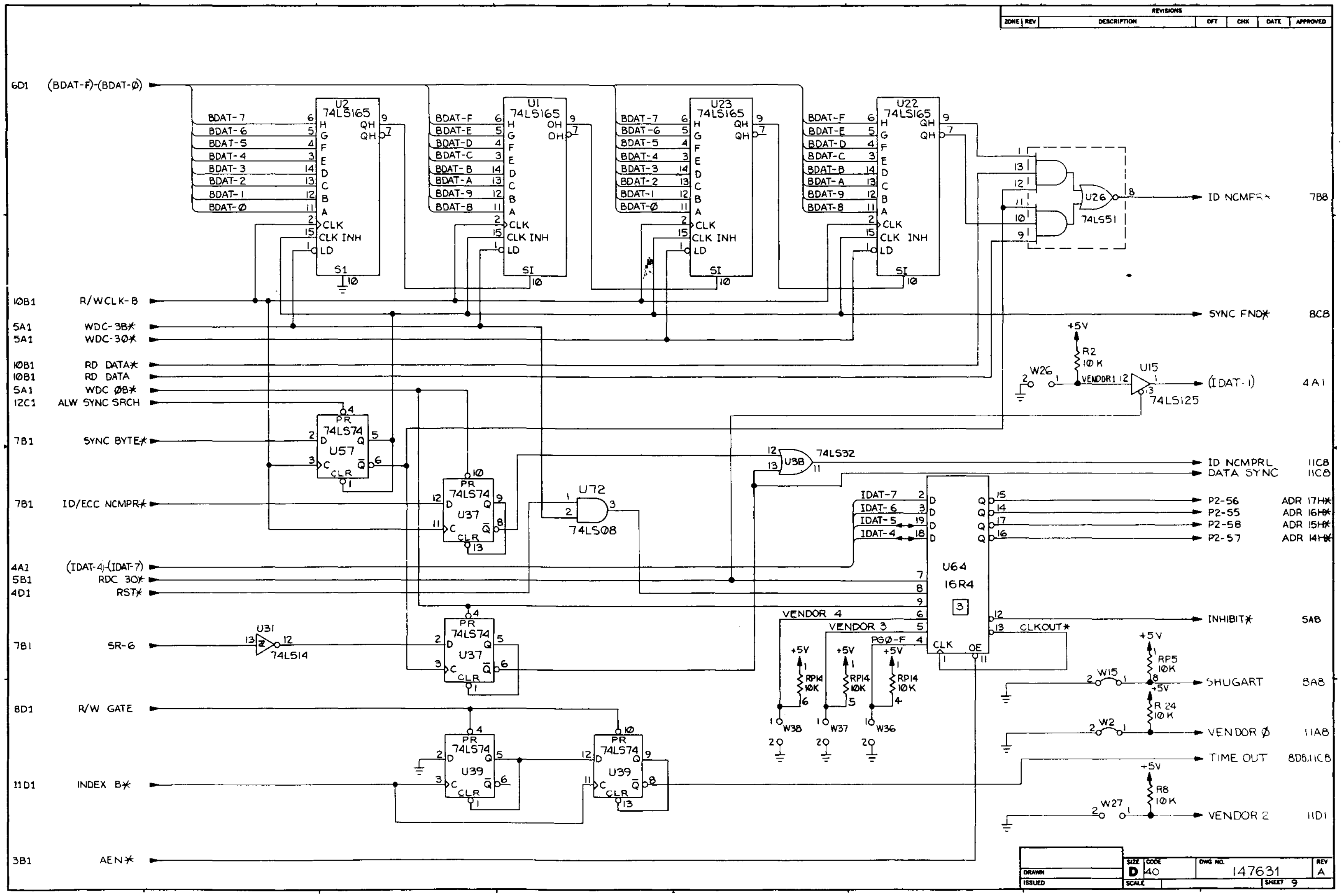
4A1 (IDAT-8)-(IDAT-F)
4A1 (IDAT-Ø)-(IDAT-7)
5C1 (IADR-1)-(IADR-D)
U88
2764
U87
2764
+5V
U99
2114A-5
U100
2114A-5
U101
2114A-5
U102
2114A-5
IADR-C
W31
U43
74LS244
U45
74LS244
(BDAT-Ø)-(BDAT-A) 12D8
(BDAT-Ø)-(BDAT-F) 7D8 9D8
BDAT-8 3C8
R14
270
5B1 CSROM*
3C1 I-IORC*
5B1 CSRAM*
3D1 I-AIOWC*
5C1 IADR-Ø
4C1 BHE*
74LS32
U67
U72
74LS08
J3-
J4-
IDAT-Ø
IDAT-F
REVISIONS
ZONE
REV
DESCRIPTION
DFT
CHK
DATE
APPROVED
DRAWN
ISSUED
SIZE
D
CODE
40
DWG NO.
147631
REV
A
SCALE
SHEET 6

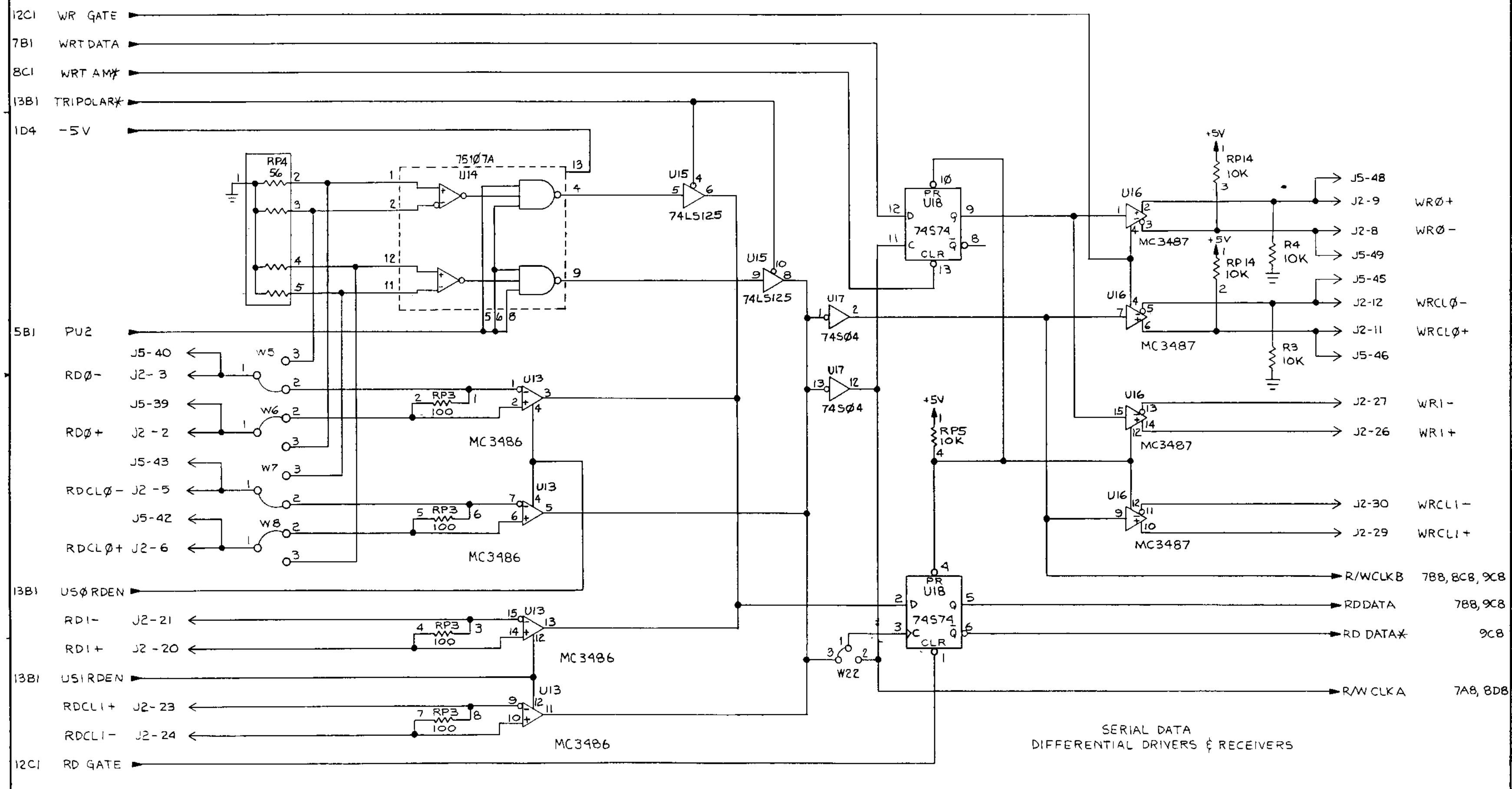


REVISIONS
ZONE REV DESCRIPTION DFT CHK DATE APPROVED
6D1 (BDAT-Ø)-(BDAT-F)
8C1 LD SR
12C1 REV GATE*
5A1 WDC 28*
8C1 WRT XFER*
10B1 R/W CLKB
10B1 RD DATA
8C1 WRT XFER DLYD
8D1 BR-Ø
5B1 RDC 28*
8A1 SERIAL 4E
9D1 ID NCMP*
8A1 ECC TIME*
10A1 R/WCLK A
8C1 ENBL XFER
U49 74LS374
U46 74LS374
U50 74LS299
U47 74LS299
U51 74LS374
U48 74LS374
+5V
R10 10K
R9 270
R21 270
U54 74LS20
U73 74LS02
U68 74LS00
U67 74LS32
U70 74LS257
U103 74LS164
U104 74LS164
U105 74LS164
U106 74LS164
U71 74LS280
EVEN
ODD
(IDAT-8)-(IDAT-F) 4A1
(IDAT-Ø)-(IDAT-7) 4A1
SR-6 9B8
SYNC BYTE* 9C8
ID/ECC NCMPR* 9B8
WRT DATA 10D8
SERIAL - DESERIALIZER
ECC LOGIC
DRAWN
ISSUED
SIZE D
CODE 40
DWG NO. 147631
REV A
SCALE
SHEET 7



REVISIONS
ZONE | REV | DESCRIPTION | DFT | CHK | DATE | APPROVED
SEE SHEET 1
13D1 MWAIT*
9A1 TIME OUT
5B1 RDC 28*
5A1 WDC 28*
10A1 R/W CLK-A
12C1 R/W GATE
9C1 SYNC FND*
12C1 WRT GATE
10B1 R/W CLK B
12C1 AM SEARCH*
AMFND SECTOR*
11B1
12B1 FORMAT
5A1 WDC 30*
INDEX-SECTOR*
11C1
4A1 (IDAT-Ø)-(IDAT-7)
5C1 IADR-2
5C1 IADR-1
5B1 RDC 20*
5A1 WDC 20*
12B1 WRT GAT-F
7A1 SHUGART
4B1 RST*
R/WDC 28
RDY 4C8
R/W GATE 9A8
BR-Ø 7B8
ENBL XFER 7A8
WRT XFER* 7B8
WRT XFER DLYD 7B8
LD SR 7C8
WRT AM* 12D8, 10D8
R/W GATE RST* 12C8
END TIME 4C8
ID FIELD 12C8
ECC TIME* 7A8
SERIAL 4E 7B8
U20 74LS161
U21 74LS74
U19 74LS51
U33 74LS02
U41 74LS32
U40 74LS10
U12 74LS08
U61 74LS00
U63 74LS279
U62 74LS74
U60 74S02
U58 7406
U17 74S04
U15 74LS125
U38 74LS32
U32 74LS32
U39 74LS279
U42 74LS174
U59 74LS08
U68 74LS00
U69 8254
U11 74LS00
RP5 10K
RP10 1K
R5 270
W16
+5V
DISK
TIMING CONTROL
DRAWN
ISSUED
SIZE D
CODE 40
DWG NO. 147631
REV A
SCALE —
SHEET 8


REVISIONS
ZONE REV DESCRIPTION DFT CHK DATE APPROVED
6D1 (BDAT-F)-(BDAT-Ø)
U2 74LS165
U1 74LS165
U23 74LS165
U22 74LS165
BDAT-7 BDAT-6 BDAT-5 BDAT-4 BDAT-3 BDAT-2 BDAT-1 BDAT-Ø
BDAT-F BDAT-E BDAT-D BDAT-C BDAT-B BDAT-A BDAT-9 BDAT-8
CLK CLK INH LD SI QH
U26 74LS51
ID NCMPR* 7B8
10B1 R/WCLK-B
5A1 WDC-3B*
5A1 WDC-3Ø*
SYNC FND* 8C8
10B1 RD DATA*
10B1 RD DATA
5A1 WDC ØB*
12C1 ALW SYNC SRCH
+5V R2 10K W26 VENDOR1 U15 74LS125
(IDAT-1) 4A1
7B1 SYNC BYTE*
U57 74LS74
U38 74LS32
ID NCMPRL 11C8
DATA SYNC 11C8
7B1 ID/ECC NCMPR*
U37 74LS74
U72 74LS08
IDAT-7 IDAT-6 IDAT-5 IDAT-4
U64 16R4
P2-56 ADR 17H*
P2-55 ADR 16H*
P2-58 ADR 15H*
P2-57 ADR 14H*
4A1 (IDAT-4)-(IDAT-7)
5B1 RDC 30*
4D1 RST*
VENDOR 4
VENDOR 3
PGØ-F
CLK OE
INHIBIT* 5A8
CLKOUT*
7B1 SR-6
U31 74LS14
U37 74LS74
RP14 10K W38 W37 W36
RP5 10K W15
SHUGART 8A8
R24 10K W2
VENDOR Ø 11A8
8D1 R/W GATE
U39 74LS74
U39 74LS74
TIME OUT 8D8,11C8
11D1 INDEX B*
R8 10K W27
VENDOR 2 11D1
3B1 AEN*
DRAWN
ISSUED
SIZE D CODE 40
DWG NO. 147631
REV A
SCALE
SHEET 9



SERIAL DATA
DIFFERENTIAL DRIVERS & RECEIVERS

REVISIONS						
ZONE	REV	DESCRIPTION	DFT	CHK	DATE	APPROVED
		SEE SHEET 1				

DRAWN	SIZE D	CODE 40	DWG NO. 147631	REV A
ISSUED	SCALE —		SHEET 10	

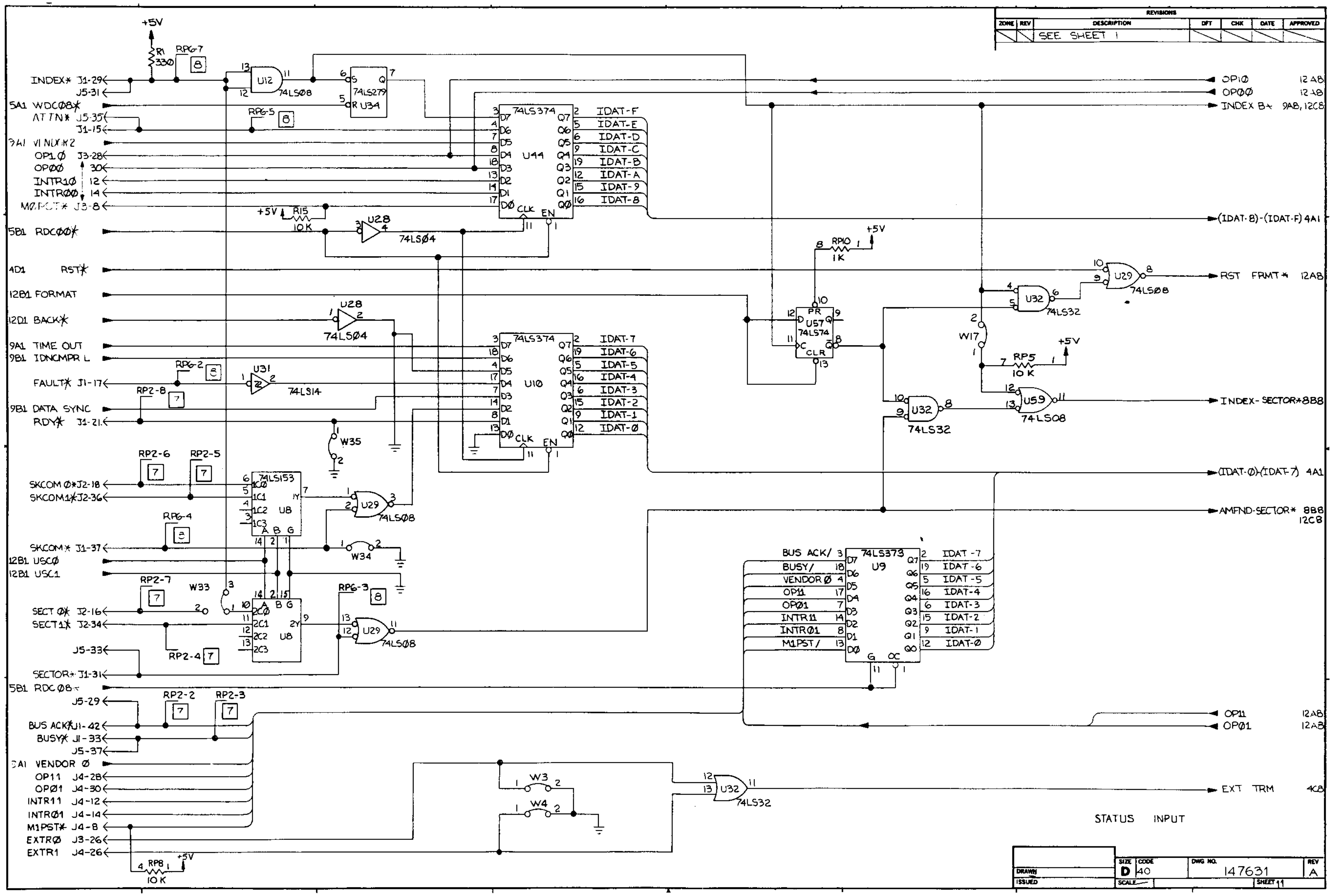
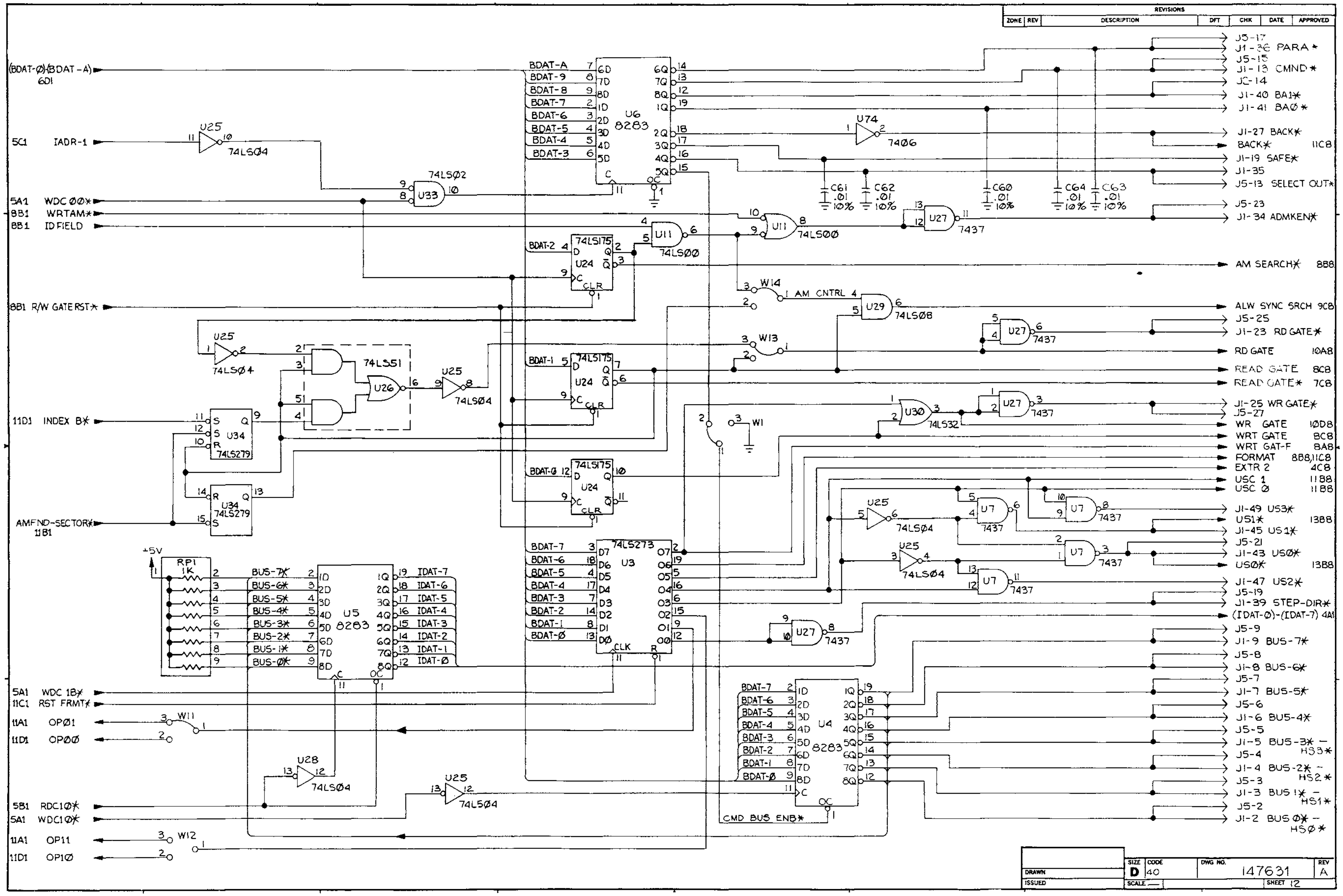
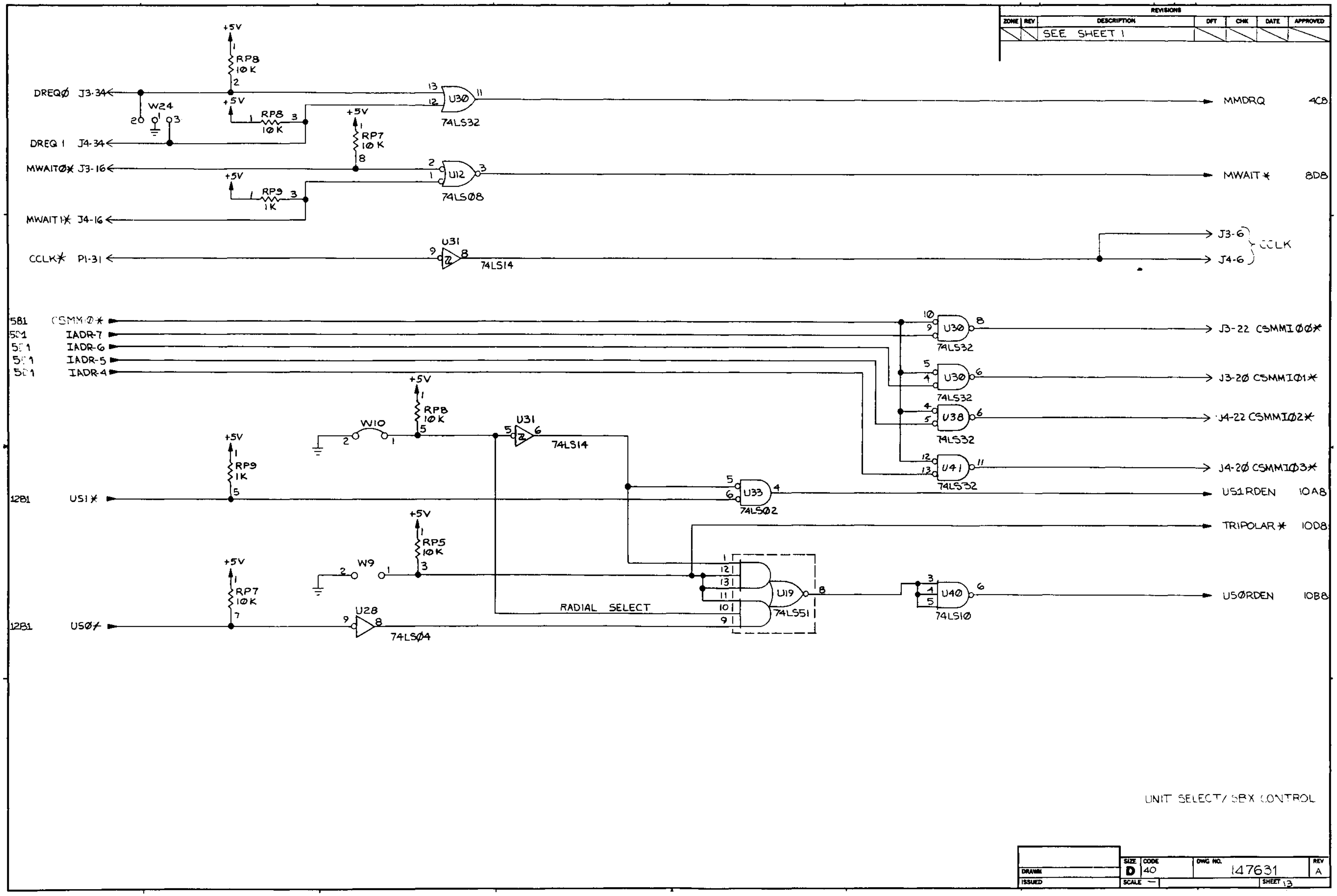


Figure 5-4. Board Schematic Diagram (Sheet 11 of 14)



REVISIONS
ZONE REV DESCRIPTION DFT CHK DATE APPROVED
(BDAT-Ø)-(BDAT-A) 6D1
5C1 IADR-1
5A1 WDC ØØ*
8B1 WRTAM*
8B1 IDFIELD
8B1 R/W GATE RST*
11D1 INDEX B*
AMFND-SECTOR* 11B1
5A1 WDC 1B*
11C1 RST FRMT*
11A1 OPØ1
11D1 OPØØ
5B1 RDC1Ø*
5A1 WDC1Ø*
11A1 OP11
11D1 OP1Ø
U6 8283
U5 8283
U4 8283
74LS273 U3
U24 74LS175
U34 74LS279
74LS51
U25 74LSØ4
U33 74LSØ2
U11 74LSØØ
U29 74LSØ8
U30 74LS32
U27 7437
U7 7437
U74 7406
U28 74LSØ4
RP1 1K
+5V
C61 .01 10%
C62 .01 10%
C60 .01 10%
C64 .01 10%
C63 .01 10%
W1 W11 W12 W13 W14
AM CNTRL
CMD BUS ENB*
J5-17
J1-36 PARA*
J5-15
J1-13 CMND*
J5-14
J1-40 BA1*
J1-41 BAØ*
J1-27 BACK*
BACK* 11C8
J1-19 SAFE*
J1-35
J5-13 SELECT OUT*
J5-23
J1-34 ADMKEN*
AM SEARCH* 8B8
ALW SYNC SRCH 9C8
J5-25
J1-23 RD GATE*
RD GATE 10A8
READ GATE 8C8
READ GATE* 7C8
J1-25 WR GATE*
J5-27
WR GATE 10D8
WRT GATE 8C8
WRT GAT-F 8A8
FORMAT 8B8,11C8
EXTR 2 4C8
USC 1 11B8
USC Ø 11B8
J1-49 US3*
US1* 13B8
J1-45 US1*
J5-21
J1-43 USØ*
USØ* 13B8
J1-47 US2*
J5-19
J1-39 STEP-DIR*
(IDAT-Ø)-(IDAT-7) 4A1
J5-9
J1-9 BUS-7*
J5-8
J1-8 BUS-6*
J5-7
J1-7 BUS-5*
J5-6
J1-6 BUS-4*
J5-5
J1-5 BUS-3* — HS3*
J5-4
J1-4 BUS-2* — HS2*
J5-3
J1-3 BUS 1* — HS1*
J5-2
J1-2 BUS Ø* — HSØ*
DRAWN
ISSUED
SIZE D CODE 40
DWG NO. 147631
REV A
SCALE —
SHEET 12



REVISIONS
ZONE | REV | DESCRIPTION | DFT | CHK | DATE | APPROVED
SEE SHEET 1
DREQØ J3-34
DREQ 1 J4-34
W24
RP8 10K
+5V
U3Ø 74LS32
MMDRQ 4C8
MWAITØ* J3-16
MWAIT1* J4-16
RP7 10K
RP9 1K
U12 74LSØ8
MWAIT* 8D8
CCLK* P1-31
U31 74LS14
J3-6
J4-6
CCLK
CSMMØ*
IADR-7
IADR-6
IADR-5
IADR-4
U30 74LS32
U41 74LS32
J3-22 CSMMIØØ*
J3-2Ø CSMMIØ1*
J4-22 CSMMIØ2*
J4-2Ø CSMMIØ3*
W10
U33 74LSØ2
US1RDEN 10A8
US1*
TRIPOLAR* 10D8
W9
RP5 10K
U19 74LS51
RADIAL SELECT
U40 74LS1Ø
USØRDEN 10B8
USØ*
U28 74LSØ4
12B1
UNIT SELECT/ SBX CONTROL
SIZE D
CODE 40
DWG NO. 147631
REV A
DRAWN
ISSUED
SCALE —
SHEET 13

VENDOR CONFIGURATION TABLE

		DRIVE INTERFACE											
WIRE NO.	FUNCTION	SHUGART SA 1000 RMS 5¼ CMI 5¼	SHUGART SA 4000 MEMOREX FUJITSU	PRIAM	PRIAM ANSI	PERTEC SOFT-SECTOR ANSI	3M ANSI	MICROPOLIS ANSI	BASF ANSI	SLI ANSI	CDC	ANSI PROGRAM HARD SECTORED	ANSI PROGRAM SOFT SECTORED
W1	CMD BUS ENB*	1-3	1-3	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-3	1-2	1-2
W2	VENDOR Ø	—	1-2	1-2	1-2	—	1-2	1-2	1-2	1-2	1-2	1-2	—
W5	RDØ-	1-3	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-3	1-2	1-2
W6	RDØ+	1-3	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-3	1-2	1-2
W7	RDCLØ-	1-3	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-3	1-2	1-2
W8	RDCLØ+	1-3	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-3	1-2	1-2
W9	TRIPOLAR * (SA1000)	1-2	—	—	—	—	—	—	—	—	1-2	—	—
W10	RADIAL SELECT	1-2	—	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2
W13	RD GATE	1-2	1-3	1-3	1-3	1-2	1-3	1-3	1-3	1-3	1-3	1-3	1-2
W14	AM CNTRL (SA1000)	1-2	1-3	1-3	1-3	1-3	1-3	1-3	1-3	1-3	1-3	1-3	1-3
W15	SHUGART (GAP CNTRL)	—	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2
W16	HARD/SOFT SECTORING	1-2	1-3	1-3	1-3	1-2	1-3	1-3	1-3	1-3	1-3	1-3	1-2
W17	INDEX SELECT	1-2	1-2	—	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2	1-2
W22	RD CL	1-2	1-2	1-2	1-3	1-3	1-3	1-3	1-3	1-3	1-2	1-3	1-3
W26	VENDOR 1	—	1-2	1-2	—	—	1-2	—	1-2	—	1-2	1-2	1-2
W27	VENDOR 2	1-2	—	1-2	—	1-2	—	—	—	1-2	—	1-2	1-2
W37	VENDOR 3	1-2	1-2	1-2	—	—	—	1-2	1-2	1-2	1-2	—	—
W38	VENDOR 4	1-2	1-2	1-2	—	—	—	—	—	—	1-2	1-2	1-2
W33	SECTOR	1-2	1-2	1-2	1-3	1-2	1-3	1-3	1-3	1-3	1-2	1-3	1-2
W34	SKCOM *	—	—	—	1-2	1-2	1-2	1-2	1-2	1-2	—	1-2	1-2
W35	RDY *	—	—	—	1-2	1-2	1-2	1-2	1-2	1-2	—	1-2	1-2
W20	-10/-12V	1-2	—	—	—	—	—	—	—	—	—	—	—
W21	-5 V	1-3	—	—	—	—	—	—	—	—	—	—	—

As shown in Table 3-5, bytes 26 through 29 of the I/O parameter block can be set to allow use of the ANSI X3T9/1126 interface. With this feature, present and future ANSI X3T9 drives not fully supported by the iSBC 215G board can still be used by transferring the formatting information to the board at initialization time. This information is contained in the four bytes in the general address pointer of the I/O parameter block and must be there only for initialization. To calculate the values of the four bytes, the user must have available the recommended data format from the disk drive product specification. As an example, assume a data format as shown in Figure A-1.

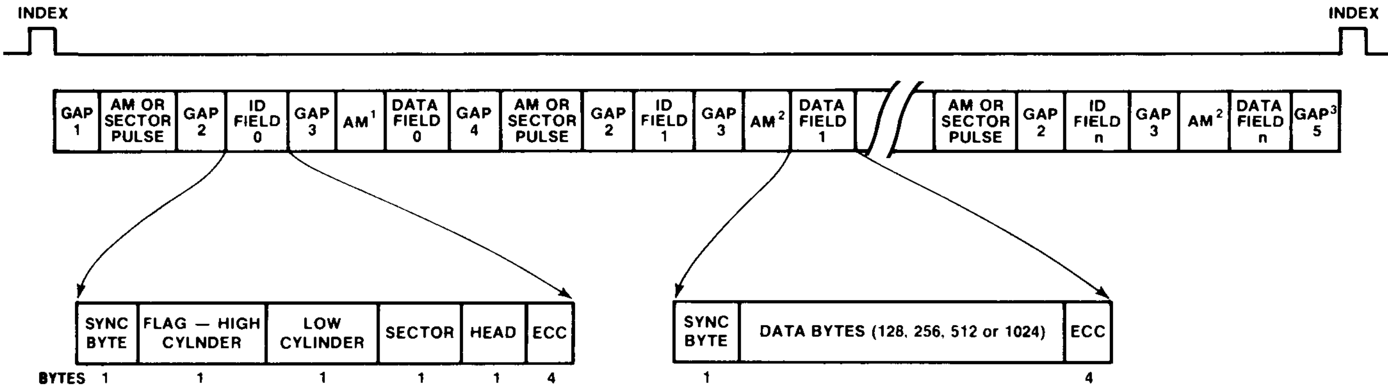


Figure A-1. Recommended Sector Format

The four bytes to be calculated ($n_Ø$, n_1, n_2, n_3) are shown below:

- n_1 If the drive is hard-sectored (that is, the drive sends sector pulses), byte 1 will be Ø. If the drive is soft-sectored (that is, address marks are on the disk), byte 1 is calculated as follows:

$$n_1 = \frac{24 + P2 + PAD}{2}$$

Where P2 = preamble 2 and consists of bytes of Ø's for the purpose of synchronizing the phase lock loop; and PAD = remaining bytes before sector end, which provides a guard zone between sectors to account for speed and clock tolerances.

- n_2 For hard-sectored drives, n_2 is calculated as follows:

  $$n_2 = \frac{P1 - 5}{2}$$

  For soft-sectored drives, n_2 is calculated as follows:

  $$n_2 = \frac{P1 - 2}{2}$$

- n_3 For hard- and soft-sectored drives:

  $$n_3 = \frac{P2 - 14}{2}$$

- $n_\emptyset$ The total overhead per sector $n_\emptyset$ is thus:

Soft Sectored Drive	Hard Sectored Drive
$n_\emptyset = 2n_2 + 2n_3 + 2\emptyset$	$n_\emptyset = 2n_2 + 2n_3 + 23$

The user can easily calculate the number of sectors per track for a given number of bytes per sector data size.

$$\frac{\text{number of sectors}}{\text{track}} = \frac{\text{number of bytes per track}}{\frac{\text{number of bytes}}{\text{sector}} + n_\emptyset}$$

Refer to Table 2-4 (the jumper configuration table) for the correct configuration depending on whether the drive is soft-sectored or hard-sectored.

***

B.1 INTRODUCTION

The information contained in this appendix is provided to illustrate various methods of implementing data transfers between one or more host CPU's and the iSBC 215G board. The flow charts illustrate the handshake procedures required between a host CPU and the board. User sequences are shown for both single and multi-user processing environments. A sequence for initiating overlapped seeks is also given.

The program listing provides an example program that a host CPU would run to direct data transfer between the host and the iSBC 215G board. The program is written in MCS-86 Macro Assembler language, and illustrates the data structure that the iSBC 215G board requires and shows a few simple disk operations drivers.

B.2 SINGLE USER SEQUENCE

The flow chart in Figure B-1 shows the handshake sequence between a single host CPU and the iSBC 215G board for basic data transfer operations (with no overlapping seeks). Note that communication between the host and the board is through the status semaphore and operation system bytes of the controller invocation block.

B.3 SINGLE USER SEQUENCE WITH OVERLAPPING SEEKS

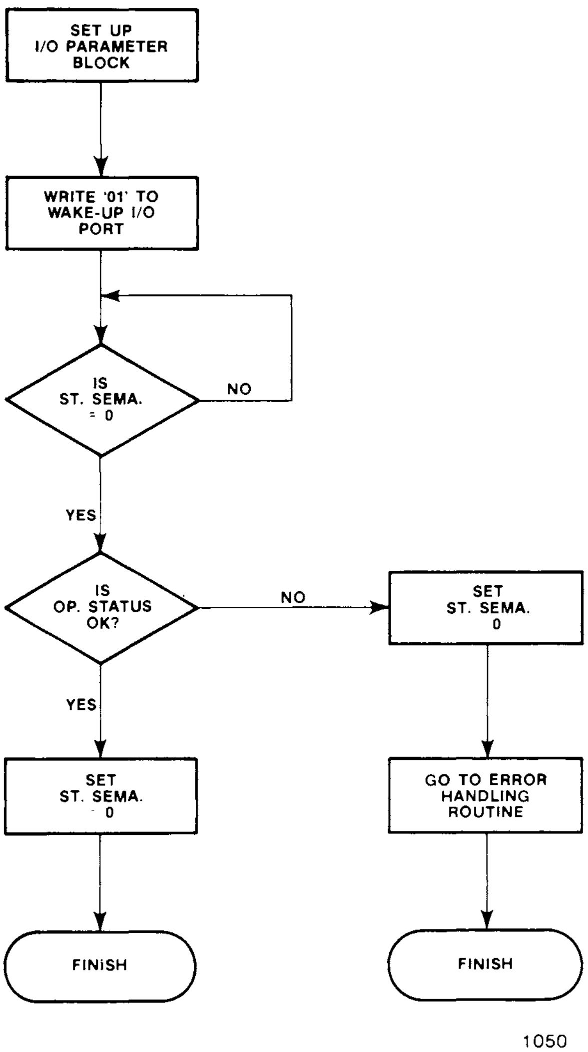
The flow chart in Figure B-2 shows the handshake sequence between a single host CPU and the iSBC 215G board for data transfer operations that user overlapping seeks.

B.4 MULTI-USER SEQUENCE

The flow chart in Figure B-3 shows the handshake sequence between a host CPU and the iSBC 215G board when more than one CPU is transferring data between the disk drives through the same board (multi-processor environment). Note that in this case the command semaphore byte in the controller invocation block is also used. Overlapping seeks in a multi-processor environment are implemented the same as in single processor environments.

B.5 EXAMPLE HOST PROCESSOR DISK CONTROL PROGRAM

The following program example is for a single user environment. Some of the techniques illustrated in the flow charts in this appendix are implemented in this program, but not all.



Set up command and parameters for desired data transfer operation.

Initiate data transfer operation.

Check to see if controller has completed operation (STATUS SEMAPHORE byte is non-zero).

Check OPERATION STATUS byte to see if operation was completed without error.

Check Error Status buffer and process results.

Figure B-1. Flow Chart for Single User Handshake Sequence Without Overlapping Seeks

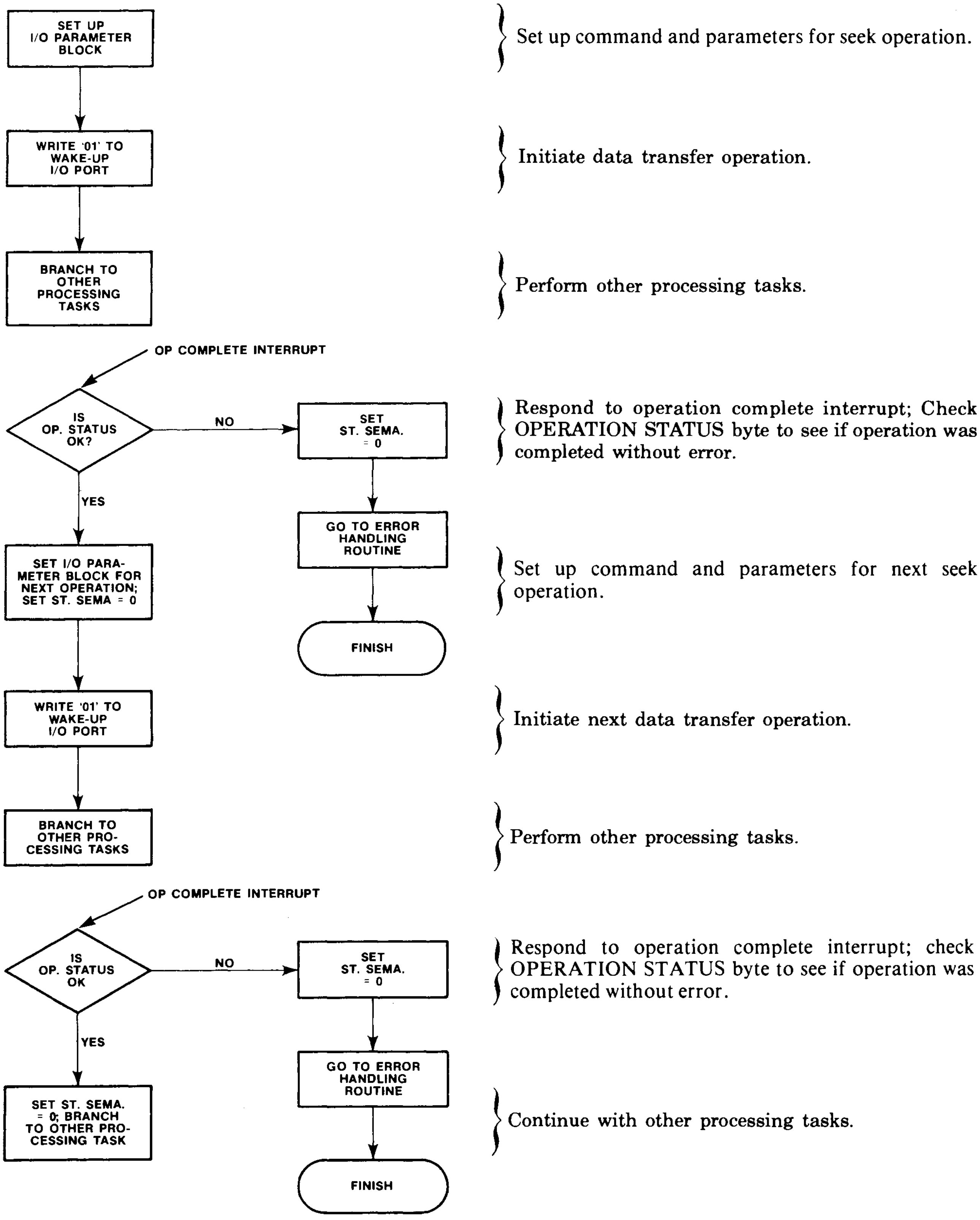
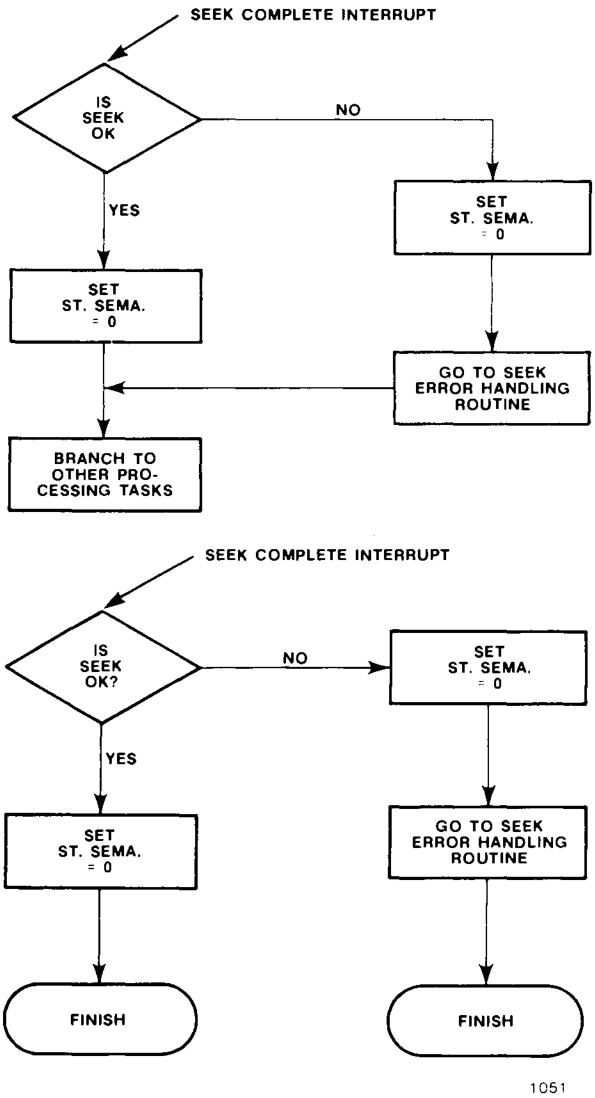


Figure B-2. Flow Chart for Single User Handshake Sequence With Overlapping Seeks



Respond to seek complete interrupt for first operation; check status buffer to determine if seek operation was completed without error.

Continue with other processing tasks.

Respond to seek complete interrupt for second operation; check status buffer to determine if seek operation was completed without error.

Figure B-2. Flow Chart for Single User Handshake Sequence With Overlapping Seeks (continued)

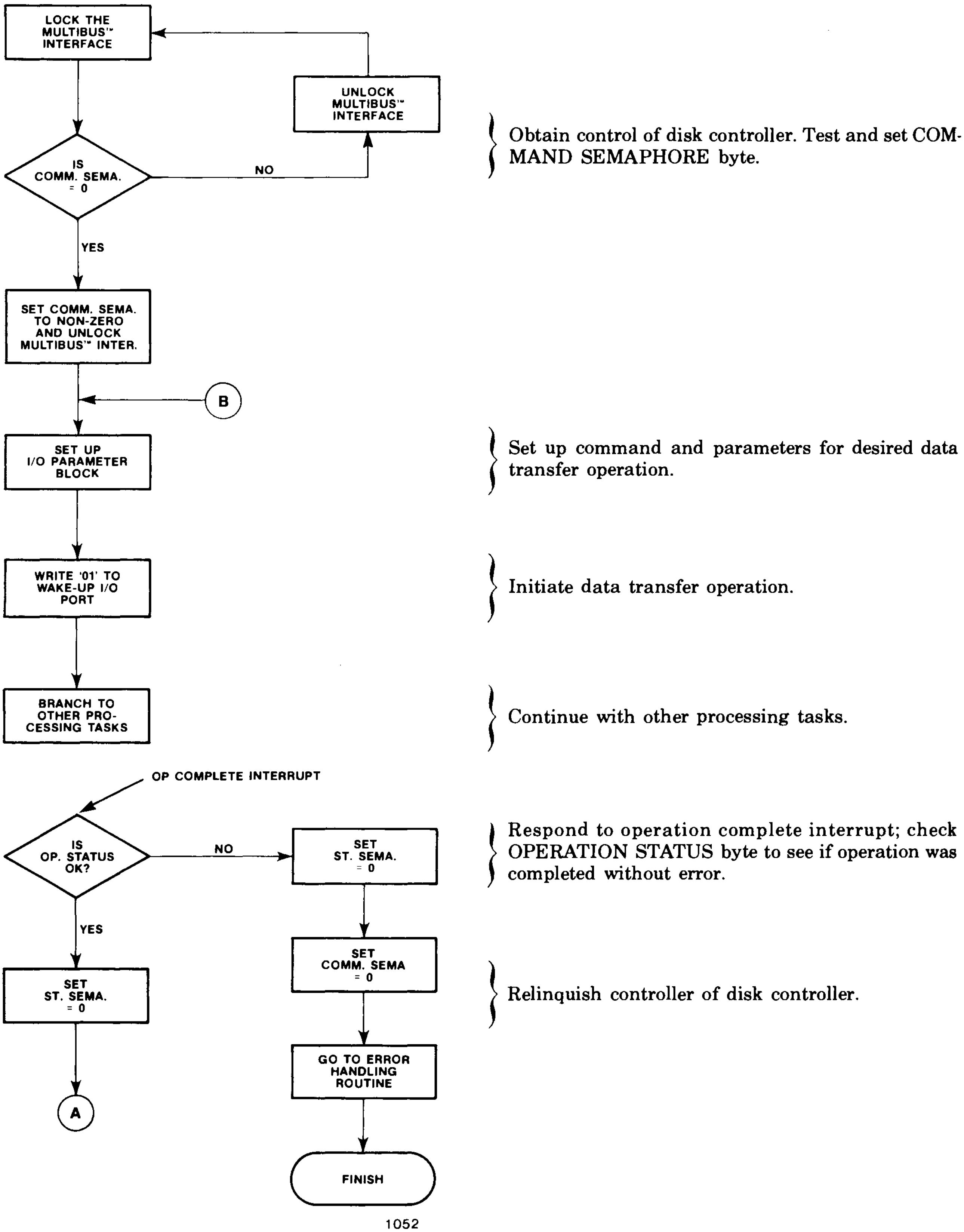


Figure B-3. Flow Chart for Multi-User Handshake Sequence

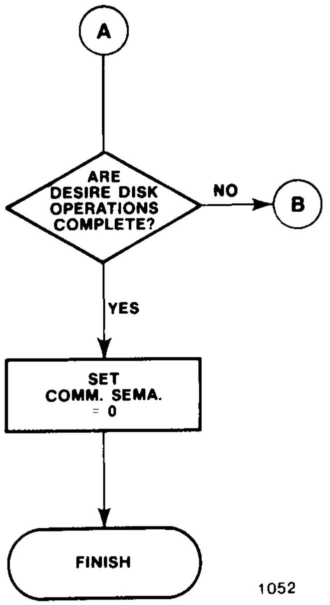


Figure B-3. Flow Chart for Multi-User Handshake Sequence (continued)

HANDSHAKE SEQUENCES AND EXAMPLE HOST PROCESSOR DISK CONTROL PROGRAM

```
MCS-86 MACRO ASSEMBLER    iSBC 215 8" WINCHESTER DISK CONTROLLER PROGRAMMING EXAMPLE


ISIS-II MCS-86 MACRO ASSEMBLER V2.1 ASSEMBLY OF MODULE EXMPRG
OBJECT MODULE PLACED IN :F1:EXMPRG.OBJ
ASSEMBLER INVOKED BY:  ASM86 :F1:EXMPRG.MMD DATE(10/27/80) XREF DEBUG


LOC  OBJ                        SOURCE

                                $PAGELENGTH(85) PAGEWIDTH(115) TITLE(iSBC 215 8" WINCHESTER DISK CONTROLLER PROG
                                RAMMING EXAMPLE) XREF
                                ;
                                ; ######################################################################
                                ; ##                                                                  ##
                                ; ##      iSBC 215  DISK CONTROLLER PROGRAMMING EXAMPLE    ##
                                ; ##                                                                  ##
                                ; ######################################################################
                                ;
                                ; THIS PROGRAM ILLUSTRATES THE DATA STRUCTURES REQUIRED BY THE iSBC 215
                                ; DISK CONTROLLER.  A FEW SIMPLE DISK OPERATION DRIVERS ARE ALSO SHOWN.
                                ;
                                ; THE HARDWARE CONFIGURATION SUPPORTED IS:
                                ;
                                ;        1.  iSBC 86/12A HOST CPU
                                ;        2.  20 BIT SYSTEM MEMORY ADDRESS WIDTH
                                ;        3.  16 BIT SYSTEM DATA BUS WIDTH
                                ;        4.  16 BIT SYSTEM I/O ADDRESS WIDTH
                                ;        5.  iSBC 215
                                ;                a.  WAKE UP ADDRESS ( WUA ) AT I/O PORT 0635H
                                ;                b.  INTERRUPT 5
                                ;                c.  -12 VOLTS INPUT
                                ;                d.  RELINQUISH BUS CONTROL ON ANY REQUEST
                                ;
                                ; FOR (2), PROGRAMMING OF DATA TRANSFERS MUST TAKE THIS INTO ACCOUNT,e.g. THERE
                                ;          IS NO WRAPAROUND IN SEGMENTS IF MORE THAN 64K BYTES ARE TRANSFERRED.
                                ;
                                ; iSBC 215 SWITCH AND JUMPER SETTINGS:
                                ;
                                ; FOR (3), SWITCH S2-1 IS CLOSED.
                                ; FOR (4), SWITCH S2-2 IS CLOSED.
                                ; FOR (5a), SWITCHES S1-6,S1-7,S2-5,S2-6,S2-8, AND S2-10 ARE CLOSED, THE
                                ;          REMAINING ADDRESS SELECT SWITCHES ARE OPEN.
                                ; FOR (5b), W19-C CONNECTS TO W19-5; INTERRUPT VECTORS MUST BE SET UP PROPERLY.
                                ; FOR (5c), W21-1 CONNECTS TO W21-3
                                ; FOR (5d), W2-1 CONNECTS TO W2-2.
                                ;
                          +1    $INCLUDE(:F1:COMBLK.MMD)
=1                        +1    $EJECT TITLE(iSBC 215 COMMUNICATION BLOCKS)
```

HANDSHAKE SEQUENCES AND EXAMPLE HOST PROCESSOR DISK CONTROL PROGRAM

```
MCS-86 MACRO ASSEMBLER    iSBC 215 COMMUNICATION BLOCKS

LOC  OBJ                           SOURCE

                        =1         ; --------------------------------
                        =1         ; |                              |
                        =1         ; |      COMMUNICATION BLOCKS    |
                        =1         ; |                              |
                        =1         ; --------------------------------
                        =1         ;
                        =1         ; ============
                        =1         ;   I.  SCB
                        =1         ; ============
                        =1         ;
                        =1         ;       THE SCB TELLS THE 8089 ON THE iSBC 215 THE WIDTH OF THE 8089's LOCAL
                        =1         ;       BUS AND POINTS TO THE CCB.
                        =1         ;
                        =1         ;       *****************************************************************
                        =1         ;       *  THE MEMORY ADDRESS OF THE SCB IS EQUAL TO THE I/O WAKE-UP ADDRESS  *
                        =1         ;       *                ( WUA ) OF THE iSBC 215 MULTIPLIED BY 16.           *
                        =1         ;       *****************************************************************
                        =1         ;
  0635                  =1                 WUA      EQU      0635H            ; WAKE-UP ADDRESS I/O PORT NUMBER
                        =1         ;
----                    =1         SCBSEG  SEGMENT AT WUA                     ; PUTS SCB AT ADDRESS 06350H
                        =1         ;
0000                    =1         SCB     LABEL    FAR
0000 01                 =1         SOC     DB       01H                       ; TELL 8089 IT IS ON A 16 BIT LOCAL BUS
0001 00                 =1                 DB       00H                       ; RESERVED
0002 0000----         R =1         CCBPTR  DD       CCB                       ; POINTER (SEGMENT + OFFSET) TO CCB
                        =1         ;
----                    =1         SCBSEG  ENDS
                        =1         ;
                        =1         ; =============
                        =1         ;   II.  CCB
                        =1         ; =============
                        =1         ;
                        =1         ;       THIS BLOCK CONTAINS THE CONTROL BYTES, BUSY FLAGS, AND POINTERS TO THE
                        =1         ;       STARTING ADDRESSES OF THE CHANNEL PROGRAMS FOR THE 8089.
                        =1         ;
----                    =1         CCBSEG  SEGMENT                            ; CCB MUST BE CONTIGUOUS
                        =1         ;
0000                    =1         CCB     LABEL    FAR
0000 01                 =1         CCW1    DB       01H                       ; START CH. 1 PROGRAM IN LOCAL MEMORY
0001 00                 =1         BSYFLG1 DB       00H                       ; CH. 1 BUSY FLAG
0002 0400----         R =1         CH1PTR  DD       CH1PC                     ; POINTER TO FIFTH BYTE OF CIB, WHICH
                        =1                                                    ;  CONTAINS STARTING ADDRESS OF CH. 1
                        =1                                                    ;  FIRMWARE PROGRAM
0006 0000               =1                 DW       0000H                     ; RESERVED
0008 01                 =1         CCW2    DB       01H                       ; START CH. 2 PROGRAM IN LOCAL MEMORY
0009 00                 =1         BSYFLG2 DB       00H                       ; CH. 2 BUSY FLAG
000A 0E00----         R =1         CH2PTR  DD       CH2PC                     ; POINTER TO LAST WORD OF CCB, WHICH
                        =1                                                    ;  CONTAINS STARTING ADDRESS OF CH. 2
                        =1                                                    ;  FIRMWARE PROGRAM
000E                    =1         CH2PC   LABEL    FAR
000E 0400               =1                 DW       0004H                     ; STARTING ADDRESS OF CH. 2 PROGRAM
                        =1         ;
----                    =1         CCBSEG  ENDS
                        =1         ;
                        =1     +1  $EJECT
```

```
MCS-86 MACRO ASSEMBLER    iSBC 215 COMMUNICATION BLOCKS

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 215 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 215
0001 00               =1   107     OPSTS   DB      00H                      ; CIB STATUS BYTE IS USED BY iSBC 215
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 215 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 0000             =1   130     DEVCOD  DW      0000H                    ; DEVICE CODE (0H-WINCHESTER 01H-FLOPPY)
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(:F1:INITBL.MMD)
                      =1   145 +1  $EJECT TITLE(DISK DRIVE INITIALIZATION TABLES)
```

```
MCS-86 MACRO ASSEMBLER    DISK DRIVE INITIALIZATION TABLES

LOC  OBJ                  SOURCE

                     =1   ; ---------------------------------------------------
                     =1   ; |                                                   |
                     =1   ; |    DISK DRIVE INITIALIZATION PARAMETER TABLES     |
                     =1   ; |                                                   |
                     =1   ; ---------------------------------------------------
                     =1   ;
                     =1   ;      THIS SEGMENT CONTAINS THE DRIVE CONFIGURATION DATA TABLES THAT ARE USED
                     =1   ;      BY THE INITIALIZATION ROUTINE.  THEY MUST BE MODIFIED TO REFLECT THE
                     =1   ;      PARTICULAR DRIVES BEING USED WITH THE iSBC 215 DISK CONTROLLER.
                     =1   ;
                     =1   ; - IF A DRIVE IS NOT PRESENT, ITS INITIALIZATION TABLE MUST BE ALL ZEROES.
                     =1   ;
                     =1   ;   ....................................................
                     =1   ;   |          8 " WINCHESTER HARD DISK DRIVES             |
                     =1   ;   ....................................................
                     =1   ;   | BYTES PER SECTOR  |   MAXIMUM SECTORS PER TRACK   |
                     =1   ;   |-------------------+-------------------------------|
                     =1   ;   |       128         |            54                 |
                     =1   ;   |       256         |            31                 |
                     =1   ;   |       512         |            17                 |
                     =1   ;   |      1024         |            9                  |
                     =1   ;   ....................................................
                     =1   ;------------------------------------------------------------------------
                     =1   ;
----                 =1   INITBLSEG       SEGMENT
                     =1   ;
                     =1   ; DRIVE #0 ---SHUGART MODEL SA1004 (10.6 MEGABYTES STORAGE)
                     =1   ;
0000 0001            =1           DW      256            ; NUMBER OF CYLINDERS
0002 04              =1           DB      4              ; NUMBER OF FIXED READ/WRITE SURFACES
0003 00              =1           DB      0              ; NUMBER OF REMOVABLE R/W SURFACES
0004 1F              =1           DB      31             ; NUMBER OF SECTORS PER TRACK
0005 0001            =1           DW      256            ; NUMBER OF BYTES PER SECTOR
0007 05              =1           DB      5              ; NUMBER OF ALTERNATE CYLINDERS
                     =1   ;
                     =1   ; DRIVE #1 ---SHUGART MODEL SA1002 (5.3 MEGABYTES STORAGE)
                     =1   ;
0008 0001            =1           DW      256            ; NUMBER OF CYLINDERS
000A 02              =1           DB      2              ; NUMBER OF FIXED READ/WRITE SURFACES
000B 00              =1           DB      0              ; NUMBER OF REMOVABLE R/W SURFACES
000C 11              =1           DB      17             ; NUMBER OF SECTORS PER TRACK
000D 0002            =1           DW      512            ; NUMBER OF BYTES PER SECTOR
000F 05              =1           DB      5              ; NUMBER OF ALTERNATE CYLINDERS
                     =1   ;
                     =1   ; DRIVE #2 --- NONEXISTENT
                     =1   ;
0010 0000            =1           DW      0000H          ; NUMBER OF CYLINDERS
0012 00              =1           DB      00H            ; NUMBER OF FIXED READ/WRITE SURFACES
0013 00              =1           DB      00H            ; NUMBER OF REMOVABLE R/W SURFACES
0014 00              =1           DB      00H            ; NUMBER OF SECTORS PER TRACK
0015 0000            =1           DW      0000H          ; NUMBER OF BYTES PER SECTOR
0017 00              =1           DB      00H            ; NUMBER OF ALTERNATE CYLINDERS
                     =1   ;
                     =1   ; DRIVE #3 --- NONEXISTENT
                     =1   ;
0018 0000            =1           DW      0000H          ; NUMBER OF CYLINDERS
001A 00              =1           DB      00H            ; NUMBER OF FIXED READ/WRITE SURFACES
001B 00              =1           DB      00H            ; NUMBER OF REMOVABLE R/W SURFACES
001C 00              =1           DB      00H            ; NUMBER OF SECTORS PER TRACK
001D 0000            =1           DW      0000H          ; NUMBER OF BYTES PER SECTOR
001F 00              =1           DB      00H            ; NUMBER OF ALTERNATE CYLINDERS
                     =1   ;
                     =1 +1 $EJECT
```

```
MCS-86 MACRO ASSEMBLER     DISK DRIVE INITIALIZATION TABLES

LOC  OBJ                       SOURCE

                    =1         ;
                    =1         ;     .......................................................
                    =1         ;     |             8"  FLEXIBLE  DISK DRIVES                   |
                    =1         ;     .......................................................
                    =1         ;     | BYTES PER SECTOR    |    MAXIMUM SECTORS PER TRACK    |
                    =1         ;     |---------------------+-----------------------------------|
                    =1         ;     |        128          |            26 (FM)                |
                    =1         ;     |        256          |            26 (MFM)               |
                    =1         ;     |        512          |            15 (MFM)               |
                    =1         ;     |       1024          |            8  (MFM)               |
                    =1         ;     .......................................................
                    =1         ;----------------------------------------------------------------------------
                    =1         ;
                    =1         ; DRIVE #0 ---SHUGART MODEL 850     (1.0 MEGABYTES STORAGE)
                    =1         ;
0020 4D00           =1                 DW      77                   ; NUMBER OF CYLINDERS
0022 00             =1                 DB      0                    ; NUMBER OF FIXED READ/WRITE SURFACES
0023 02             =1                 DB      2                    ; NUMBER OF REMOVABLE R/W SURFACES
0024 1A             =1                 DB      26                   ; NUMBER OF SECTORS PER TRACK
0025 0001           =1                 DW      256                  ; NUMBER OF BYTES PER SECTOR
0027 01             =1                 DB      01                   ; MFM(1) OR FM(0) RECORDING MODE
                    =1         ;
                    =1         ; DRIVE #1 ---SHUGART MODEL 850    (1.0) MEGABYTES STORAGE)
                    =1         ;
0028 4D00           =1                 DW      77                   ; NUMBER OF CYLINDERS
002A 00             =1                 DB      0                    ; NUMBER OF FIXED READ/WRITE SURFACES
002B 02             =1                 DB      2                    ; NUMBER OF REMOVABLE R/W SURFACES
002C 1A             =1                 DB      26                   ; NUMBER OF SECTORS PER TRACK
002D 8000           =1                 DW      128                  ; NUMBER OF BYTES PER SECTOR
002F 00             =1                 DB      00                   ; MFM(1) OR FM(0) RECORDING MODE
                    =1         ;
                    =1         ; DRIVE #2 --- NONEXISTENT
                    =1         ;
0030 0000           =1                 DW      0000H                ; NUMBER OF CYLINDERS
0032 00             =1                 DB      00H                  ; NUMBER OF FIXED READ/WRITE SURFACES
0033 00             =1                 DB      00H                  ; NUMBER OF REMOVABLE R/W SURFACES
0034 00             =1                 DB      00H                  ; NUMBER OF SECTORS PER TRACK
0035 0000           =1                 DW      0000H                ; NUMBER OF BYTES PER SECTOR
0037 00             =1                 DB      00H                  ; MFM(1) OR FM(0) RECORDING MODE
                    =1         ;
                    =1         ; DRIVE #3 --- NONEXISTENT
                    =1         ;
0038 0000           =1                 DW      0000H                ; NUMBER OF CYLINDERS
003A 00             =1                 DB      00H                  ; NUMBER OF FIXED READ/WRITE SURFACES
003B 00             =1                 DB      00H                  ; NUMBER OF REMOVABLE R/W SURFACES
003C 00             =1                 DB      00H                  ; NUMBER OF SECTORS PER TRACK
003D 0000           =1                 DW      0000H                ; NUMBER OF BYTES PER SECTOR
                    =1         ;       DB      00H                  ; MFM(1) OR FM(0) RECORDING MODE
                    =1         ;
----                =1         INITBLSEG        ENDS
                               ;
                          +1   $INCLUDE(:F1:DATSEG.MMD)
                    =1    +1   $EJECT TITLE(DATA SEGMENT)
```

```
MCS-86 MACRO ASSEMBLER    DATA SEGMENT

LOC  OBJ                 LINE     SOURCE

                    =1            ; -------------------------
                    =1            ; |                         |
                    =1            ; |      DATA SEGMENT       |
                    =1            ; |                         |
                    =1            ; -------------------------
                    =1            ;
----                =1            DATASEG SEGMENT
                    =1            ;
                    =1            ;       THIS SEGMENT CONTAINS VARIOUS DATA THAT ARE USED BY THE iSBC 215 DRIVER
                    =1            ;       SOFTWARE.
                    =1            ;
                    =1            ; - THE FLAGS ARE SET BY THE INTERRUPT SERVICE ROUTINE, AND ARE COPIES OF THE
                    =1            ;       CIB STATUS POSTED BY THE iSBC 215.  THE ROUTINES THAT USE THE FLAGS ARE
                    =1            ;       RESPONSIBLE FOR CLEARING THEM AFTER USE.
                    =1            ;---------------------------------------------------------------------------
                    =1            ;
                    =1                    PUBLIC  OPCMP,SKCMP,PKCHG,ERRSTS
                    =1            ;
                    =1            ;       OPERATION COMPLETE FLAGS
                    =1            ;
0000                =1            OPCMP   LABEL   BYTE
0000 00             =1            OPCMP0  DB      00H                 ; OPERATION COMPLETE ON UNIT 0
0001 00             =1            OPCMP1  DB      00H                 ; OPERATION COMPLETE ON UNIT 1
0002 00             =1            OPCMP2  DB      00H                 ; OPERATION COMPLETE ON UNIT 2
0003 00             =1            OPCMP3  DB      00H                 ; OPERATION COMPLETE ON UNIT 3
                    =1            ;
                    =1            ;       SEEK COMPLETE FLAGS
                    =1            ;
0004                =1            SKCMP   LABEL   BYTE
0004 00             =1            SKCMP0  DB      00H                 ; SEEK COMPLETE ON UNIT 0
0005 00             =1            SKCMP1  DB      00H                 ; SEEK COMPLETE ON UNIT 1
0006 00             =1            SKCMP2  DB      00H                 ; SEEK COMPLETE ON UNIT 2
0007 00             =1            SKCMP3  DB      00H                 ; SEEK COMPLETE ON UNIT 3
                    =1            ;
                    =1            ;       PACK CHANGE FLAGS
                    =1            ;
0008                =1            PKCHG   LABEL   BYTE
0008 00             =1            PKCHG0  DB      00H                 ; PACK CHANGE ON UNIT 0
0009 00             =1            PKCHG1  DB      00H                 ; PACK CHANGE ON UNIT 1
000A 00             =1            PKCHG2  DB      00H                 ; PACK CHANGE ON UNIT 2
000B 00             =1            PKCHG3  DB      00H                 ; PACK CHANGE ON UNIT 3
                    =1            ;
                    =1            ;       ERROR STATUS BLOCK
                    =1            ;       (LOADED FROM CONTROLLER BY ERROR HANDLER)
                    =1            ;
000C 0000           =1            ERRSTS  DW      0000H               ; ERROR STATUS WORD
000E 00             =1            SFERST  DB      00H                 ; SOFT ERROR STATUS BYTE
000F 0000           =1            DESCYL  DW      0000H               ; DESIRED CYLINDER
0011 00             =1            DESHD   DB      00H                 ; DESIRED HEAD
0012 00             =1            DESSEC  DB      00H                 ; DESIRED SECTOR
0013 0000           =1            ACTCYL  DW      0000H               ; ACTUAL CYLINDER + FLAG BITS
0015 00             =1            ACTHD   DB      00H                 ; ACTUAL HEAD
0016 00             =1            ACTSEC  DB      00H                 ; ACTUAL SECTOR
0017 00             =1            NMRTRY  DB      00H                 ; NUMBER OF RETRIES MADE
                    =1            ;
                    =1            ;       LAST OPERATION COMPLETE BYTE
                    =1            ;       (COPIED FROM CIB BY WAIT215)
                    =1            ;
0018 00             =1            LSTSTS  DB      00H
                    =1            ;
0019 90             =1                    EVEN
                    =1            ;
001A                =1            ENDDAT  LABEL   FAR                 ; END OF DATA SEGMENT
                    =1            ;
----                =1            DATASEG ENDS
                                  ;
                             +1   $INCLUDE(:F1:USER.MMD)
                    =1       +1   $EJECT TITLE(SYSTEM DEPENDENT INITIALIZATION)
```

```
MCS-86 MACRO ASSEMBLER    SYSTEM DEPENDENT INITIALIZATION

LOC  OBJ                 LINE     SOURCE

                    =1    330     ; -----------------------------------------
                    =1    331     ; |                                       |
                    =1    332     ; |      SYSTEM DEPENDENT INITIALIZATION   |
                    =1    333     ; |                                       |
                    =1    334     ; -----------------------------------------
                    =1    335     ;
                    =1    336     ;        THIS ROUTINE SETS UP THE INTERRUPT VECTOR FOR AN iSBC 86/12A CPU
                    =1    337     ;        RUNNING UNDER THE iSBC 957A INTERFACE/EXECUTION PACKAGE.
                    =1    338     ;
                    =1    339     ; - THE 8259 INTERRUPT CONTROLLER AND OTHER INITIALZATIONS ARE PERFORMED
                    =1    340     ;        BY THE iSBC 957A FIRMWARE.
                    =1    341     ; ---------------------------------------------------------------------------
                    =1    342     ;
                    =1    343     ;        ==========================
                    =1    344     ;        INTERRUPT VECTOR DEFINITION
                    =1    345     ;        ==========================
                    =1    346     ;
  0005              =1    347             INTRPT  EQU     5               ; iSBC 220 INTERRUPT NUMBER
                    =1    348     ;
----                =1    349     SEG0000 SEGMENT AT 0000H                ; INTERRUPT VECTORS ARE FROM ABSOLUTE
                    =1    350                                             ;    ADDRESSES 00000H TO 00FF0H
                    =1    351     ;
0094                =1    352             ORG     80H + 4*INTRPT          ; LOCATION OF INTERRUPT VECTOR WITH
                    =1    353                                             ;    iSBC 957A FIRMWARE
0094 0000           =1    354     INTRIP  DW      0000H                   ; - INSTRUCTION POINTER
0096 0000           =1    355     INTRCS  DW      0000H                   ; - CODE SEGMENT
                    =1    356     ;
----                =1    357     SEG0000 ENDS
                    =1    358     ;
                    =1    359     ;        ================
                    =1    360     ;        STACK ALLOCATION
                    =1    361     ;        ================
                    =1    362     ;
----                =1    363     STACK   SEGMENT                         ; STACK SEGMENT
                    =1    364     ;
0000 (64            =1    365             DB      64 DUP(00H)             ; ALLOW 64 BYTES FOR STACK
     00
     )
                    =1    366     ;
0040                =1    367     ENDSTK  LABEL   FAR
                    =1    368     ;
----                =1    369     STACK   ENDS
                    =1    370     ;
                    =1    371     ;        ========================================
                    =1    372     ;        STACK AND INTERRUPT CONFIGURATION ROUTINE
                    =1    373     ;        ========================================
                    =1    374     ;
----                =1    375     USERSEG SEGMENT
                    =1    376     ;
                    =1    377             PUBLIC  CONFIG
                    =1    378             ASSUME  DS:SEG0000
                    =1    379     ;
0000                =1    380     CONFIG  PROC FAR
                    =1    381     ;
0000 FA             =1    382             CLI                             ; DISABLE INTERRUPTS WHILE SETTING UP
0001 B8----       R =1    383             MOV     AX,STACK                ;;; SET UP STACK
0004 8ED0           =1    384             MOV     SS,AX                   ;;;
0006 BC4000         =1    385             MOV     SP,OFFSET ENDSTK        ;;;
0009 B80000         =1    386             MOV     AX,0000H                ;;; GET POINTER TO SEGMENT 0000H
000C 8ED8           =1    387             MOV     DS,AX                   ;;;
000E C70694003D02   =1    388             MOV     INTRIP,OFFSET INT215    ;;; SET UP INTERRUPT VECTOR
0014 C7069600---- R =1    389             MOV     INTRCS,SEG INT215       ;;;
001A E4C2           =1    390             IN      AL,0C2H                 ;;; INPUT INTERRUPT MASK FROM 8259
001C 24DF           =1    391             AND     AL,11011111B            ;;; ENABLE INTERRUPT 5
001E E6C2           =1    392             OUT     0C2H,AL                 ;;; WRITE NEW MASK OUT TO 8259
0020 FB             =1    393             STI                             ;;; ENABLE INTERRUPTS
0021 CC             =1    394             INT     3                       ;;; GO TO MONITOR
                    =1    395     ;
                    =1    396     CONFIG  ENDP
                    =1    397     ;
----                =1    398     USERSEG ENDS
                          399     ;
----                      400     SBC215DRIVER    SEGMENT
                          401     ;
                          402             ASSUME CS:SBC215DRIVER
                          403     ;
                          404 +1  $INCLUDE(:F1:RESET.MMD)
                    =1    405 +1  $EJECT TITLE(CONTROLLER RESET ROUTINE)
```

```
MCS-86 MACRO ASSEMBLER    CONTROLLER RESET ROUTINE

LOC  OBJ                  LINE      SOURCE

                      =1   406      ; ------------------------------------
                      =1   407      ; |                                  |
                      =1   408      ; |      CONTROLLER RESET ROUTINE    |
                      =1   409      ; |                                  |
                      =1   410      ; ------------------------------------
                      =1   411      ;
                      =1   412      ;       RES215 SETS UP THE COMMUNICATION BLOCKS FOR THE iSBC 215, LINKS THEM
                      =1   413      ;       TOGETHER AND GIVES A RESET, CLEAR RESET, CHANNEL ATTENTION SEQUENCE TO
                      =1   414      ;       THE CONTROLLER.  THIS CAUSES THE 8089 ON THE CONTROLLER TO SET UP ITS
                      =1   415      ;       INTERNAL POINTER TO THE CCB BY THREADING DOWN THE LINKS STARTING WITH
                      =1   416      ;       THE SWITCHES ON THE CONTROLLER.  SUBSEQUENT CA's WILL CAUSE THE 8089 TO
                      =1   417      ;       FETCH ITS POINTERS STARTING AT THE CCB.
                      =1   418      ;
                      =1   419      ; - IF THE CH. 1 BUSY FLAG IS NOT CLEARED WITHIN A "REASONABLE" AMOUNT OF TIME,
                      =1   420      ;       THEN THE iSBC 215 IS PROBABLY NOT RESPONDING TO THE CHANNEL ATTENTION.
                      =1   421      ;       ON THE CONTROLLER: CHECK SWITCH SETTINGS; VOLTAGES; RESET, CLEAR RESET,
                      =1   422      ;       CHANNEL ATTENTION SIGNALS; READY INPUT TO 8089; 8089 STATUS LINES; R/W
                      =1   423      ;       STROBES.
                      =1   424      ;
                      =1   425      ; - THE SYSTEM INTERRUPT LOGIC AND VECTORS FOR THE CONTROLLER ARE ASSUMED TO BE
                      =1   426      ;       CONFIGURED BY AN EXTERNAL PROGRAM.
                      =1   427      ;
                      =1   428      ; INPUT DATA:
                      =1   429      ;       NONE
                      =1   430      ;
                      =1   431      ; OUTPUT DATA:
                      =1   432      ;       CARRY FLAG:      = 0 IF RESET OKAY
                      =1   433      ;                        = 1 IF CH. 1 BUSY FLAG NOT RESET (NOT RESPONDING)
                      =1   434      ; --------------------------------------------------------------------------
                      =1   435      ;
                      =1   436              PUBLIC  RES215
                      =1   437      ;
0000                  =1   438      RES215  PROC    FAR
                      =1   439      ;
0000 50               =1   440              PUSH    AX                          ; SAVE REGISTERS
0001 53               =1   441              PUSH    BX
0002 51               =1   442              PUSH    CX
0003 52               =1   443              PUSH    DX
0004 1E               =1   444              PUSH    DS
                      =1   445      ;
                      =1   446      ;       SET UP LINKS BETWEEN COMMUNICATION BLOCKS
                      =1   447      ;
                      =1   448      ; SCB
                      =1   449      ;
                      =1   450              ASSUME  DS:SCBSEG
0005 B83506           =1   451              MOV     AX,SCBSEG                   ; GET POINTER TO SCB
0008 8ED8             =1   452              MOV     DS,AX
000A C70600000100     =1   453              MOV WORD PTR SOC,0001H              ; SET SOC BYTE AND CLEAR RESERVED BYTE
0010 C70602000000     =1   454              MOV WORD PTR CCBPTR,OFFSET CCB      ; SET POINTER TO CCB
0016 C7060400----   R =1   455              MOV WORD PTR CCBPTR+2,SEG CCB
                      =1   456      ;
                      =1   457      ; CCB
                      =1   458      ;
001C C5060200         =1   459              LDS     AX,CCBPTR                   ; GET POINTER TO CCB
                      =1   460              ASSUME  DS:CCBSEG
0020 C706000001FF     =1   461              MOV WORD PTR CCW1,0FF01H            ; SET CCW1 AND CH. 1 BUSY FLAG
0026 C70602000400     =1   462              MOV WORD PTR CH1PTR,OFFSET CH1PC; SET POINTER TO FIFTH BYTE OF CIB
002C C7060400----   R =1   463              MOV WORD PTR CH1PTR+2,SEG CH1PC ;    (HAS STARTING ADDRESS FOR CH. 1)
0032 C70608000100     =1   464              MOV WORD PTR CCW2,0001H             ; SET CCW2 AND CLEAR CH. 2 BUSY FLAG
0038 C7060A000E00     =1   465              MOV WORD PTR CH2PTR,OFFSET CH2PC; SET POINTER TO CH. 2 STARTING ADDRESS
003E C7060C00----   R =1   466              MOV WORD PTR CH2PTR+2,SEG CH2PC
0044 C7060E000400     =1   467              MOV WORD PTR CH2PC,0004H            ; SET CH. 2 STARTING ADDRESS
                      =1   468      ;
                      =1   469      ; CIB
                      =1   470      ;
                      =1   471              ASSUME  DS:CIBSEG
004A B8----         R =1   472              MOV     AX,CIBSEG                   ; GET POINTER TO CIB
004D 8ED8             =1   473              MOV     DS,AX
004F C70600000000     =1   474              MOV WORD PTR CIBCMD,0000H           ; CLEAR CIB COMMAND AND CIB STATUS BYTES
0055 C70602000000     =1   475              MOV WORD PTR CMDSEM,0000H           ;    ...AND SEMAPHORES
005B C70604000000     =1   476              MOV WORD PTR CH1PC,0000H            ; SET CH. 1 STARTING ADDRESS
0061 C70608000000     =1   477              MOV     IOPBOFF,OFFSET IOPB         ; SET IOPB POINTER
0067 C7060A00----   R =1   478              MOV     IOPBSG,SEG IOPB
                      =1   479      ;
                      =1   480 +1   $EJECT
```

```
MCS-86 MACRO ASSEMBLER    CONTROLLER RESET ROUTINE

LOC  OBJ                       SOURCE

                        =1     ;         CLEAR OUT DATA SEGMENT
                        =1     ;
                        =1               ASSUME  DS:DATASEG
006D B8----           R =1               MOV     AX,DATASEG              ; GET POINTER TO DATA SEGMENT
0070 8ED8               =1               MOV     DS,AX
0072 B90D00             =1               MOV     CX,(OFFSET ENDDAT)/2    ; GET COUNT (# WORDS IN DATA SEGMENT)
0075 BB0000             =1               MOV     BX,0000H                ; CLEAR INDEX REGISTER
0078 C7070000           =1     CLRLP:    MOV WORD PTR [BX],0000H         ; CLEAR NEXT WORD IN DATA SEGMENT
007C 43                 =1               INC     BX                      ; POINT TO NEXT WORD
007D 43                 =1               INC     BX
007E E0F8               =1               LOOPNE  CLRLP                   ; DONE?
                        =1                                               ; NO--CLEAR ANOTHER WORD
                        =1                                               ; YES--INITIALIZE COMMUNICATION LINKS
                        =1     ;
                        =1     ;         OUTPUT RESET/CLEAR RESET/CHANNEL ATTENTION TO CONTROLLER
                        =1     ;
0080 BA3506             =1               MOV     DX,WUA                  ; GET WAKE-UP I/O PORT ADDRESS
0083 B002               =1               MOV     AL,02H                  ; GET RESET COMMAND BYTE
0085 EE                 =1               OUT     DX,AL                   ; OUTPUT TO WAKE-UP I/O PORT
0086 B000               =1               MOV     AL,00H                  ; GET CLEAR RESET COMMAND BYTE
0088 EE                 =1               OUT     DX,AL                   ; OUTPUT TO WAKE-UP I/O PORT
0089 B001               =1               MOV     AL,01H                  ; GET CHANNEL ATTENTION COMMAND BYTE
008B EE                 =1               OUT     DX,AL                   ; OUTPUT  TO WAKE-UP I/O PORT
                        =1               ASSUME  DS:CCBSEG
008C B8----           R =1               MOV     AX,CCBSEG               ; GET POINTER TO CCB
008F 8ED8               =1               MOV     DS,AX
                        =1                                               ; (OTHER IMPLEMENTATIONS OF RES215 COULD
                        =1                                               ; INITIALIZE OTHER DEVICES WHILE THE
                        =1                                               ; iSBC 215 DOES ITS RESET SEQUENCE HERE)
0091 B90010             =1               MOV     CX,1000H                ; SET TIME-OUT COUNTER
0094 F8                 =1               CLC                             ; CLEAR CARRY FLAG
0095 F6060100FF         =1     RESLP:    TEST    BSYFLG1,00FFH           ; CHECK CH. 1 BUSY FLAG:
                        =1                                               ;    ZERO FLAG = BSYFLG1 & FFH
009A 7403               =1               JZ      RESDN                   ; BUSY FLAG CLEARED?
                        =1                                               ; YES--RETURN CARRY CLEAR
009C E0F7               =1               LOOPNE  RESLP                   ; NO--DECREMENT COUNTER
                        =1                                               ; IF CX = 0, THEN BSYFLG1 NEVER GOT
009E F9                 =1               STC                             ;    CLEARED, SO SET CARRY FLAG
009F 1F                 =1     RESDN:    POP     DS                      ; RESTORE REGISTERS
00A0 5A                 =1               POP     DX
00A1 59                 =1               POP     CX
00A2 5B                 =1               POP     BX
00A3 58                 =1               POP     AX
00A4 CB                 =1               RET                             ; RETURN
                        =1     ;
                        =1     RES215    ENDP
                               ;
                          +1   $INCLUDE(:F1:INITEX.MMD)
                        =1 +1  $EJECT TITLE(INITIALIZATION ROUTINE)
```

```
MCS-86 MACRO ASSEMBLER    INITIALIZATION ROUTINE

LOC  OBJ                        SOURCE

                         =1     ; -----------------------------------
                         =1     ; |                                 |
                         =1     ; |       INITIALIZATION ROUTINE    |
                         =1     ; |                                 |
                         =1     ; -----------------------------------
                         =1     ;
                         =1     ;       INIT215 INITIALIZES THE iSBC 215 CONTROLLER BY LOADING PERTINENT INFOR-
                         =1     ;       MATION ABOUT THE DISK DRIVE(S) ATTACHED.
                         =1     ;
                         =1     ; - IF A DRIVE THAT IS SPECIFIED AS PRESENT WILL NOT RESPOND, INIT215 RETURNS
                         =1     ;       IMMEDIATELY WITH THE CARRY FLAG SET.
                         =1     ;
                         =1     ; INPUT DATA:
                         =1     ;       DISK DRIVE INITIALIZATION TABLES, IN SEGMENT "INITBLSEG".
                         =1     ;
                         =1     ; OUTPUT DATA:
                         =1     ;       CARRY FLAG              = 0 IF CONTROLLER INITIALIZED SUCCESSFULLY
                         =1     ;                               = 1 IF INITIALIZATION ERROR
                         =1     ; ------------------------------------------------------------------------------
                         =1     ;
                         =1             PUBLIC  INIT215
                         =1             ASSUME  DS:IOPBSEG
                         =1     ;
00A5                     =1     INIT215 PROC    FAR
                         =1     ;
00A5 50                  =1             PUSH    AX                      ; SAVE REGISTERS
00A6 1E                  =1             PUSH    DS
00A7 B8----            R =1             MOV     AX,IOPBSEG              ; GET POINTER TO IOPB
00AA 8ED8                =1             MOV     DS,AX                   ; PUT IN DS REGISTER
00AC C70608000000        =1             MOV     DEVCOD,00H              ; WINCHESTER DRIVES INITIALIZED FIRST
00B2 C6060B0000          =1             MOV     FUNC,00H                ; SET IOPB FUNCTION BYTE = INITIALIZE
00B7 C7060C000000        =1             MOV     MODIFY,0000H            ; CLEAR MODIFIER (ENABLE RETRIES AND
                         =1                                             ;    INTERRUPT ON COMPLETION)
00BD C7061400----      R =1             MOV     BUFSEG,INITBLSEG        ; PUT INITIALIZATION TABLES' SEGMENT IN
                         =1                                             ;   IOPB DATA BUFFER POINTER
00C3 C7061200F8FF        =1             MOV     BUFOFF,-8               ; START INITIALIZE WITH UNIT 0
00C9 B000                =1             MOV     AL,00H                  ; CLEAR UNIT COUNTER
00CB 8306120008          =1     INITLP: ADD     BUFOFF,8                ; POINT TO NEXT DRIVE's INITIALIZE TABLE
00D0 A20A00              =1             MOV     UNIT,AL                 ; PUT UNIT INTO IOPB
00D3 E8EC00              =1             CALL    GO215                   ; DO INITIALIZE
                         =1                                             ;    (RETURNS CARRY FLAG SET OR CLEAR)
00D6 7214                =1             JC      INITDN                  ; UNIT INITIALIZED?
                         =1                                             ; NO--TERMINATE WITH CARRY BIT SET
00D8 40                  =1             INC     AX                      ; YES--INCREMENT UNIT COUNTER
00D9 3C04                =1             CMP     AL,4                    ; CHECK UNIT COUNTER (CLEARS CARRY)
00DB 75EE                =1             JNZ     INITLP                  ; LAST DRIVE INITIALIZED?
                         =1                                             ; NO--INITIALIZE NEXT DRIVE
00DD A10800              =1             MOV     AX,DEVCOD               ; YES--FLOPPIES INITIALIZED YET?
00E0 3C00                =1             CMP     AL,0
00E2 7508                =1             JNZ     INITDN                  ; YES--INITIALIZE FUNCTION FINISHED
00E4 C70608000100        =1             MOV     DEVCOD,01               ; NO---INITIALIZE FLOPPY DRIVES
00EA EBDF                =1             JMP     INITLP
00EC 1F                  =1     INITDN: POP     DS                      ; RESTORE REGISTERS
00ED 58                  =1             POP     AX
00EE CB                  =1             RET                             ; RETURN
                         =1     ;
                         =1     INIT215 ENDP
                                ;
                           +1   $INCLUDE(:F1:FORMAT.MMD)
                         =1 +1  $EJECT TITLE(FORMAT TRACK ROUTINE)
```

```
MCS-86 MACRO ASSEMBLER    FORMAT TRACK ROUTINE

LOC  OBJ                    LINE     SOURCE

                       =1           ; -------------------------------
                       =1           ; |                             |
                       =1           ; |     FORMAT TRACK ROUTINE    |
                       =1           ; |                             |
                       =1           ; -------------------------------
                       =1           ;
                       =1           ;       FMTTRK SETS UP THE IOPB FOR A FORMAT TRACK FUNCTION, AND
                       =1           ;       INVOKES THE iSBC 215 CONTROLLER TO PERFORM THE OPERATION.
                       =1           ;
                       =1           ; INPUT DATA:
                       =1           ;       BP + 10 =>  DEVICE CODE
                       =1           ;       BP + 9  =>  INTERLEAVE FACTOR
                       =1           ;       BP + 8  =>  USER DATA BYTE 3
                       =1           ;       BP + 7  =>  USER DATA BYTE 2
                       =1           ;       BP + 6  =>  USER DATA BYTE 1
                       =1           ;       BP + 5  =>  USER DATA BYTE 0
                       =1           ;       BP + 4  =>  TYPE OF FORMAT
                       =1           ;       BP + 3  =>  HEAD
                       =1           ;       BP + 1  =>  CYLINDER
                       =1           ;       BP      =>  UNIT
                       =1           ;
                       =1           ; OUTPUT DATA:
                       =1           ;       CARRY FLAG      = 0 IF TRACK FORMATTED SUCCESSFULLY
                       =1           ;                       = 1 IF NON-RECOVERABLE ERROR OCCURRED
                       =1           ;
                       =1           ; - INTERLEAVE FACTOR OF 1 IMPLIES SEQUENTIAL SECTOR NUMBERING.
                       =1           ; - USER DATA BYTES 0 - 3 ARE REPLICATED THROUGHOUT THE DATA FIELD.
                       =1           ; - INTERLEAVE TYPES:
                       =1           ;       00 = NORMAL TRACK (ONLY FORMAT FOR FLOPPY)
                       =1           ;       40 = ALTERNATE TRACK (POINTED TO BY EXACTLY ONE DEFECTIVE TRACK,
                       =1           ;               CANNOT SUBSEQUENTLY BE FORMATTED DEFECTIVE)
                       =1           ;       80 = DEFECTIVE TRACK (DATA FIELD POINTS TO ALTERNATE TRACK)
                       =1           ; - TO SET UP A POINTER TO AN ALTERNATE TRACK, SET:
                       =1           ;       USER DATA BYTE 0 = ALTERNATE CYLINDER LOW BYTE
                       =1           ;       USER DATA BYTE 1 = ALTERNATE CYLINDER HIGH BYTE
                       =1           ;       USER DATA BYTE 2 = ALTERNATE HEAD
                       =1           ;       USER DATA BYTE 3 = 00H
                       =1           ; ---------------------------------------------------------------------------
                       =1           ;
                       =1                   PUBLIC  FMT215
                       =1                   ASSUME  DS:IOPBSEG
                       =1           ;
00EF                   =1           FMT215  PROC    FAR
                       =1           ;
00EF 50                =1                   PUSH    AX                      ; SAVE REGISTERS
00F0 1E                =1                   PUSH    DS
00F1 B8----          R =1                   MOV     AX,IOPBSEG              ; GET POINTER TO IOPB
00F4 8ED8              =1                   MOV     DS,AX
00F6 8B460A            =1                   MOV     AX,[BP+10]              ; GET DEVICE CODE INTO IOPB
00F9 A30800            =1                   MOV     DEVCOD,AX
00FC 8A4600            =1                   MOV     AL,[BP]                 ; GET UNIT NUMBER INTO IOPB
00FF A20A00            =1                   MOV     UNIT,AL
0102 8B4601            =1                   MOV     AX,[BP+1]               ; GET CYLINDER NUMBER INTO IOPB
0105 A30E00            =1                   MOV     CYLNDR,AX
0108 8A4603            =1                   MOV     AL,[BP+3]               ; GET HEAD INTO IOPB
010B A21000            =1                   MOV     HEAD,AL
010E 892E1200          =1                   MOV     BUFOFF,BP               ; GET POINTER TO FORMAT ARGUMENT LIST
0112 8306120004        =1                   ADD     BUFOFF,4                ;    INTO DATA BUFFER POINTER
0117 8C161400          =1                   MOV     BUFSEG,SS
011B C6060B0002        =1                   MOV     FUNC,02H                ; SET FUNCTION = FORMAT
0120 C7060C000000      =1                   MOV     MODIFY,0000H            ; CLEAR MODIFIER (ALLOW ERROR RECOVERY
                       =1                                                   ;    AND INTERRUPT ON COMPLETION)
0126 E89900            =1                   CALL    GO215                   ; START iSBC 215 AND WAIT FOR DONE
                       =1                                                   ;    (RETURNS CARRY FLAG SET OR CLEAR)
0129 1F                =1           FMTDN:  POP     DS                      ; RESTORE REGISTERS
012A 58                =1                   POP     AX
012B CA0A00            =1                   RET     10                      ; RETURN (AND POP INPUT DATA OFF STACK)
                       =1           ;
                       =1           FMT215  ENDP
                                    ;
                                +1  $INCLUDE(:F1:RDWRT.MMD)
                       =1       +1  $EJECT TITLE(READ DATA ROUTINE)
```

```
MCS-86 MACRO ASSEMBLER    READ DATA ROUTINE

LOC  OBJ                  LINE     SOURCE

                   =1              ; ---------------------
                   =1              ; |                   |
                   =1              ; |     READ DATA     |
                   =1              ; |                   |
                   =1              ; ---------------------
                   =1              ;
                   =1              ;       RD215 SETS UP THE IOPB FOR A READ OPERATION, AND
                   =1              ;       INVOKES THE iSBC 215 TO PERFORM THE OPERATION.
                   =1              ;
                   =1              ; INPUT DATA:
                   =1              ;       BP + 13 =>  DEVICE CODE
                   =1              ;       BP + 11 =>  BYTE COUNT HIGH WORD
                   =1              ;       BP +  9 =>  BYTE COUNT LOW WORD
                   =1              ;       BP +  7 =>  DATA BUFFER SEGMENT
                   =1              ;       BP +  5 =>  DATA BUFFER OFFSET
                   =1              ;       BP +  4 =>  SECTOR
                   =1              ;       BP +  3 =>  HEAD
                   =1              ;       BP +  1 =>  CYLINDER
                   =1              ;       BP      =>  UNIT
                   =1              ;
                   =1              ; OUTPUT DATA:
                   =1              ;       CARRY FLAG      = 0 IF TRANSFER OCCURRED WITH NO OR RECOVERABLE ERROR
                   =1              ;                       = 1 IF UNRECOVERABLE ERROR OCCURRED
                   =1              ;       DATA BUFFER     FILLED WITH DATA FROM DISK IF NO UNRECOVERABLE ERROR
                   =1              ; ---------------------------------------------------------------------------------
                   =1              ;
                   =1                      PUBLIC  RD215
                   =1                      ASSUME  DS:IOPBSEG
                   =1              ;
012E               =1              RD215   PROC    FAR
                   =1              ;
012E 50            =1                      PUSH    AX                      ; SAVE REGISTERS
012F 1E            =1                      PUSH    DS
0130 B8----      R =1                      MOV     AX,IOPBSEG              ; GET POINTER TO IOPB
0133 8ED8          =1                      MOV     DS,AX
0135 8B460D        =1                      MOV     AX,[BP+13]              ; GET DEVICE CODE INTO IOPB
0138 A30800        =1                      MOV     DEVCOD,AX
013B 8A4600        =1                      MOV     AL,[BP]                 ; GET UNIT INTO IOPB
013E A20A00        =1                      MOV     UNIT,AL
0141 8B4601        =1                      MOV     AX,[BP+1]               ; GET CYLINDER INTO IOPB
0144 A30E00        =1                      MOV     CYLNDR,AX
0147 8B4603        =1                      MOV     AX,[BP+3]               ; GET HEAD AND SECTOR INTO IOPB
014A A31000        =1                      MOV WORD PTR HEAD,AX
014D 8B4605        =1                      MOV     AX,[BP+5]               ; GET DATA BUFFER POINTER INTO IOPB
0150 A31200        =1                      MOV     BUFOFF,AX
0153 8B4607        =1                      MOV     AX,[BP+7]
0156 A31400        =1                      MOV     BUFSEG,AX
0159 8B4609        =1                      MOV     AX,[BP+9]               ; GET BYTE COUNT INTO IOPB
015C A31600        =1                      MOV WORD PTR REQCNT,AX
015F 8B460B        =1                      MOV     AX,[BP+11]
0162 A31800        =1                      MOV WORD PTR REQCNT+2,AX
0165 C7060C000000  =1                      MOV     MODIFY,0000H            ; CLEAR MODIFIER (ENABLE INTERRUPT ON
                   =1                                                      ;     COMPLETION AND RETRIES)
016B C6060B0004    =1                      MOV     FUNC,04H                ; SET FUNCTION = READ DATA
0170 E84F00        =1                      CALL    GO215                   ; START FUNCTION AND WAIT FOR COMPLETION
                   =1                                                      ;    (RETURNS CARRY FLAG SET OR CLEAR)
0173 1F            =1                      POP     DS                      ; RESTORE REGISTERS
0174 58            =1                      POP     AX
0175 CA0D00        =1                      RET     13                      ; POP PARAMETERS OFF STACK AND RETURN
                   =1              ;
                   =1              RD215   ENDP
                   =1              ;
                   =1          +1  $EJECT TITLE(WRITE DATA ROUTINE)
```

```
MCS-86 MACRO ASSEMBLER    WRITE DATA ROUTINE

LOC  OBJ                    LINE     SOURCE

                        =1          ; ---------------------
                        =1          ; |                   |
                        =1          ; |     WRITE DATA    |
                        =1          ; |                   |
                        =1          ; ---------------------
                        =1          ;
                        =1          ;       WRT215 SETS UP THE IOPB FOR A WRITE OPERATION, AND
                        =1          ;       INVOKES THE iSBC 215 TO PERFORM THE OPERATION.
                        =1          ;
                        =1          ; INPUT DATA:
                        =1          ;       BP + 13 =>  DEVICE CODE
                        =1          ;       BP + 11 =>  BYTE COUNT HIGH WORD
                        =1          ;       BP +  9 =>  BYTE COUNT LOW WORD
                        =1          ;       BP +  7 =>  DATA BUFFER SEGMENT
                        =1          ;       BP +  5 =>  DATA BUFFER OFFSET
                        =1          ;       BP +  4 =>  SECTOR
                        =1          ;       BP +  3 =>  HEAD
                        =1          ;       BP +  1 =>  CYLINDER
                        =1          ;       BP      =>  UNIT
                        =1          ;
                        =1          ;       DATA BUFFER CONTAINS INFORMATION TO BE WRITTEN TO DISK
                        =1          ;
                        =1          ; OUTPUT DATA:
                        =1          ;       CARRY FLAG       = 0 IF TRANSFER OCCURRED WITH NO OR RECOVERABLE ERROR
                        =1          ;                        = 1 IF UNRECOVERABLE ERROR OCCURRED
                        =1          ; ---------------------------------------------------------------------------
                        =1          ;
                        =1                  PUBLIC  WRT215
                        =1                  ASSUME  DS:IOPBSEG
                        =1          ;
0178                    =1          WRT215  PROC    FAR
                        =1          ;
0178 50                 =1                  PUSH    AX                      ; SAVE REGISTERS
0179 1E                 =1                  PUSH    DS
017A B8----           R =1                  MOV     AX,IOPBSEG              ; GET POINTER TO IOPB
017D 8ED8               =1                  MOV     DS,AX
017F 8B460D             =1                  MOV     AX,[BP+13]              ; PUT DEVICE CODE IN IOPB
0182 A30800             =1                  MOV     DEVCOD,AX
0185 8A4600             =1                  MOV     AL,[BP]                 ; GET UNIT INTO IOPB
0188 A20A00             =1                  MOV     UNIT,AL
018B 8B4601             =1                  MOV     AX,[BP+1]               ; GET CYLINDER INTO IOPB
018E A30E00             =1                  MOV     CYLNDR,AX
0191 8B4603             =1                  MOV     AX,[BP+3]               ; GET HEAD AND SECTOR INTO IOPB
0194 A31000             =1                  MOV WORD PTR HEAD,AX
0197 8B4605             =1                  MOV     AX,[BP+5]               ; GET DATA BUFFER POINTER INTO IOPB
019A A31200             =1                  MOV     BUFOFF,AX
019D 8B4607             =1                  MOV     AX,[BP+7]
01A0 A31400             =1                  MOV     BUFSEG,AX
01A3 8B4609             =1                  MOV     AX,[BP+9]               ; GET BYTE COUNT INTO IOPB
01A6 A31600             =1                  MOV WORD PTR REQCNT,AX
01A9 8B460B             =1                  MOV     AX,[BP+11]
01AC A31800             =1                  MOV WORD PTR REQCNT+2,AX
01AF C7060C000000       =1                  MOV     MODIFY,0000H            ; CLEAR MODIFIER (ENABLE INTERRUPT ON
                        =1                                                  ;     COMPLETION AND RETRIES)
01B5 C6060B0006         =1                  MOV     FUNC,06H                ; SET FUNCTION = WRITE DATA
01BA E80500             =1                  CALL    GO215                   ; START iSBC 215 AND WAIT FOR DONE
                        =1                                                  ;    (RETURNS WITH CARRY SET OR CLEAR)
01BD 1F                 =1                  POP     DS                      ; RESTORE REGISTERS
01BE 58                 =1                  POP     AX
01BF CA0D00             =1                  RET     13                      ; POP PARAMETERS OFF STACK AND RETURN
                        =1          ;
                        =1          WRT215  ENDP
                                    ;
                                +1  $INCLUDE(:F1:CORE.MMD)
                        =1      +1  $EJECT TITLE(START FUNCTION AND WAIT FOR COMPLETION)
```

```
MCS-86 MACRO ASSEMBLER    START FUNCTION AND WAIT FOR COMPLETION

LOC  OBJ                LINE     SOURCE

                     =1          ; -------------------------------------------------
                     =1          ; |                                               |
                     =1          ; |      START FUNCTION AND WAIT FOR COMPLETION     |
                     =1          ; |                                               |
                     =1          ; -------------------------------------------------
                     =1          ;
                     =1          ;        THIS ROUTINE GIVES A CHANNEL ATTENTION (WAKE-UP) TO THE iSBC 215 AND
                     =1          ;        WAITS FOR THE FUNCTION SPECIFIED (BY THE CALLING PROCEDURE) TO FINISH.
                     =1          ;        IF AN ERROR OCCURRED, THE ERROR HANDLER IS INVOKED.
                     =1          ;
                     =1          ; INPUTS:
                     =1          ;        NONE
                     =1          ;
                     =1          ; OUTPUTS:
                     =1          ;        CARRY FLAG:      = 0 IF NO ERROR OR A RECOVERABLE ERROR OCCURRED
                     =1          ;                         = 1 IF UNRECOVERABLE ERROR OCCURRED.
                     =1          ; ---------------------------------------------------------------------------
                     =1          ;
01C2                 =1          GO215    PROC    NEAR
                     =1          ;
01C2 50              =1                   PUSH    AX                      ; SAVE REGISTERS
01C3 52              =1                   PUSH    DX
01C4 BA3506          =1                   MOV     DX,WUA                  ; GET ADDRESS OF WAKE-UP I/O PORT
01C7 B001            =1                   MOV     AL,01H                  ; GET WAKE-UP COMMAND BYTE
01C9 EE              =1                   OUT     DX,AL                   ; GIVE WAKE-UP TO iSBC 215
01CA E80800          =1                   CALL    WAIT215                 ; WAIT FOR FUNCTION COMPLETE
01CD 7303            =1                   JNC     DONE                    ; ERROR?
                     =1                                                   ; NO--RETURN
01CF E82900          =1                   CALL    ERROR                   ; YES--CALL ERROR HANDLER (RETURNS WITH
                     =1                                                   ;    CARRY FLAG SET OR CLEAR)
01D2 5A              =1          DONE:    POP     DX                      ; RESTORE REGISTERS
01D3 58              =1                   POP     AX
01D4 C3              =1                   RET                             ; RETURN
                     =1          ;
                     =1          GO215    ENDP
                     =1          ;
                     =1      +1  $EJECT TITLE(WAIT FOR FUNCTION COMPLETE ROUTINE)
```

```
MCS-86 MACRO ASSEMBLER    WAIT FOR FUNCTION COMPLETE ROUTINE

LOC  OBJ                  LINE     SOURCE

                       =1          ; ------------------------------------
                       =1          ; |                                  |
                       =1          ; |      WAIT FOR FUNCTION COMPLETE  |
                       =1          ; |                                  |
                       =1          ; ------------------------------------
                       =1          ;
                       =1          ;        NORMALLY, THIS WAIT ROUTINE WOULD TRAP TO THE SYSTEM DISPATCHER/
                       =1          ;        SCHEDULER TO ALLOW ANOTHER TASK TO EXECUTE WHILE THE iSBC 215 COMPLETED
                       =1          ;        ITS FUNCTION.  HOWEVER, FOR THIS EXAMPLE, THE ROUTINE SIMPLY WAITS FOR
                       =1          ;        THE INTERRUPT SERVICE ROUTINE TO LOAD THE OPERATION COMPLETE STATUS
                       =1          ;        FROM THE CIB OPERATION STATUS INTO THE DATA SEGMENT.  IF AN ERROR
                       =1          ;        OCCURRED, THE STATUS IS AVAILABLE THERE FOR SUBSEQUENT PROCESSING BY
                       =1          ;        AN ERROR HANDLER.
                       =1          ;
                       =1          ; INPUT DATA:
                       =1          ;        OPERATION COMPLETE STATUS FROM THE CIB, COPIED INTO THE DATA SEGMENT
                       =1          ;                BY THE INTERRUPT ROUTINE
                       =1          ;
                       =1          ; OUTPUT DATA:
                       =1          ;        OPERATION COMPLETE BYTE          CLEARED
                       =1          ;        CARRY FLAG                       = 0 IF NO ERROR
                       =1          ;                                         = 1 IF ERROR OCCURRED
                       =1          ;        COPY OF CIB OPERATION STATUS     IN "LSTSTS" IF ERROR OCCURRED
                       =1          ;
                       =1          ;        ( OPERATION COMPLETE BYTE AND "LSTSTS" ARE IN SEGMENT "DATASEG" )
                       =1          ;------------------------------------------------------------------------------
                       =1          ;
                       =1                  ASSUME   DS:DATASEG
                       =1          ;
01D5                   =1          WAIT215 PROC     NEAR
                       =1          ;
01D5 50                =1                  PUSH     AX                      ; SAVE REGISTERS
01D6 53                =1                  PUSH     BX
01D7 1E                =1                  PUSH     DS
01D8 BB----          R =1                  MOV      BX,DATASEG              ; GET POINTER TO DATA SEGMENT
01DB 8EDB              =1                  MOV      DS,BX
01DD BBFFFF            =1                  MOV      BX,-1                   ; INITIALIZE INDEX REGISTER
01E0 FB                =1                  STI                              ; MAKE SURE INTERRUPT CAN GET THROUGH
01E1 F4                =1                  HLT                              ; *****  WAIT FOR INTERRUPT  *****
01E2 43                =1          WAITLP: INC      BX                      ; GET INDEX FOR NEXT UNIT
01E3 81E30300          =1                  AND      BX,0003H                ; MASK UPPER BITS
01E7 F607FF            =1                  TEST     BYTE PTR [BX],0FFH      ; OPERATION COMPLETE STATUS = 00H?
                       =1                                                   ;    (SIGN FLAG = BIT 7 OF OP. STATUS,
                       =1                                                   ;    TEST INSTR. CLEARS CARRY FLAG)
01EA 74F6              =1                  JZ       WAITLP                  ; STATUS <> 00H (OPERATION COMPLETE)?
                       =1                                                   ; NO--CHECK NEXT UNIT
01EC 7906              =1                  JNS      WAITDN                  ; YES--ERROR OCCURRED DURING FUNCTION?
                       =1                                                   ; NO--RETURN WITH CARRY FLAG CLEAR
01EE 8A07              =1                  MOV      AL,[BX]                 ; YES--SAVE CIB OP. STATUS IN "LSTSTS"
01F0 A21800            =1                  MOV      LSTSTS,AL
01F3 F9                =1                  STC                              ; SET CARRY FLAG TO INDICATE ERROR
01F4 C60700            =1          WAITDN: MOV BYTE PTR [BX],00H            ; CLEAR OPERATION COMPLETE BYTE
01F7 1F                =1                  POP      DS                      ; RESTORE REGISTERS
01F8 5B                =1                  POP      BX
01F9 58                =1                  POP      AX
01FA C3                =1                  RET                              ; RETURN
                       =1          ;
                       =1          WAIT215 ENDP
                                   ;
                              +1   $INCLUDE(:F1:ERROR.MMD)
                       =1     +1   $EJECT TITLE(ERROR HANDLER)
```

```
MCS-86 MACRO ASSEMBLER    ERROR HANDLER

LOC  OBJ                  LINE     SOURCE

                     =1    888     ; ------------------------
                     =1    889     ; |                      |
                     =1    890     ; |      ERROR HANDLER   |
                     =1    891     ; |                      |
                     =1    892     ; ------------------------
                     =1    893     ;
                     =1    894     ;       THIS ROUTINE IS SYSTEM DEPENDENT.  IN THIS EXAMPLE, THE ERROR INFOR-
                     =1    895     ;       MATION FROM THE CONTROLLER IS READ INTO SOFTWARE REGISTERS IN DATASEG,
                     =1    896     ;       WHERE IT CAN BE EXAMINED.  MORE SOPHISTICATED SYSTEMS MIGHT LOG THE
                     =1    897     ;       ERRORS TO DETERMINE WHEN A TRACK IS GOING BAD, FOR EXAMPLE.
                     =1    898     ;
                     =1    899     ; - THE TRANSFER STATUS FUNCTION WILL NOT RETURN AN ERROR.
                     =1    900     ; - THE UNIT NUMBER IN THE IOPB IS NOT CHANGED, SO THAT THE OPERATION COMPLETE
                     =1    901     ;       STATUS FOR THE TRANSFER STATUS FUNCTION WILL BE POSTED AGAINST THE SAME
                     =1    902     ;       UNIT AS CAUSED THE ERROR.
                     =1    903     ;
                     =1    904     ; INPUT DATA:
                     =1    905     ;       CIB OPERATION STATUS     IN "LSTSTS" IN DATA SEGMENT
                     =1    906     ;
                     =1    907     ; OUTPUT DATA:
                     =1    908     ;       ERROR STATUS FROM CONTROLLER      IN DATA SEGMENT
                     =1    909     ;       CIB OPERATION STATUS              IN "LSTSTS" IN DATA SEGMENT
                     =1    910     ;       CARRY FLAG                        = 0 IF SOFT (RECOVERABLE) ERROR
                     =1    911     ;                                         = 1 IF HARD (UNRECOVERABLE) ERROR
                     =1    912     ; ---------------------------------------------------------------------------
                     =1    913     ;
                     =1    914             ASSUME  DS:IOPBSEG
                     =1    915     ;
01FB                 =1    916     ERROR   PROC    NEAR
                     =1    917     ;
01FB 50              =1    918             PUSH    AX                      ; SAVE REGISTERS
01FC 1E              =1    919             PUSH    DS
01FD B8----        R =1    920             MOV     AX,IOPBSEG              ; GET POINTER TO IOPB
0200 8ED8            =1    921             MOV     DS,AX
0202 A11200          =1    922             MOV     AX,BUFOFF               ; SAVE IOPB DATA BUFFER POINTER
0205 50              =1    923             PUSH    AX
0206 A11400          =1    924             MOV     AX,BUFSEG
0209 50              =1    925             PUSH    AX
020A C70612000C00    =1    926             MOV     BUFOFF,OFFSET ERRSTS    ; GET POINTER TO DATA SEGMENT ERROR
0210 C7061400----  R =1    927             MOV     BUFSEG,DATASEG          ;    STATUS REGISTERS
0216 C6060B0001      =1    928             MOV     FUNC,01H                ; SET FUNCTION = TRANSFER STATUS
021B C7060C000000    =1    929             MOV     MODIFY,0000H            ; CLEAR MODIFIER (ENABLE INTERRUPT ON
                     =1    930                                             ;    COMPLETION AND RETRIES)
0221 E89EFF          =1    931             CALL    G0215                   ; START FUNCTION AND WAIT FOR COMPLETE
0224 58              =1    932             POP     AX                      ; RESTORE IOPB DATA BUFFER POINTER
0225 A31400          =1    933             MOV     BUFSEG,AX
0228 58              =1    934             POP     AX
0229 A31200          =1    935             MOV     BUFOFF,AX
022C B8----        R =1    936             MOV     AX,DATASEG              ; GET POINTER TO DATA SEGMENT
022F 8ED8            =1    937             MOV     DS,AX
0231 F8              =1    938             CLC                             ; CLEAR CARRY FLAG
0232 A01800          =1    939             MOV     AL,DS:LSTSTS            ; GET OLD (ERROR) CIB OPERATION STATUS
0235 2440            =1    940             AND     AL,40H                  ; CHECK HARD ERROR BIT
0237 7401            =1    941             JZ      SFTERR                  ; HARD ERROR BIT SET?
                     =1    942                                             ; NO--LEAVE CARRY FLAG CLEAR
0239 F9              =1    943             STC                             ; YES--SET CARRY FLAG
023A 1F              =1    944     SFTERR: POP     DS                      ; RESTORE REGISTERS
023B 58              =1    945             POP     AX
023C C3              =1    946             RET
                     =1    947     ;
                     =1    948     ERROR   ENDP
                           949     ;
                           950 +1  $INCLUDE(:F1:INTRPT.MMD)
                     =1    951 +1  $EJECT TITLE(INTERRUPT SERVICE ROUTINE)
```

```
MCS-86 MACRO ASSEMBLER    INTERRUPT SERVICE ROUTINE

LOC  OBJ                 LINE     SOURCE

                    =1    952     ; -----------------------------------
                    =1    953     ; |                                 |
                    =1    954     ; |     INTERRUPT SERVICE ROUTINE   |
                    =1    955     ; |                                 |
                    =1    956     ; -----------------------------------
                    =1    957     ;
                    =1    958     ;       THIS ROUTINE SERVICES THE INTERRUPT GENERATED BY THE iSBC 215 UPON
                    =1    959     ;       OPERATION COMPLETE, SEEK COMPLETE, OR DISK PACK CHANGE.  IT COPIES THE
                    =1    960     ;       CIB OPERATION STATUS INTO ONE OF FOUR BYTES ASSOCIATED WITH EACH OF
                    =1    961     ;       THESE EVENTS.  IT IS ASSUMED THAT SYSTEM PROGRAMS MAKE USE OF THE
                    =1    962     ;       INFORMATION TO RESUME TASKS, HANDLE ERROR LOGGING/RECOVERY, AND KEEP
                    =1    963     ;       TRACK OF DIRECTORY INFORMATION.  FOR THIS PROGRAMMING EXAMPLE, ONLY
                    =1    964     ;       THE OPERATION COMPLETE BYTES ARE USED.
                    =1    965     ;
                    =1    966     ; - THE SYSTEM INTERRUPTS ARE CONFIGURED BY EXTERNAL PROGRAMS.
                    =1    967     ; ---------------------------------------------------------------------
                    =1    968     ;
                    =1    969             PUBLIC  INT215
                    =1    970     ;
023D                =1    971     INT215  PROC    FAR
                    =1    972     ;
023D FB             =1    973             STI                          ;;; ENABLE HIGHER PRIORITY INTERRUPTS
023E 50             =1    974             PUSH    AX                   ;;; SAVE REGISTERS
023F 53             =1    975             PUSH    BX
0240 52             =1    976             PUSH    DX
0241 1E             =1    977             PUSH    DS
                    =1    978             ASSUME  DS:CIBSEG
0242 B8----       R =1    979             MOV     AX,CIBSEG            ; GET POINTER TO CIB
0245 8ED8           =1    980             MOV     DS,AX
0247 A00100         =1    981             MOV     AL,OPSTS             ; GET CIB OPERATION STATUS
024A 8AD0           =1    982             MOV     DL,AL                ; SAVE IT
024C C606030000     =1    983             MOV     STSSEM,00H           ; CLEAR CIB STATUS SEMAPHORE
0251 8AD8           =1    984             MOV     BL,AL                ; MOVE IT TO INDEX REGISTER
0253 81E33000       =1    985             AND     BX,0030H             ; MASK ALL BITS EXCEPT UNIT NUMBER
0257 D1EB           =1    986             SHR     BX,1                 ; SHIFT UNIT NUMBER TO BITS 0 AND 1
0259 D1EB           =1    987             SHR     BX,1
025B D1EB           =1    988             SHR     BX,1
025D D1EB           =1    989             SHR     BX,1
025F 250600         =1    990             AND     AX,0006H             ; MASK ALL BITS EXCEPT SEEK COMPLETE
                    =1    991                                          ;    AND PACK CHANGE
0262 D1E0           =1    992             SHL     AX,1                 ; SHIFT LEFT TO GET OFFSET INTO PROPER
                    =1    993                                          ;    BYTE IN DATA SEGMENT
0264 03D8           =1    994             ADD     BX,AX                ; COMBINE WITH UNIT IN INDEX REGISTER
                    =1    995             ASSUME  DS:DATASEG
0266 B8----       R =1    996             MOV     AX,DATASEG           ; GET POINTER TO DATA SEGMENT
0269 8ED8           =1    997             MOV     DS,AX
026B 8817           =1    998             MOV     [BX],DL              ; MOVE OPERATION STATUS TO DATA SEGMENT
026D BA5063         =1    999             MOV     DX,WUA               ; GET POINTER TO I/O WAKE-UP ADDRESS
0270 B002           =1   1000             MOV     AL,00H               ; GET CLEAR INTERRUPT COMMAND BYTE
0272 EE             =1   1001             OUT     DX,AL                ; OUTPUT TO iSBC 215
                    =1   1002     ;
0273 1F             =1   1003             POP     DS                   ; RESTORE REGISTERS
0274 5A             =1   1004             POP     DX
0275 5B             =1   1005             POP     BX
0276 FA             =1   1006             CLI                          ; DISABLE INTERRUPTS FOR RESTORE
                    =1   1007                                          ; (RESTORATION OF INTERRUPT LOGIC STATE
                    =1   1008                                          ; IS SYSTEM DEPENDENT. THIS EXAMPLE USES
                    =1   1009                                          ; THE iSBC 86/12A CPU.)
0277 B020           =1   1010             MOV     AL,20H               ;;; GET END-OF-INTERRUPT COMMAND
0279 E6C0           =1   1011             OUT     0C0H,AL              ;;; OUTPUT EOI COMMAND TO 8259
027B 58             =1   1012             POP     AX                   ;;;
027C CF             =1   1013             IRET                         ;;; INTERRUPT RETURN ENABLES INTERRUPTS
                    =1   1014     ;
                    =1   1015     INT215  ENDP
                         1016     ;
----                     1017     SBC215DRIVER    ENDS                 ; END OF iSBC 215 DRIVER CODE
                         1018     ;
                         1019 +1  $TITLE(SYMBOL TABLE AND CROSS REFERENCE)
                         1020     ;
                         1021             END                          ; END OF PROGRAMMING EXAMPLE
```

```
MCS-86 MACRO ASSEMBLER     SYMBOL TABLE AND CROSS REFERENCE

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  312#
ACTHD . . . . V BYTE    0015H  DATASEG  313#
ACTSEC. . . . V BYTE    0016H  DATASEG  314#
BSYFLG1 . . . V BYTE    0001H  CCBSEG  79# 512
BSYFLG2 . . . V BYTE    0009H  CCBSEG  85#
BUFOFF. . . . V WORD    0012H  IOPBSEG  137# 565 567 646 647 706 770 922 926 935
BUFSEG. . . . V WORD    0014H  IOPBSEG  138# 563 648 708 772 924 927 933
CCB . . . . . L FAR     0000H  CCBSEG  64 77# 454 455
CCBPTR. . . . V DWORD   0002H  SCBSEG  64# 454 455 459
CCBSEG. . . . SEGMENT            SIZE=0010H PARA     75# 92 460 504 505
CCW1. . . . . V BYTE    0000H  CCBSEG  78# 461
CCW2. . . . . V BYTE    0008H  CCBSEG  84# 464
CH1PC . . . . L FAR     0004H  CIBSEG  80 110# 462 463 476
CH1PTR. . . . V DWORD   0002H  CCBSEG  80# 462 463
CH2PC . . . . L FAR     000EH  CCBSEG  86 89# 465 466 467
CH2PTR. . . . V DWORD   000AH  CCBSEG  86# 465 466
CIB . . . . . L FAR     0000H  CIBSEG  105#
CIBCMD. . . . V BYTE    0000H  CIBSEG  106# 474
CIBSEG. . . . SEGMENT            SIZE=0010H PARA     103# 116 471 472 978 979
CLRLP . . . . L NEAR    0078H  SBC215DRIVER  488# 491
CMDSEM. . . . V BYTE    0002H  CIBSEG  108# 475
CONFIG. . . . L FAR     0000H  USERSEG PUBLIC  377 380# 396
CYLNDR. . . . V WORD    000EH  IOPBSEG  134# 643 702 766
DATASEG . . . SEGMENT            SIZE=001AH PARA     268# 326 483 484 854 861 927 936 995 996
DESCYL. . . . V WORD    000FH  DATASEG  309#
DESHD . . . . V BYTE    0011H  DATASEG  310#
DESSEC. . . . V BYTE    0012H  DATASEG  311#
DEVCOD. . . . V WORD    0008H  IOPBSEG  130# 559 577 580 639 698 762
DONE. . . . . L NEAR    01D2H  SBC215DRIVER  816 820#
ENDDAT. . . . L FAR     001AH  DATASEG  324# 486
ENDSTK. . . . L FAR     0040H  STACK  367# 385
ERROR . . . . L NEAR    01FBH  SBC215DRIVER  818 916# 948
ERRSTS. . . . V WORD    000CH  DATASEG PUBLIC  278 307# 926
FMT215. . . . L FAR     00EFH  SBC215DRIVER PUBLIC  629 632# 658
FMTDN . . . . L NEAR    0129H  SBC215DRIVER  654#
FUNC. . . . . V BYTE    000BH  IOPBSEG  132# 560 649 715 779 928
GO215 . . . . L NEAR    01C2H  SBC215DRIVER  569 652 716 780 808# 824 931
HEAD. . . . . V BYTE    0010H  IOPBSEG  135# 645 704 768
INIT215 . . . L FAR     00A5H  SBC215DRIVER PUBLIC  550 553# 586
INITBLSEG . . SEGMENT            SIZE=003FH PARA     170# 258 563
INITDN. . . . L NEAR    00ECH  SBC215DRIVER  571 579 582#
INITLP. . . . L NEAR    00CBH  SBC215DRIVER  567# 575 581
INT215. . . . L FAR     023DH  SBC215DRIVER PUBLIC  388 389 969 971# 1015
INTRCS. . . . V WORD    0096H  SEG0000  355# 389
INTRIP. . . . V WORD    0094H  SEG0000  354# 388
INTRPT. . . . NUMBER    0005H    347# 352
IOPB. . . . . L FAR     0000H  IOPBSEG  112 127# 477 478
IOPBOFF . . . V WORD    0008H  CIBSEG  112# 477
IOPBSEG . . . SEGMENT            SIZE=001EH PARA       113 125# 142 551 557 630 636 689 695 753 759 914 920
IOPBSG. . . . V WORD    000AH  CIBSEG  113# 478
LSTSTS. . . . V BYTE    0018H  DATASEG  320# 876 939
MODIFY. . . . V WORD    000CH  IOPBSEG  133# 561 650 713 777 929
NMRTRY. . . . V BYTE    0017H  DATASEG  315#
OPCMP . . . . V BYTE    0000H  DATASEG PUBLIC  278 282#
OPCMP0. . . . V BYTE    0000H  DATASEG  283#
OPCMP1. . . . V BYTE    0001H  DATASEG  284#
OPCMP2. . . . V BYTE    0002H  DATASEG  285#
OPCMP3. . . . V BYTE    0003H  DATASEG  286#
OPSTS . . . . V BYTE    0001H  CIBSEG  107# 981
PKCHG . . . . V BYTE    0008H  DATASEG PUBLIC  278 298#
PKCHG0. . . . V BYTE    0008H  DATASEG  299#
PKCHG1. . . . V BYTE    0009H  DATASEG  300#
PKCHG2. . . . V BYTE    000AH  DATASEG  301#
PKCHG3. . . . V BYTE    000BH  DATASEG  302#
RD215 . . . . L FAR     012EH  SBC215DRIVER PUBLIC  688 691# 722
REQCNT. . . . V DWORD   0016H  IOPBSEG  139# 710 712 774 776
RES215. . . . L FAR     0000H  SBC215DRIVER PUBLIC  436 438# 526
RESDN . . . . L NEAR    009FH  SBC215DRIVER  514 519#
RESLP . . . . L NEAR    0095H  SBC215DRIVER  512# 516
SBC215DRIVER. SEGMENT            SIZE=027DH PARA     400# 402 1017
SCB . . . . . L FAR     0000H  SCBSEG  61#
SCBSEG. . . . SEGMENT            SIZE=0006H PARA ABS    59# 66 450 451
SECTOR. . . . V BYTE    0011H  IOPBSEG  136#
SEG0000 . . . SEGMENT            SIZE=0098H PARA ABS    349# 357 378
```

```
MCS-86 MACRO ASSEMBLER    SYMBOL TABLE AND CROSS REFERENCE


NAME           TYPE     VALUE  ATTRIBUTES, XREFS

SFERST. . . . V BYTE    000EH  DATASEG  308#
SFTERR. . . . L NEAR    023AH  SBC215DRIVER  941 944#
SKCMP . . . . V BYTE    0004H  DATASEG PUBLIC  278 290#
SKCMP0. . . . V BYTE    0004H  DATASEG  291#
SKCMP1. . . . V BYTE    0005H  DATASEG  292#
SKCMP2. . . . V BYTE    0006H  DATASEG  293#
SKCMP3. . . . V BYTE    0007H  DATASEG  294#
SOC . . . . . V BYTE    0000H  SCBSEG  62# 453
STACK . . . . SEGMENT          SIZE=0040H PARA
STSSEM. . . . V BYTE    0003H  CIBSEG  109# 983
UNIT. . . . . V BYTE    000AH  IOPBSEG  131# 568 641 700 764
USERSEG . . . SEGMENT          SIZE=0022H PARA    375# 398
WAIT215 . . . L NEAR    01D5H  SBC215DRIVER  815 856# 884
WAITDN. . . . L NEAR    01F4H  SBC215DRIVER  873 878#
WAITLP. . . . L NEAR    01E2H  SBC215DRIVER  866# 871
WRT215. . . . L FAR     0178H  SBC215DRIVER PUBLIC  752 755# 786
WUA . . . . . NUMBER    0635H    57# 59 497 812 999

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

***

***

iSBC® 215 GWD
144780-00

REQUEST FOR READER'S COMMENTS

Intel's Technical Publications Departments attempt to provide publications that meet the needs of all Int product users. This form lets you participate directly in the publication process. Your comments will he us correct and improve our publications. Please take a few minutes to respond.

Please restrict your comments to the usability, accuracy, readability, organization, and completeness this publication. If you have any comments on the product that this publication describes, please conta your Intel representative. If you wish to order publications, contact the Intel Literature Department (se page ii of this manual).

1. Please describe any errors you found in this publication (include page number).

2. Does the publication cover the information you expected or required? Please make suggestions fo improvement.

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

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

5. Please rate this publication on a scale of 1 to 5 (5 being the best rating). ____________

NAME ____________ DATE ____________

TITLE ____________

COMPANY NAME/DEPARTMENT ____________

ADDRESS ____________

CITY ____________ STATE ____________ (COUNTRY) ZIP CODE ____________

Please check here if you require a written reply. ☐

/E'D LIKE YOUR COMMENTS . . .

his document is one of a series describing Intel products. Your comments on the back of this form ill help us produce better manuals. Each reply will be carefully reviewed by the responsible erson. 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
5200 N.E. Elam Young Pkwy.
Hillsboro, Oregon 97123

OMO Technical Publications

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

Printed in U.S.A.

SINGLE BOARD COMPUTER

0144/2K/1284/AP/AD