iSBC® 214
PERIPHERAL CONTROLLER SUBSYSTEM
HARDWARE REFERENCE MANUAL

Order Number: 134910-001

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

Additional copies of this manual 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 to merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update nor to keep current the information contained in this document.

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

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

AEDIT	iDIS	iPDS	OpenNET
BITBUS	iLBX	iRMX	Plug-A-Bubble
BXP	i_m	iSBC	PROMPT
COMMputer	iMDDX	iSBX	Promware
CREDIT	iMMX	iSDM	QueX
Data Pipeline	Insite	iSXM	QUEST
Genius	Intel	KEPROM	Ripplemode
i	int_el	Library Manager	RMX/80
$\triangle$	int_elBOS	MCS	RUPI
i^2	Intelevision	Megachassis	Seamless
iATC	$int_eligent$ Identifier	MICROMAINFRAME	SLD
I^2ICE	$int_eligent$ Programming	MULTIBUS	SOLO
ICE	Intellec	MULTICHANNEL	SYSTEM 2000
iCS	Intellink	MULTIMODULE	UPI
iDBP	iOSP		

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

REV.	REVISION HISTORY	DATE
-001	Original Issue.	07/85

PREFACE

This manual describes the uses and functions of the iSBC 214 multi-peripheral controller board. The information presented in the manual will enable the user to configure and install this controller board for any MULTIBUS-compatible system. The manual assumes that the reader is familiar with the standards for Intel single-board computers and associated peripheral storage devices. It also assumes the reader has a general knowledge of programming which includes specific Intel device programming.

This manual references the information contained in the following technical publication:

- *Intel MULTIBUS Handbook*, Order Number: 2108833.

This manual uses the asterisk (*) after signal mnemonics to indicate that the signal is an active-low signal. A signal mnemonic without the trailing asterisk is an active-high signal.

TABLE OF CONTENTS

Contents

Tables

Figures

CHAPTER 1
GENERAL INFORMATION

1.1 Introduction

The iSBC 214 multi-peripheral controller provides MULTIBUS systems (8-bit and 16-bit data paths) with an interface for Winchester disk drives, flexible diskette drives, and streaming tape drives. A maximum of ten peripheral devices can interface with one board in the following configuration: two 5.25-inch Winchester disk drives, four 5.25-inch flexible diskette drives, and four ¼-inch streaming tape drives.

The iSBC 214 controller board emulates the iSBC 215G Winchester controller board communications protocol for interfacing the Winchester drives. The iSBX 217C and iSBX 218A MULTIMODULE® communications protocol is emulated for the tape drive and flexible disk drive interfaces. The Intel 80186 microprocessor serves as the controller's CPU. The Intel 80186 features direct memory access (DMA) and uses a dual-bus architecture to maximize data transfer between the host memory and attached peripheral devices. Each peripheral device interface has a dedicated on-board VLSI controller to perform the data I/O functions requested by the host system.

This chapter describes the major features and capabilities of the iSBC 214 controller. Figure 1-1 illustrates the board and identifies the MULTIBUS interface and peripheral device interface connections. The environmental and electrical specifications for the controller are contained in Table 1-1.

1.2 iSBC® 214 Controller Features

The iSBC 214 controller increases the peripheral storage device control capabilities of Intel's micro-systems. It uses VLSI technology for device interface control rather than MULTIMODULES. The following list briefly describes some of the major features of the iSBC 214 controller:

- Supports the ST506/412 interface for one or two 5.25-inch Winchester drives
- Supports the SA450/460 interface for up to four 5.25-inch flexible disk drives
- Supports the QIC-2 interface for up to four ¼-inch Streaming Tape Drives
- Supports 24-bit MULTIBUS addressing
- Emulates iSBC 215G controller I/O Parameter Block communications protocol (with or without the iSBX 217C and iSBX 218A MULTIMODULES)
- Supports overlapped seek operations
- Performs retry, Error Checking, and Correcting operations on Winchester drive data transfers
- Software selectable bytes per sector, sectors per track, and tracks per surface for Winchester and flexible disk drives
- Provides on-board power-up diagnostics

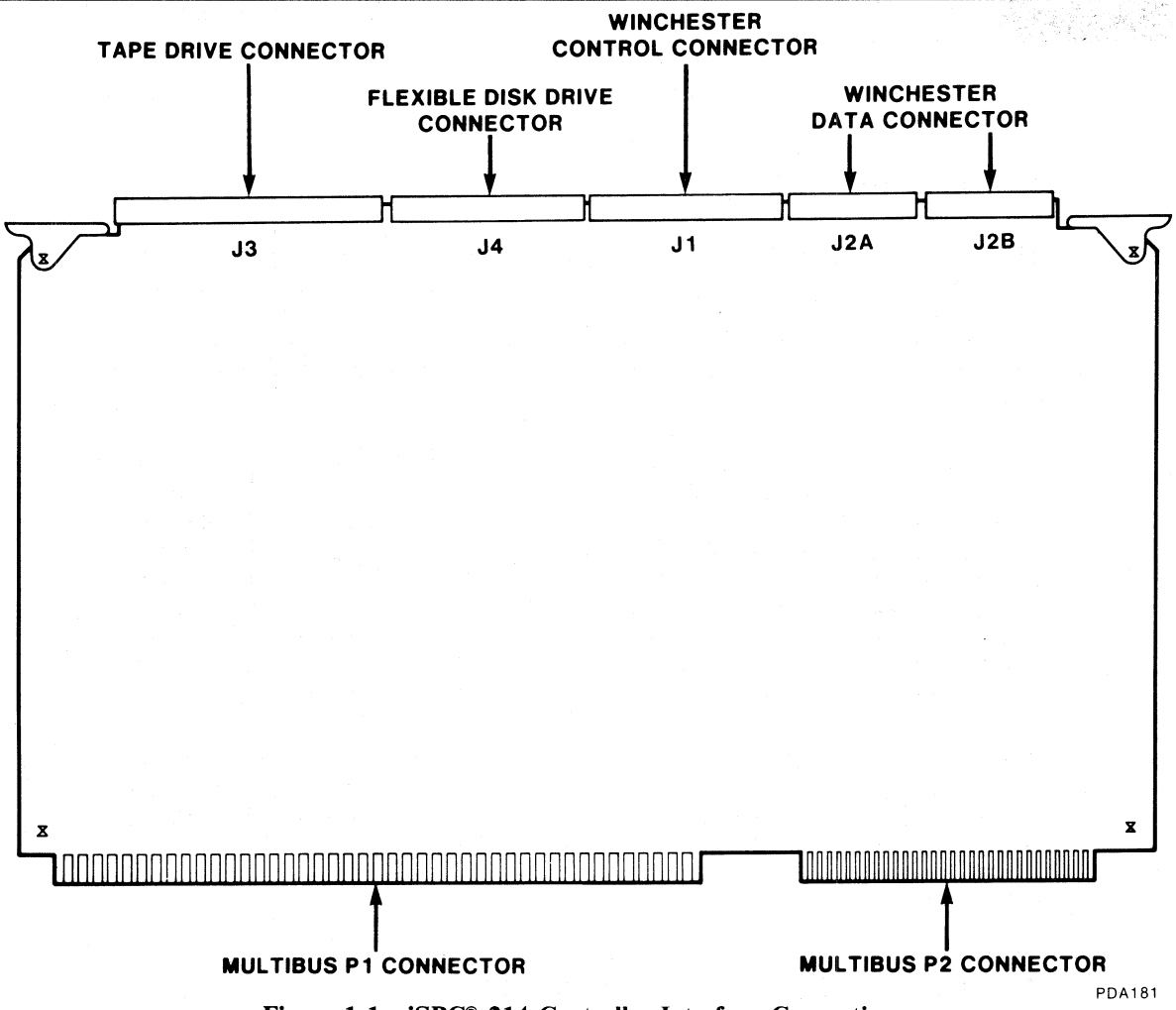


Figure 1-1. iSBC® 214 Controller Interface Connections

Track Buffering of the Winchester disk drive is also a standard feature on the iSBC 214 controller. The algorithms and features of track buffering include the following:

- Parts or all of the three most recently used tracks are stored in the track buffers at any one time.
- The requested sector plus the remainder of the track is read from the disk and written to the buffer.
- A rotational position sensor ensures the iSBC 214 controller will read the full track in one rotation time.
- Any read operation that references the data stored in the buffer will access the buffer and not the disk.
- Read operations that reference a track other than those stored in the track buffers will cause replacement of the least recently used track with data from the currently referenced track.
- The requested data is transferred to the host system as soon as it is read into the buffer and ECC checked.

- All write operations are written immediately to the disk and the buffer if applicable. If the write operation is not applicable to the buffer, the buffer data remains unaltered.

Physical connections for the Winchester interface consist of a 34-pin right-angle connector for the control cable and two 20-pin right-angle connectors for the data cables. The QIC-2 interface for tape drives consists of one 50-pin right-angle connector. The interface for the 5.25-inch flexible disk drives is a single 34-pin right-angle connector. Refer to Figure 1-1 for the interface locations on the board. Additional information on the pin assignments and signals for the peripheral device interfaces and the MULTIBUS interface is contained in Chapter 5.

1.3 Documentation

The latest version of board's schematics will be shipped with the iSBC 214 Controller Board.

1.4 Specifications

Table 1-1 lists the environmental and electrical specifications of the iSBC 214 controller.

Table 1-1. iSBC® 214 Controller Specifications

Physical Characteristics:	
Width:	6.75″ (17.2 cm)
Length:	12.00″ (30.5 cm)
Height:	0.5″ (1.3 cm)
Power Requirements:	
+5VDC (± 5%) @4.5 Amperes	
Environmental Requirements:	
Temperature	
operating:	0° C to +55° C (+50° F to +131° F)
non-operating:	−55° C to +85° C (−67° F to +185° F)
Humidity: up to 90% non-condensing	
Memory Capacity:	
ROM: 32K bytes	
RAM: 32K bytes	
Interface:	
MULTIBUS:	IEEE 796 System Bus Specification
Winchester Drive:	ST506/ST412
Flexible Disk Drive:	SA460/450 (5.25 inch drive)
Tape Drive:	QIC-2 Streaming Tape

Table 1-1. iSBC® 214 Controller Specifications (Cont'd)

Transfer Rates:

Controller to or from:	Transfer Rate
Host memory (16 bit data path)	1.6M bytes/sec (max)
Host memory (8 bit data path)	0.8M bytes/sec (max)
Winchester drive (ST506)	0.63M bytes/sec (fixed)
Flex. drive (D/D 5.25")	31.25K bytes/sec (fixed)
Flex. drive (S/D 5.25")	15.63K bytes/sec (fixed)
Tape drive (stream. 90 ips)	11.25K bytes/sec (avg)
Tape drive (stream. 30 ips)	30.00K bytes/sec (avg)

Address Selection:

User selectable between 20-bit and 24-bit address modes

MULTIBUS Compliance Levels:

Data bus:	D8-bit or D16-bit
Memory Address:	user defined M20 or M24
I/O Address:	user selectable, 8-bit or 16-bit
Interrupt Attributes:	V0 (non-bus vectored interrupts)

CHAPTER 2
CONTROLLER OPERATION

2.1 Introduction

This chapter provides a high-level explanation of the iSBC 214 controller board operation. The information presented is not intended to be a detailed theory of operation, rather a description of the functional operation of the board's major functions as shown in the logical block diagram, Figure 2-1.

2.2 iSBC® 214 Controller Board Functional Description

The iSBC 214 controller provides MULTIBUS compatible systems with an intelligent interface for 5.25-inch Winchester and flexible disk drives and ¼-inch streaming tape drives in a single board package. The controller performs data I/O operations at the request of the host system. Detailed command information, contained in four blocks of dedicated host memory, is fetched by the controller CPU. The controller CPU then initializes the appropriate on-board VLSI device controller to perform the actual operation. Chapter 3 contains a detailed description of the communications protocol used by the iSBC 214 controller. Figure 2-1 is a logical block diagram of the iSBC 214 controller. The following sections describe the major functions on the board as illustrated in Figure 2-1.

2.2.1 CPU

The iSBC 214 Board uses the Intel 16-bit, 80186 processor. The 80186 operates at 5 MHz and consists of a CPU, an interrupt controller, internal DMA controller, address select logic, wait-state generator, and clock divider circuit. The CPU is responsible for all transfer operations between the host system and the controller; data transfer between the host system and controller memory is handled by the 2-channel internal DMA controller. The CPU fetches I/O command structures from the host system memory and returns operational status to the host. The program store for the 80186 CPU consists of two Intel 27128 EPROMs which provide the controller with 32K bytes of Read-Only memory on the local bus.

2.2.2 Interfaces

The MULTIBUS interface is supported via two standard card edge connectors. The MULTIBUS interface subsystem consists of the MULTIBUS interrupt port, controller Wake-up port, and the address/data interface. The P1 connector is an 86-pin card edge connector which serves as the host system's communication channel to the controller. Power for the iSBC 214 Board is also provided through the P1 connector. The P2 connector is a 60-pin card edge connector which supports the optional 24-bit MULTIBUS addressing scheme and the power-fail ACLO signal.

The ST506/412 Winchester disk drive interface consists of a 34-pin right-angle connector for the control signal cable and two 20-pin right-angle connectors for the data I/O cables. The Winchester controller subsystem consists of three discrete LSI devices and four registered buffers. The Intel 82062 device functions as

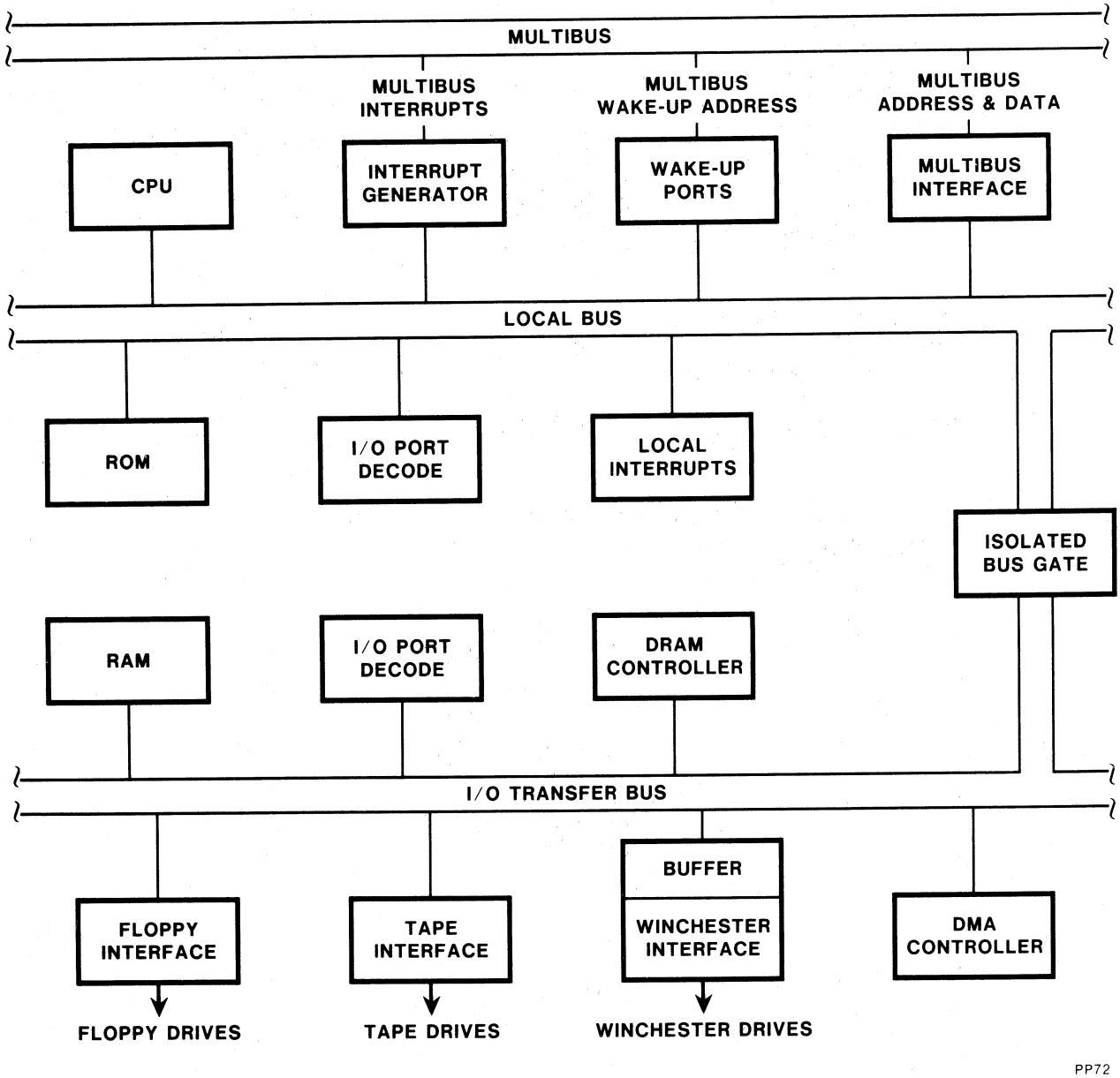


Figure 2-1. iSBC® 214 Controller Logic Block Diagram

the Winchester controller. This device performs read and write operations on Winchester drives. The WD1100-13 device generates the Error Correction Code (ECC).

The SA450/460 flexible disk drive interface uses a single 34-pin right-angle connector. The flexible disk drive controller is the WD2797 device. This device transfers data directly to the I/O transfer bus from the attached flexible disk drive(s).

The QIC-2 ¼-inch streaming tape drive interface uses a single 50-pin right-angle connector on the board. The tape drive controller subsystem is based on the Intel 8742 Universal Peripheral Interface (UPI). This device controls the tape drive interface and offloads the Intel 80186 CPU of much of the lowlevel hardware control required by the tape interface.

2.2.3 DMA Controller

The DMA controller on the iSBC 214 board is an Intel 8237A-5 DMA controller. This device is driven from the Intel 80186 5 MHz. clock and controls four independent DMA channels. The Intel 8237A-5 controls concurrent data transfer operations between the respective peripheral device controllers and the board's dynamic RAM. The four DMA channels have assigned fixed priorities; the highest priority is the Winchester controller, followed in descending priority by the flexible disk controller, tape controller, and DRAM refresh controller.

2.2.4 Memory

The iSBC 214 controller has 32K bytes of contiguous dynamic RAM, implemented with four 16K x 4, 18-pin devices. The RAM is accessed by the Intel 80186 CPU and the Intel 8237A-5 DMA controller over the I/O transfer bus. It supports I/O data transfer buffering and the Intel 80186 stack and interrupt vectors. The RAM is configured for 16-bit (word) access but also supports byte-swapping (occurring on DMA transfers between the controller memory and either the tape or flexible disk controllers) for 8-bit DMA I/O data transfers. The DRAM refresh controller consists of a discrete latch in conjunction with the fourth 8237A-5 DMA controller channel. In addition, a registered PAL serves as the state machine sequencer for all memory cycles while two non-registered PALs perform address multiplexing and shifting operations.

2.2.5 Interrupt Controller

The control firmware for the board is based on a custom real-time peripheral controller operating system and is interrupt driven. The Intel 80186 internal priority interrupt controller and two Intel 8259A Programmable Interrupt Controllers (PIC) are responsible for interrupt support. The Intel 80186 CPU can be interrupted by internally generated interrupts, Wake-Up port 8259A interrupts, I/O system 8259A interrupts, or non-maskable controller error interrupts. From a systems point-of-view, the iSBC 214 controller is configured for MULTIBUS interrupt 5.

2.2.5.1 Intel 80186 Internal Interrupts

The 80186 CPU acts as the master interrupt controller for the board. Internal interrupts may be generated from the Real-Time clock, the 80186 internal DMA controller, either of the two external (to the CPU) 8259A PICs, or the externally-driven NMI (Non-Maskable Interrupt) error signal.

2.2.5.2 Wake-up Port 8259A PIC

The Wake-Up Port 8259A Programmable Interrupt Controller (PIC) interrupts the on-board CPU whenever the host performs a write operation to the Wake-Up Port address.

2.2.5.3 I/O System 8259A Interrupts

The four remaining internal interrupt inputs to the I/O system 8259A PIC come from the three peripheral device I/O controllers and the Winchester ECC generation/checking circuitry. Interrupts through these input lines signify the completion of an I/O operation or a device error. The Winchester ECC circuit generates interrupts to the Intel 80186 CPU when there is an ECC error in the Winchester data flow.

2.2.5.4 Non-Maskable Interrupts

Three types of non-maskable interrupts (NMI) can occur through the input line to the on-board CPU. For each occurence of an NMI, the iSBC 214 controller requires a host system Wake-up or power-up restart to continue operations. The interrupts are as follows:

1. **Special Wake-up port channel 0 attention**—indicates that the host system is trying to issue a hardware reset to the controller and restart the board as if a power-up has just occurred.
2. **Bus timeout**—indicates that an 80186 requested access to its local bus, the I/O transfer bus, or the MULTIBUS and was not granted access within the time-out period of seven (7) milliseconds. This type of interrupt also causes the controller to post error status to the host and halt ALL I/O activity.
3. **A power failure**—indicated by the MULTIBUS ACLO signal which signifies that power to the system has dropped below the system's threshold. The Intel 80186 immediately halts ALL I/O operations and goes into a halt state.

2.2.6 Bus Architecture

The iSBC 214 controller uses a dual bus architecture for maximum I/O transfer efficiency. As shown in Figure 2-1, the board has a local bus and an I/O transfer bus. The local bus is a 16-bit data path accessible only by the 80186 CPU. The 80186 processor operates with 0 Wait States on the bus to process function codes and/or DMA data. The I/O transfer bus is also a 16-bit data path, but it is shared by the 80186 DMA processor and the 8237A-5 DMA controller to handle I/O data transfers to and from the peripheral storage devices.

CHAPTER 3
INSTALLATION

3.1 Introduction

This chapter provides the necessary information for preparing the iSBC 214 controller board and installing the board into a MULTIBUS-compatible system. Included in this chapter are instructions for unpacking and inspecting the controller. The remaining sections define the factory-installed jumper configuration of the iSBC 214 board. Table 3-7 at the end of this chapter summarizes the jumper configuration and options on the iSBC 214 board.

3.2 Unpacking and Inspecting the Controller

Immediately upon receipt of the controller package, inspect the shipping carton for evidence of damage or mishandling while in 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 all packing materials for a subsequent inspection by the agent.

For repair of the product damaged during shipment, contact the Intel Product Service Center to obtain a Return Authorization Number and receive further instructions. Chapter 6 lists the telephone numbers for the various service centers. Because a purchase order is required to complete the repair, submit a copy of the purchase order to the shipping carrier with your claim.

3.3 Installation Considerations

The iSBC 214 controller board can be installed in any Intel cardcage/backplane or user-designed backplane that complies with the MULTIBUS IEEE-796 interface specification. The board occupies one backplane slot.

The iSBC 214 board operates as a MULTIBUS multi-master interfacing Winchester drives, flexible disk drives and tape drives with the host system. For this reason, the slot where the board is installed must include bus priority arbitration.

3.3.1 Power Requirements

The iSBC 214 controller board requires a +5VDC (± 5%) power at a maximum current of 4.5 A. The power required by the tape drive, flexible disk drive, and Winchester drive is supplied through separate and direct cabling.

3.3.2 Cooling Requirements

The iSBC 214 controller board generates 21.329 kilogram calories of heat per minute (85.32 BTU per hour). (Figures based on a maximum of 25 Watts per hour.) Sufficient circulating, cooling air (about 200 linear feet per minute under normal operating conditions) must be provided to keep the environmental temperature within the required operating range (0° C to 55° C).

3.3.3 Physical Characteristics

The iSBC 214 controller board is a standard iSBC board with the following dimensions: 6.75 in. by 12.0 in. by 0.5 in. Table 1-1 provides a complete list of the controller's specifications.

3.4 iSBC® 214 Controller Jumper Configuration

The iSBC 214 board contains 138 stake pins which can be jumpered to configure the controller's functions in various system environments. Figure 6-1 illustrates the physical location and the default configuration of jumpered stake pins. Each stake pin on the board is identified by the letter E and a number and includes several hard-wired pads. The following sections explain the functions of each set or array of jumpered stake pins.

NOTE

A asterisk (*) following a signal mnemonic indicates the signal is active when it is in the low state.

3.4.1 Wake-Up Port Address Selection

The host CPU communicates with the iSBC 214 board through an I/O port, called the Wake-up port, and four communication blocks in the host system memory. The wake-up port is a byte wide port in the systems I/O address space whose address is defined by a group of jumpers on the iSBC 214 board.

Sixteen pairs of jumpers on the iSBC 214 board define all sixteen bits of the wake-up port address. These jumpers are located in a logical manner from left to right, corresponding to the MSB (Most Significant Bit) through LSB (Least Significant Bit) of the wake-up port address. Installed jumper plugs define the corresponding address bits to "1". Refer to table 3-1 to find the correlation between the jumper pairs and the bits of the wake-up port address.

For a host CPU which has 8 bit I/O address space (0 through 255) only eight of these sixteen jumpers (the ones corresponding to the least significant eight bits of the wake-up port address) should be configured for the required wake-up port address. The other eight jumpers should also be configured for the Wake-up Block Address calculation. See section 3.4.2 for the details of the wake-up block address.

Jumpers E119–E120 determine the host CPU's I/O address space range. A jumper plug must be installed for the host CPU's with 16-bit I/O address space, such as Intel 8086 (**default configuration**); not installed for the host CPU's with 8-bit I/O address space, such as Intel 8085.

Table 3-1. Wake-up Port Address Jumpers

Jumper		Wake-up Port Address Bit
From	To	
E43	E44	F
E45	E46	E
E47	E48	D
E49	E50	C
E51	E52	B
E53	E54	A
E55	E56	9
E57	E58	8 (*)
E59	E60	7
E61	E62	6
E63	E64	5
E65	E66	4
E67	E68	3
E69	E70	2
E71	E72	1
E73	E74	0

(*)—Default configuration; jumper installed for the default wake-up port address 0100H.

3.4.2 Wake-up Block Address Selection

Four I/O communication blocks in the host system memory are used as part of the communication protocol between the host CPU and the iSBC 214 board. Refer to Chapter 4—Programming Information—for a detailed discussion of the communication protocol and these I/O communication blocks. The first of these blocks is the Wake-up block, which requires six bytes in the host system memory. The wake-up block address and the wake-up port address together identify the location of the wake-up block. The iSBC 214 board reads the wake-up port address jumpers to calculate the wake-up block address. This gives a 20-bit address to locate the wake-up block anywhere in a 1 Megabyte memory address space. In Multibus systems with up to 16 Megabyte address space (24-bit addressing), the wake-up block can be located in either the first 1 Megabyte page or the last 1 Megabyte page. Installing jumper E75-E76 (**default configuration**) places the wake-up block in the first Megabyte page (0XXXXXH), and removing the jumper E75-E76 places the wake-up block in the last Megabyte page (FXXXXXH). Jumper E75-E76 must be installed for the systems with up to 1 Megabyte memory address space.

Note that all sixteen bits of the wake-up port address is used for the wake-up block address calculation, therefore the most significant eight bits of the wake-up port jumpers should be configured for the required wake-up block address even if the board is configured for 8-bit I/O address space. Table 3-2 is an example to clarify these options.

Table 3-2. Wake-up Block Address Selection Example

Wake-up Port Address Jumpers = 1234H		Wake-up Port Address (I/O address space)	Wake-up Block Address (Memory address space)
8-bit I/O Address Space	PAGE-0 E75-E76 Inst.	34H	012340H
Jumper E119-E120 Removed	PAGE-F E75-E76 Open	34H	F12340H

Table 3-2. Wake-up Block Address Selection Example (Cont'd)

Wake-up Port Address Jumpers = 1234H		Wake-up Port Address (I/O address space)	Wake-up Block Address (Memory address space)
16-bit I/O Address Space	PAGE-0 E75-E76 Inst.	1234H	012340H
Jumper E119-E120 Removed	PAGE-F E75-E76 Open	1234H	F12340H

Assume wake-up port address jumpers are configured for 1234H.

3.4.3 System Data Bus Width Selection

System data bus width selection jumper E77-E78 sets the board for the type of system data bus with which the iSBC 214 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.

3.4.4 Interrupt Priority Number

The iSBC 214 board internal interrupt request signal can be assigned to any of eight Multibus interrupt priority numbers (INT0* through INT7*). The number is selected by wire wrapping two jumper stake pins together as indicated in Table 3-3. The default configuration is INT5* (wirewrap jumper from E126 to E129).

Table 3-3. Interrupt Priority Number Selection

Interrupt Number	Install Wire Wrap Jumper	
	From Stake Pin	To Stake Pin
0	E126	E134
1	E126	E133
2	E126	E132
3	E126	E131
4	E126	E130
5	E126	E129
6	E126	E128
7	E126	E127

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

The CBRQ* and ANYRQST signals provide the required mode select inputs to the 8289 arbiter. The arbitration options are shown in Table 3-4.

CBRQ* is a bidirectional 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, while no other master is requesting control of the bus. The signal is either supplied from the bus via connector P1 or connected to

ground, depending on the configuration of the jumper set E100, E101, and E102. This signal operates the same in a parallel or a serial priority resolution scheme.

ANYRQST is a bus arbiter input signal that controls whether the iSBC 214 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 +5V through a resistor), or low (connected to ground), depending on the configuration of the jumper set E97, E98, and E99. 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 3-4. Bus Arbitration Options

Signal	Jumper	Connect To	Description
CBRQ* and ANYRQST	E101-E102 E97-E98	Bus Ground	Arbitrate to gain access of Multibus interface. If continued access is required, iSBC 214 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	E101-E102 and E98-E99	Bus +5V	Arbitrate to gain access to and multibus interface. If continued access is required, iSBC 214 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	E100-E101 and E98-E99	Ground +5V	Arbitrate for every bus access.

3.4.5.1 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 E95-E96, or disabled for parallel resolution by removing the jumper. The **default configuration** is for parallel resolution with the jumper removed.

3.4.6 MULTIBUS® Interface ACLO Signal

A three-way jumper set, E141, E142, and E143, provides the iSBC 214 controller board with a connection to the MULTIBUS interface ACLO signal. This signal, when active, indicates a potential power failure within the host system. This signal is an optional pin on the MULTIBUS system P2 connection and may not be supported by all systems that use the iSBC 214 board. The **default configuration** jumpers E142 to E143 which disables the ACLO input. To enable the ACLO input, jumper pin E141 to pin E142. When ACLO is enabled and active, it generates a Non-Maskable Interrupt (NMI) to the Intel 80186 CPU. See Section 2.2.5.4 for additional information.

3.4.7 Flexible Disk Drive Configuration

The following paragraphs describe the **default configuration** and function of the stake pins that allow the board to interface to the SA450/460 5.25-inch flexible

disk drive standard. Table 3-5 lists the factory default jumper configuration for the iSBC 214 controller.

Table 3-5. Flexible Disk Drive Jumper Configuration

Jumper	Function
E7-E8 — open	Reserved—must be open
E9-E10 — open	Reserved—must be open
E31-E33 — jumper	Drive Ready Signal
E34-E35 — open	Disk Write Precompensation
E36-E37 — jumper	Reserved—must be installed
E38-E39 — jumper	Reserved—must be installed
E115-E116 — jumper	Reserved—must be installed
E144-E145 — jumper	Pump Filter Circuit

3.4.7.1 Disk Drive Ready Signal

A three-way jumper set, stake pins E31, E32, and E33 allows the user two options for the Drive Ready signal. The board can be configured to force the signal active (high) constantly (for drives that do not control the Drive Ready signal), or to support drives that generate their own Drive Ready Signal. The **default configuration** jumpers E31 to E33 which causes the board to drive the Drive Ready signal active (high). Jumper pin E31 to pin E32 for those drives that generate their own Drive Ready signal.

3.4.7.2 Disk Drive Write-Precompensation

Stake pins E34 and E35 allow the user two options for the flexible disk drive Write-Precompensation; write precompensation on all drive tracks, or just on tracks greater than 43. The **default configuration** opens pin E34 to pin E35 which causes the WD2797 Flexible Disk Controller to automatically invoke internal Write-Precompensation on tracks greater than 43. This causes EARLY, ON-TIME, and LATE timing delays in the Write Data output from the WD2797 circuit.

3.4.7.3 Media Change Detection and Motor Control Options

Two stake pin pairs, E113/E114 and E117/E118, work in conjunction to allow the user to select one of three valid flexible disk drive media change detection. The **default configuration** jumpers pin E113 to pin E114 and pin E117 to pin E118 which selects Option 1. The three valid options are defined as follows:

Option 1 — This option is used for drives without the Drive Ready signal. A jumper plug must be installed between jumper posts E31 and E33 to supply a ready signal to the flexible disk controller chip. This option can be used with the drives with or without the head load option, but proper Head Load signal is provided to the interface. After four seconds of inactivity, the motor is turned off to protecting the media. Note that flexible disk media changes are not detected with this option.

Option 2 — This option is used for the drives with the limited implementation of the Ready signal, where the Motor-on signal is not a qualifier (i.e. ready = door closed). A jumper plug must be installed between the jumper posts E31 and E32 to pass the Ready signal of the selected flexible disk drive to the flexible disk controller chip. This option can be used with drives with or without the head load/unload capability, but proper Head Load signal is provided to the interface. Media is protected by turning the motor off after four seconds of inactivity. Media change is detected by the firmware via periodic sampling of the Ready signal. When the iSBC 214 board detects a not ready to ready transition of a flexible disk drive, it recalibrates the drive and sends an interrupt to the host system CPU to report the media change.

Option 3 — This option is used for the drives with the Ready signal and head load/unload capability. A jumper plug must be installed between the jumper posts E31 and E32 to pass the Ready signal of the selected flexible disk drive to the flexible disk controller chip. When this option is used, the Motor-on signal is asserted to the interface all the time and media is protected by unloading the head after three seconds of inactivity. Media change is detected by the firmware via periodic sampling of the Ready signal. When the iSBC 214 board detects a not ready to ready transistion of a flexible disk drive, it recalibrates the drive and sends an interrupt to the host system CPU to report the media change.

Table 3-6. Flexible Disk Media Change Detection Options

Option	Jumpers E113 E117 E114 E118	Ready Jumpers E31 E32 E33	Comments
1 default	ON ON	E31-E33	For drives without drive ready signal. Media is protected by motor off after 4 seconds of inactivity. No media change detection.
2	ON OFF	E31-E32	For the drives with ready = door closed. Media is protected by motor off after 4 seconds of inactivity. Media changes are detected.
3	OFF OFF	E31-E32	For the drives with Ready signal and head load/unload capability. Media is protected by head unload. Media changes are detected.

3.4.8 Winchester Drive Configuration

The following paragraphs describe the stake pins and jumper functions on the controller that pertain to the Winchester disk drives.

3.4.8.1 Read/Write Head Option

A three-way jumper set, stake pins E79, E80, and E81, permits the Winchester Head Select Line 3 (interface connector J1, pin 2) to be selected as a fourth head select line or as a reduced write current control line. The **default configuration** jumpers pin E79 to E81 to support a fourth head select line. This allows interfacing to Winchester drives that have up to sixteen heads. To change the configuration to support the reduced write current control line, the stake pins E79 and

E80 must be jumpered together. This configuration allows interfacing to Winchester drives with up to eight heads.

NOTE

Jumpered stake pins E79-E80 enables reduced Write Current, E79-E81 enables Head Select (2^3). For drives with eight or fewer heads that also do not require reduced write current, remove all jumpers.

3.4.8.2 Interface Seek Operations

Stake pins E121 and E122 allow the controller to perform buffered or non-buffered seeks to the Winchester disk drive interface. The **default configuration** jumpers pin E121 to pin E122, enabling buffered seeks. Leave E121 and E122 open to enable non-buffered seeks by the iSBC controller.

3.4.9 QIC-2 Interface Parity Check

A three-way jumper set, E1, E2, and E3, determines if parity checking on the 8-bit parallel QIC-2 tape interface is enabled or disabled. The **default configuration** jumpers pin E2 to pin E3 which disables the parity check. If parity checking is desired, jumper pin E1 to pin E2.

3.4.10 iSBC® 214 Controller Hard Reset

A three-way jumper set, stake pins E4, E5, and E6, determines whether the iSBC 214 controller can be hard reset by sending a Reset command to wake-up Port 0. The **default configuration** jumper E4 to E5 disables this feature; the jumper E5 to E6 enables it. Therefore (in the default configuration) the controller can only be hard reset when the MULTIBUS interface signal, INIT, is driven active (low).

3.4.10.1 Time-Out Circuitry

A three-way set, stake pins E11, E12, and E13, determines whether or not the on-board Time-Out circuitry for bus access is enabled. When enabled, the Time-Out signal is active for about 7 milliseconds after the Intel 80186 CPU has initiated a bus cycle without receiving an Acknowledge signal. This causes a non-maskable interrupt (NMI) to the 80186 CPU and subsequently results in a HALT condition for the on-board processor. See Section 2.2.5.4 for additional information. The **default configuration** jumpers pin E12 to pin E13 which enables the Time-Out signal. To disable the Time-Out circuitry, jumper pin E11 to pin E12.

3.4.10.2 iSBC® 214 Non-Maskable Interrupt

Stake pins E82 and E83 determine whether several possible error conditions can cause a non-maskable interrupt (NMI) to the controller's on-board CPU. This NMI subsequently results in a HALT condition for the CPU which requires a wake-up from the host system to recover. The **default configuration** jumpers pin E82 to pin E83, which enables all sources of NMI.

Table 3-7. Summary of Available Jumpers

Stake Pin	Jumper or Wire Wrap	Function	Reference Section
E1-E2	Installed	Enable Tape parity check	3.4.9
*E2-E3	Installed	Disable Tape parity check	3.4.9
*E4-E5	Installed	Disable Hard Reset	3.4.10
E5-E6	Installed	Enable Hard Reset	3.4.10
*E7-E8	Not Installed	Reserved-must be open	3.4.7
*E9-E10	Not Installed	Reserved-must be open	3.4.7
E11-E12	Installed	Disable Time-Out Signal	3.4.10.2
*E12-E13	Installed	Enable Time-Out Signal	3.4.10.2
*E15-E16	Installed	Reserved	
*E17-E18	Installed	Reserved	
*E20-E21	Installed	Reserved	
*E22-E23	Installed	Reserved	
E31-E32	Installed	Flex. Drive supports own ready signal	3.4.7.1
*E31-E33	Installed	Flex. Disk Drive Ready Signal	3.4.7.1
*E34-E35	Not Installed	Enable Write Precomp	3.4.7
*E36-E37	Installed	eserved-must be jumpered	3.4.7
*E37-E39	Installed	Reserved-must be jumpered	3.4.7
E43-E44	Installed	Wake-up address bit F	3.4.1
E45-E46	Installed	Wake-up address bit E	3.4.1
E47-E48	Installed	Wake-up address bit D	3.4.1
E49-E50	Installed	Wake-up address bit C	3.4.1
E51-E52	Installed	Wake-up address bit B	3.4.1
E53-E54	Installed	Wake-up address bit A	3.4.1
E55-E56	Installed	Wake-up address bit 9	3.4.1
*E57-E58	Installed	Wake-up address bit 8	3.4.1
E59-E60	Installed	Wake-up address bit 7	3.4.1
E61-E62	Installed	Wake-up address bit 6	3.4.1
E63-E64	Installed	Wake-up address bit 5	3.4.1
E65-E66	Installed	Wake-up address bit 4	3.4.1
E67-E68	Installed	Wake-up address bit 3	3.4.1
E69-E70	Installed	Wake-up address bit 2	3.4.1
E71-E72	Installed	Wake-up address bit 1	3.4.1
E73-E74	Installed	Wake-up address bit 0	3.4.1
*E75-E76	Installed	Select first Megabyte Page	3.4.2
E75-E76	Not Installed	Select last Megabyte Page	3.4.2
*E77-E78	Installed	16-bit data bus	3.4.3
E77-E78	Not Installed	8-bit data bus	3.4.3
E79-E80	Installed	1-8 Wini heads	3.4.8.1
*E79-E81	Installed	1-16 Wini heads	3.4.8.1
*E82-E83	Installed	Enable NMI	3.4.10.3
E82-E83	Not Installed	Disable NMI	3.4.10.3
*E84-E85	Installed	Reserved	
*E86-E87	Installed	Support 27128 EPROMs	
*E89-E90	Installed	Reserved	
*E91-E92	Installed	Reserved	
*E93-E94	Installed	Reserved	
E95-E96	Installed	Serial priority resolution	3.4.5.1
*E95-E96	Not Installed	Parallel priority resolution	3.4.5.1
*E97-E98	Installed	Relinquish bus on ANYRQST	3.4.5
E98-E99	Installed	Relinquish bus on CBRQ*	3.4.5
E100-E101	Installed	Common bus request I/O	3.4.5
*E101-E102	Installed	Relinquish bus after MULTIBUS access	3.4.5
*E103-E104	Installed	Reserved	
*E105-E106	Installed	Reserved	
E107-E108	Not Installed	Reserved—must be open	
E109-E110	Not Installed	Reserved—must be open	
*E111-E112	Installed	Reserved	
*E113-114	Installed	Flex. Drive Media Change Select	3.4.7.3
*E115-E116	Installed	Reserved—must be installed	3.4.7
*E117-E118	Installed	Flex drive media change select	3.4.7.3
*E119-E120	Installed	Select 16-bit I/O address	3.4.1
E119-E120	Not Installed	Select 8-bit I/O address	3.4.1
*E121-E122	Installed	Enable Win. buffered seeks	3.4.8.2
E121-E122	Not Installed	Enable Win. non-buffered seeks	3.4.8.2

Table 3-7. Summary of Available Jumpers (Cont'd)

Stake Pin	Jumper or Wire Wrap	Function	Reference Section
E126-E134	Installed	MULTIBUS Interrupt 0	3.4.4
E126-E133	Installed	MULTIBUS Interrupt 1	3.4.4
E126-E132	Installed	MULTIBUS Interrupt 2	3.4.4
E126-E131	Installed	MULTIBUS Interrupt 3	3.4.4
E126-E120	Installed	MULTIBUS Interrupt 4	3.4.4
*E126-E129	Installed	MULTIBUS Interrupt 5	3.4.4
E126-E128	Installed	MULTIBUS Interrupt 6	3.4.4
E126-E127	Installed	MULTIBUS Interrupt 7	3.4.4
*E135-E138	Installed	Reserved	
*E136-E139	Not Installed	Reserved—must be open	
*E137-E140	Installed	Reserved	
*E142-E143	Installed	Disable ACLO input	3.4.6
E141-E142	Installed	Enable ACLO input	3.4.6
*E144-E145	Installed	Pump Filter Circuit	3.4.7

*Indicates default configuration

CHAPTER 4
PROGRAMMING INFORMATION

4.1 Introduction

This chapter describes the programming conventions necessary to initiate and monitor data transfers between the host system memory and the attached peripheral storage devices (Winchester disk, flexible disk, and streaming tape drives). The iSBC 214 controller firmware emulates the I/O communications protocol of the iSBC 215G disk controller board, the protocols associated with the iSBX 218A Flexible disk controller, and the iSBX 217C tape controller MULTIMODULES mounted on the iSBC 215G board. Hard disk operation of the iSBC 214 board is limited to the ST506/412 compatible Winchester disk drives.

4.2 Host/Board Communications

The iSBC 214 board has a DMA controller 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. 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 214 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 block 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 block must be contiguous. Each of the blocks also has a defined function related to the overall operation of the iSBC 214 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.

4.2.1 Wake-Up I/O Port

The wake-up port is the I/O address to which the iSBC 214 board responds. The I/O address is user selectable through jumpers on the board and may be either 8 or 16 bits, depending on the host CPU and the applications. 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 address of the first byte in the wake-up block. Thus wake-up I/O port address 100H also specifies wake-up block address 1000H. (Refer to section 3.4.2 for more information.)

To invoke iSBC 214 board activity, the host CPU transmits a wakeup command byte to the board through the wake-up I/O port. Three wake-up commands are allowed, as shown in Table 4-1. 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.

NOTE

Important timing restrictions exist on the commands listed in Table 4-1 during Cold Start. Refer to section 4.2.7 in for details.

Table 4-1. I/O Channel Commands

Command	Function
00H	**CLEAR INTERRUPT**—The I/O port to host CPU interrupt is reset, the channel reset is cleared.
01H	**START OPERATION**—Instructs the board to start the operation defined by the communication block element.
02H	**RESET BOARD**—Instructions to perform a software reset of the controller board. A Clear Interrupt command must be initiated following this command. Each time the controller is reset, the communication link between the board and the host must be reestablished.
03-FFH	Reserved

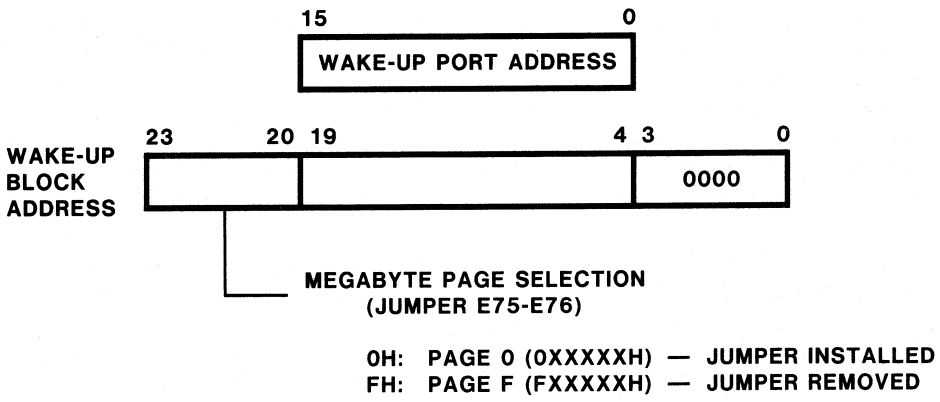


Figure 4-1. Wake-Up Block Address Structure

4.2.2 I/O Communications Block

The host CPU and the iSBC 214 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 called the wake-up block, channel control block, controller

invocation block, and I/O parameter block. The iSBC 214 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 214 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 214 board activity. The sequence (see Figure 4-2) in which the board accesses these blocks varies with the type of operation being performed. For a general data read function, 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 214 board is to perform (for example, read-data).
2. The host then transmits a wake-up command (01H) to the wakeup I/O port, signalling the board to read the I/O communication 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. All subsequent wake-up commands cause the iSBC 214 board to read the channel control block.)
4. At the I/O parameter block, the iSBC 214 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.

These I/O communication 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, addressed through the Multibus interface. Once the host has initiated board activity, the 80186 Processor handles all communications between the host CPU, host memory, and disk drive controllers. 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.

Note that in the following command block descriptions, all bytes shown as reserved in the illustrations should be set to 0 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.

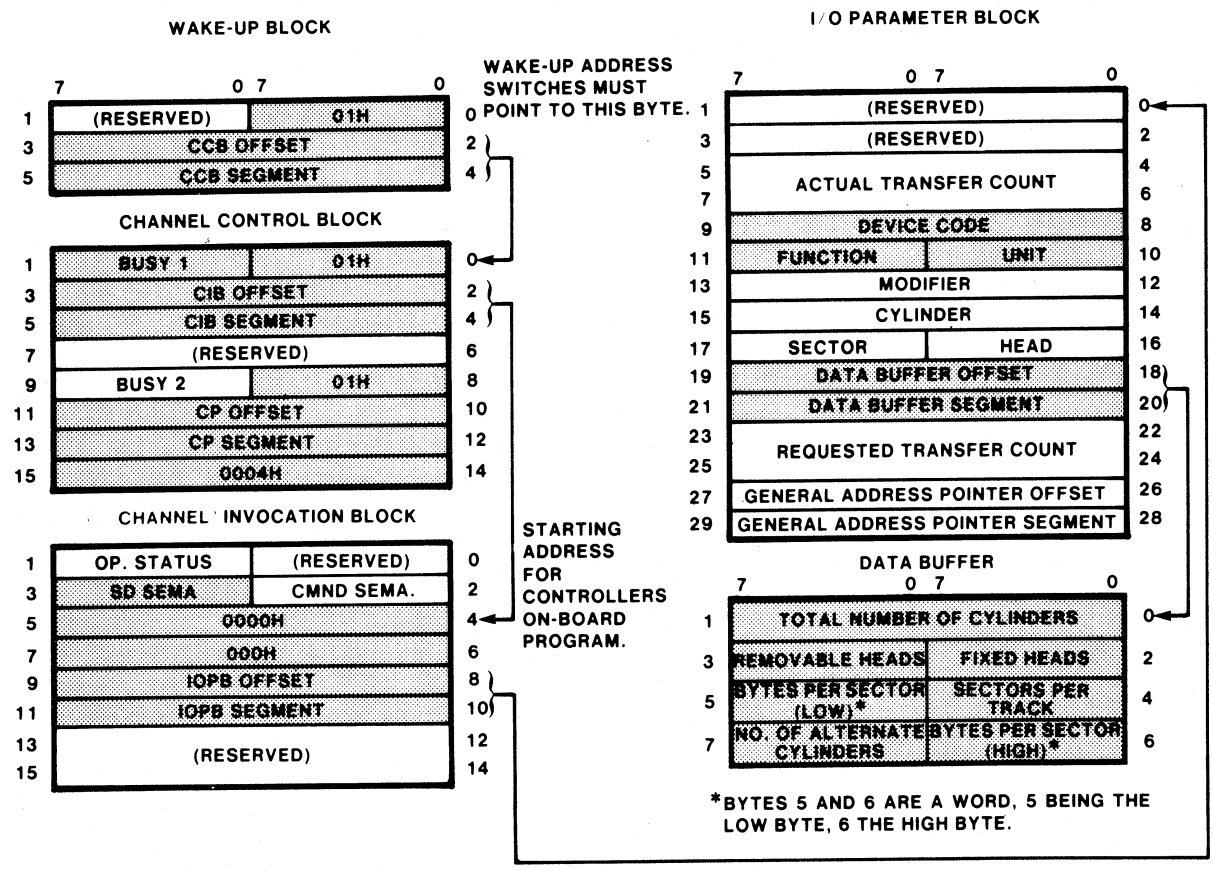


Figure 4-2. Host CPU/Board Interaction

4.2.3 Wake-Up Block

The wake-up block (see Figure 4-3 and Table 4-2) is the first of the I/O communications blocks and requires 6 bytes of memory. This block is used to establish a link between the controller and the other I/O communications blocks in the host memory.

NOTE

The wake-up block base address must be on a 16-byte boundary because the 4 least significant bits are set to zero. (Refer to section 3.4.1 for more information.)

Figure 4-1 shows how the wake-up block address is calculated. The set of jumpers that selects the I/O port address also selects the wake-up block address. The hexadecimal value defined by the jumper configuration is multiplied by 24 (shifted four places to the left) to derive the 20-bit MULTIBUS system address. When the wake-up I/O port receives the first I/O start command, the board firmware reads the wake-up block starting at this address. The firmware then fetches the wake-up block and internally saves the Channel Control Block address (the next link in the communication blocks chain). This operation is only necessary after a controller reset.

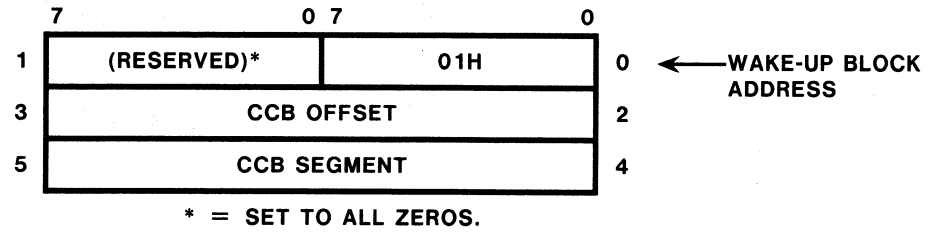


Figure 4-3. Wake-Up Block Structure

Table 4-2. Wake-Up Block Byte Contents

Byte	Function
0	SYSTEM OPERATION COMMAND (Must be set to 01H)
1	RESERVED. Set to zero
2–5	CHANNEL CONTROL BLOCK (CCB) ADDRESS Address (segment * 24 + offset) of first byte of Channel Control Block in the host system memory.

4.2.4 Channel Control Block

The Channel Control Block (CCB) indicates the status of the I/O Channel. As shown in Figure 4-4 and Table 4-3 the CCB requires 16 bytes of memory. The BUSY 1 flag (byte 1) is posted by the iSBC 214 board, (except during cold start initialization) when the board is busy processing a command and cleared by the iSBC 214 board after processing is complete. The channel control and controller invocation block addresses are stored in the iSBC 214 board memory while processing the first start I/O command to the wake-up port after a reset operation. These addresses **may not** be changed without also commanding a board reset and initialization.

Table 4-3. Channel Control Block Contents

Byte	Function
0	CHANNEL CONTROL WORD—Must be set to 01H
1	BUSY1 FLAG Indicates whether I/O port is busy or idle. 00H = Idle FFH = Busy

Table 4-3. Channel Control Block Contents (Cont'd)

Byte	Function
2-5	CONTROLLER INVOCATION BLOCK (CIB) ADDRESS—Address of byte 4 (byte 0 being the first byte) of the CIB.
The remaining bytes must be programmed as follows for iSBC 215 board compatibility. If compatible programming is not required, they should be set to zero.	
6-7	RESERVED
8	CHANNEL CONTROL WORD 2—Must contain 01H
9	BUSY 2 WORD—Not used by host CPU
10-13	CONTROL POINTER ADDRESS—Must have the address of the Control Pointer, bytes 14-15 of the CCB.
14-15	CONTROL POINTER—Must be set to 0004H.

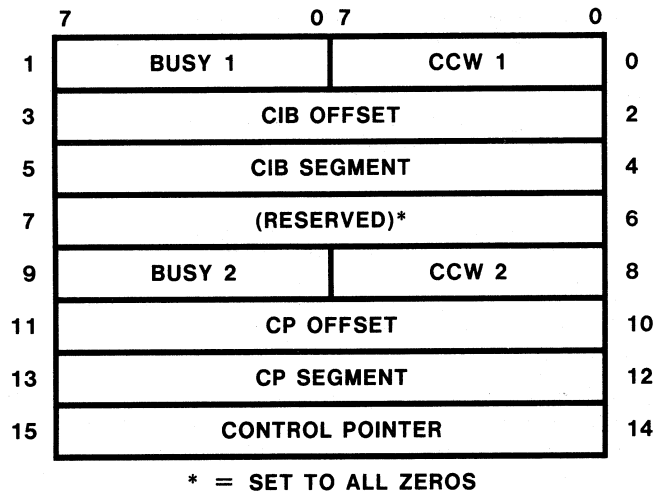


Figure 4-4. Channel Control Block Structure

4.2.5 Controller Invocation Block

The controller invocation block (see Figure 4-5 and Table 4-4) is used by the iSBC 214 board to post status to the host CPU, and issued by the host CPU to specify the starting address for the I/O parameter block. 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.

Table 4-4. Controller Invocation Block Contents

Byte	Function
0	RESERVED—Set to zero
1	OPERATION STATUS—See section 4.5 for details.
2	COMMAND SEMAPHORE—Not used by the board. Available for use as a multiprocessor interlock.

Table 4-4. Controller Invocation Block Contents (Cont'd)

Byte	Function
3	STATUS SEMAPHORE—Indicates communitcation status. Controller posts status only when this byte is 00H. When new status has been posted, board sets byte to FFH. When the host CPU has read status, it sets this byte to 00H
4–7	CSA OFFSET and CSA SEGMENT—Not used; set to zero
8–11	IOPB ADDRESS—Address of the first byte of I/O Parameter Block
12–15	RESERVED. Set to zero.

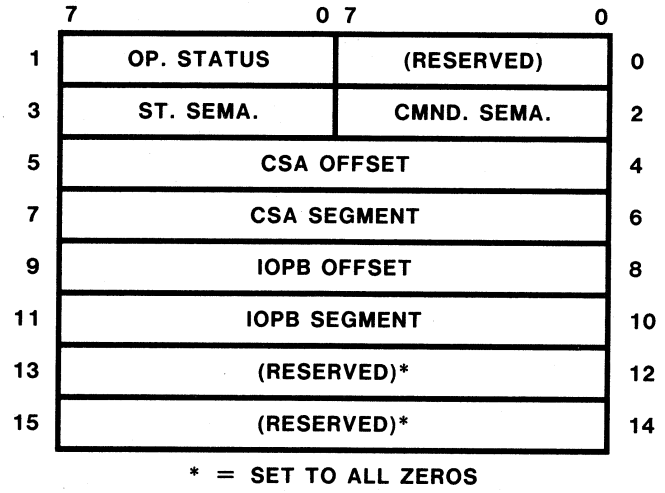


Figure 4-5. Controller Invocation Block Structure

4.2.6 I/O Parameter Block

The I/O Parameter Block (see Figure 4-6 and Table 4-5) is the primary communications link between the host system CPU and the iSBC 214 board. This communications block contains the controller operating commands, which define the function the controller is to perform (read, write, etc.) and the parameters of the function (memory address, disk head and cylinder, etc). The IOPB requires 30 bytes of host memory space.

Table 4-5. I/O Parameter Block Contents

Byte	Function
0–3	RESERVED. Set to zero.
4–7	ACTUAL TRANSFER COUNT—Count of bytes actually transferred between the system and the iSBC 214 board.
8–9	DEVICE CODE—Code for the type of device being accessed. 0000H — ST506/412 Winchester disk drive 0001H — Reserved 0002H — Reserved 0003H — 5¼-inch Flexible disk drive 0004H — QIC-02 Tape Streamer

Table 4-5. I/O Parameter Block Contents (Cont'd)

Byte	Function
10	UNIT—Number for drive being accessed. (Bits 0 and 1 provide four numbers, bits 2–7 are reserved).
11	FUNCTION—Code for the operation to be performed. Refer to Section 4.3 for operation description.
12–13	MODIFIER—Binary flags to modify functions codes. Refer to Section 4.4 for modifier field descriptions.
14–15	CYLINDER—Binary number; specifies the cylinder number for Winchester and flexible disk commands. Not used for tape commands.
16	HEAD—Binary number; specifies the head number for Winchester and flexible drive commands. Not used for tape commands. Bit 0 is least significant.
17	SECTOR—Binary number that specifies the logical sector code for Winchester and flexible disk commands. Not used for tape commands. Bit zero is least significant bit.
18–21	DATA BUFFER (DB) ADDRESS—Address of the first byte in host memory of the data (parameter) buffer. Refer to Section 4.4 for details of 24-bit addressing mode.
22–25	REQUESTED TRANSFER COUNT—Count of bytes requested to be transferred between the system and the iSBC 214. Four byte binary number, least significant bits in the first byte (Byte 22)
26–29	GENERAL ADDRESS POINTER—Not used—set to 0.

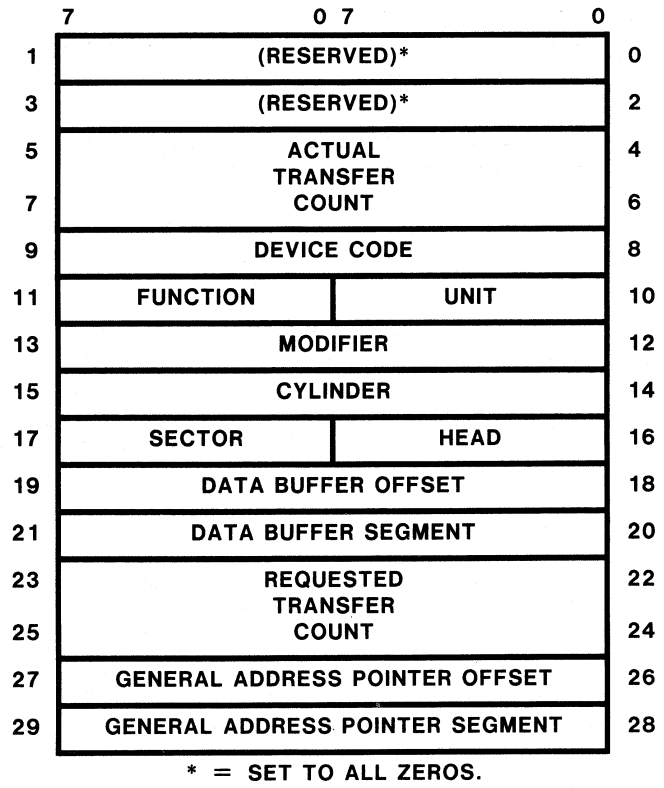


Figure 4-6. I/O Parameter Block Structure

As shown in Table 4-5, 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, the operation may have been terminated prematurely and a status check is in order. When the iSBC 214 board performs the track formatting operation, the board writes a count of 6 into the actual-transfer-count double word. When the board performs a status transfer, a count of 12 is written. When initializing hard-disk drive 0, this double word is used to display the board firmware and revision numbers. Bit 7 and 6 contain the version number minus 1; bits 5, 4, 3, and 2 contain the revision number.

4.2.7 Cold-Start Board Initialization

The iSBC 214 board cold-start initialization must be performed to establish the link between the board and the I/O communications blocks in host system memory 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.

The following procedure outlines the sequence which performs the board initializing activities. 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 3.

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

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

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

2. Sets up the bytes in the wake-up block (see Figure 4-3 and Table 4-2).
3. Sets the BUSY 1 FLAG (optional, byte 1 of the channel control block) to other than 0 (FFH). Because the iSBC 214 board resets the BUSY 1 FLAG to 0 at the completion of the cold-start operation (01H), the host CPU can monitor the flag to determine when the initialization procedure is completed.
4. Writes 02H to the wake-up I/O port to reset the iSBC 214 board.
5. Host waits at least 15ms to assure proper hardware reset of the board.
6. Writes 00H to the wake-up I/O port to clear the reset.
7. Host waits at least 15ms for the board to initialize the hardware.
8. Writes 01H to the wake-up I/O port to establish the host-CPU-to-iSBC 214-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 01H commands to the wake-up I/O port, the board reads the channel control block.

4.3 Function Commands

The iSBC 214 controller firmware includes a set of function commands for each type of peripheral device. These function commands are executed by the on-board

peripheral device controller to provide a wide range of operations. Each of the commands is invoked by setting up the communication blocks as required and issuing a START OPERATION (01H) command to the Wake-Up Port address of the iSBC 214 board.

Table 4-6 provides a comprehensive list of all the function commands available for the iSBC 214 controller including specific device applicability.

The functions available on the iSBC 214 board are divided into two general types: short-term and long-term. Short-term functions are performed with the specified device directly on-line with the controller. These functions terminate when the firmware sends a single "operation complete" interrupt to the host (if the interrupt is not suppressed). The long-term functions are initiated by the iSBC 214 board and completed off-line by the selected device. The on-line portion of the long-term function terminates when the board sends an "operation complete" interrupt to the host (if the interrupt is not suppressed). When the selected peripheral device completes the off-line portion of the function command, the iSBC 214 board sends a second interrupt to the host. (The second interrupt cannot be suppressed.)

The following description of each function command includes a list of the I/O Parameter Block (IOPB) mandatory fields required for execution. In the descriptions, only the long-term function commands are so noted.

Table 4-6. Function Command Summary

Function Command	Hex Value	Winchester Drive	Flexible Disk Drive	Streaming Tape Drive
Initialize	00	Yes	Yes	Yes
Transfer Err Status	01	Yes	Yes	Yes
Format	02	Yes	Yes	No
Read Sector ID	03	Yes	Yes	No
Read Data	04	Yes	Yes	Yes
Read to Buffer and Verify	05	Yes	Yes	No
Write Data	06	Yes	Yes	Yes
Write from Buffer	07	Yes	Yes	No
Initiate Track Seek	08	Yes	Yes	No
Buffer I/O	0E	No	No	No
Diagnostic	0F	Yes	Yes	No
Tape Initialize	10	No	No	Yes
Rewind	11	No	No	Yes
Space Foward One File	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
Tape R/W Term.	1F	No	No	Yes

4.3.1. Initialize (00H)

The Initialize function transfers the peripheral device-related parameters to the iSBC 214 board for subsequent use during the execution of other functions. To initiate this function, the host CPU establishes the following fields in the I/O Parameter Block prior to invoking this function:

Device Code	Bytes 8 and 9
Unit Number	Byte 10
Function Code	Byte 11
Modifier	Bytes 12 and 13
Data Buffer Pointer	Bytes 18 through 21

The peripheral device parameters are specified in the data buffer area and fetched by the controller (via the data buffer pointer) during function execution. Figure 4-7 illustrates the data buffer formats for the three types of devices.

To fully initialize the controller, the Initialize function must be performed for each device (present or not present). The full initialization procedure must be performed following any interruption of power, a system hardware reset, or invocation of RESET BOARD (02H) command when bypass board test modifier is not

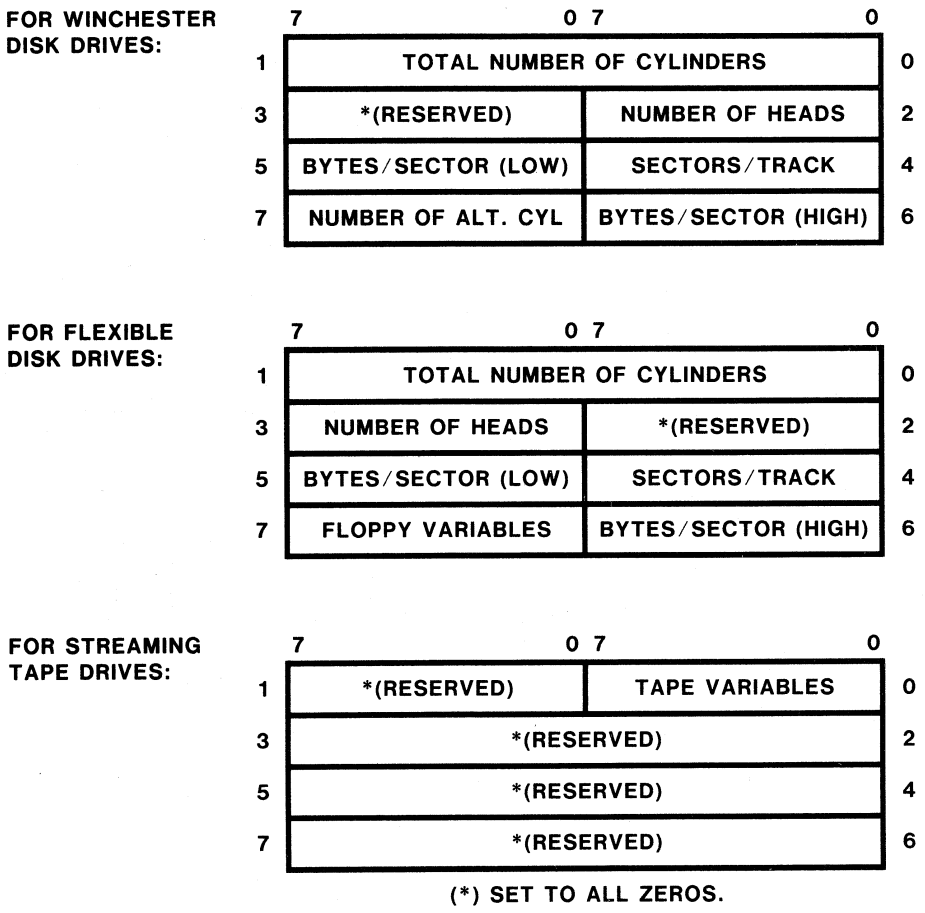


Figure 4-7. Initialize Function Data Buffer Format

set. When either Winchester disk drive or flexible disk drives are specified, the read/write heads are positioned to track 0.

4.3.1.1 Number of Cylinders

The number of cylinders is a two byte value that specifies the total number of cylinders available on the disk drive. Refer to the OEM's service manual for the particular drive to determine the correct number for this parameter. Setting the number of cylinders parameter to zero removes the specified drive from service. A drive thus removed from service can be restored to service by performing the Initialize function.

4.3.1.2 Number of Heads

The number of heads parameter is two one-byte hexadecimal values that specifies the number of available recording surfaces. Byte 2 is the number of recording surfaces on the specified Winchester drives, and byte 3 is the number of recording surfaces for the specified for flexible disk drives.

4.3.1.3 Sectors per Track

This parameter is a one byte value (in byte 4) that specifies the number of sectors available on each track of the designated drive. For Winchester disk and flexible disk drives, this value is installation and Operating System dependent and closely coupled with the "Bytes per Sector" value.

4.3.1.4 Bytes per Sector

This parameter is a two byte value that specifies the number of bytes per sector. The value is contained in bytes 5 and 6, with byte 6 as the most significant byte. The parameter value must match the formatted sector size for the specified drive. Sector sizes supported by the iSBC 214 controller are: 128, 256, 512 and 1024 bytes per sector. If the drive is not formatted, the sector size specified during formatting must match the parametric value.

4.3.1.5 Number of Alternate Cylinders

This parameter is a one byte value that specifies the number of cylinders reserved as alternates on the drive. The parameter value must match the number of cylinders reserved for alternates when the drive was formatted. If the drive is not formatted, the number of alternates during formatting must match this value.

Note that this parameter includes the two highest numbered (innermost) cylinders, namely *diagnostic* and *defective track map* cylinders. In other words, this parameter should be set to actual number of alternate cylinders plus two. (The defective track map and diagnosic cylinders.)

4.3.1.6 Flexible Disk Drive Variables

The flexible-disk-drive-variables parameter in byte 7 is a combination of values that specifies the recording format, head step rate, and head load time. Figure

4-8 illustrates the format of this parameter. When the default value is selected (000000XB) both head load and step rate times will be 30 ms.

4.3.1.7 Tape Drive Variables

The Tape Drive Variables parameter is a one byte value which specifies various tape drive related parameters. Figure 4-9 provides a detailed description of this byte.

4.3.2 Transfer Error Status (01H)

The Transfer Error Status function transfers 12 bytes of error status to the host system data buffer starting at the location specified by the data buffer pointer.

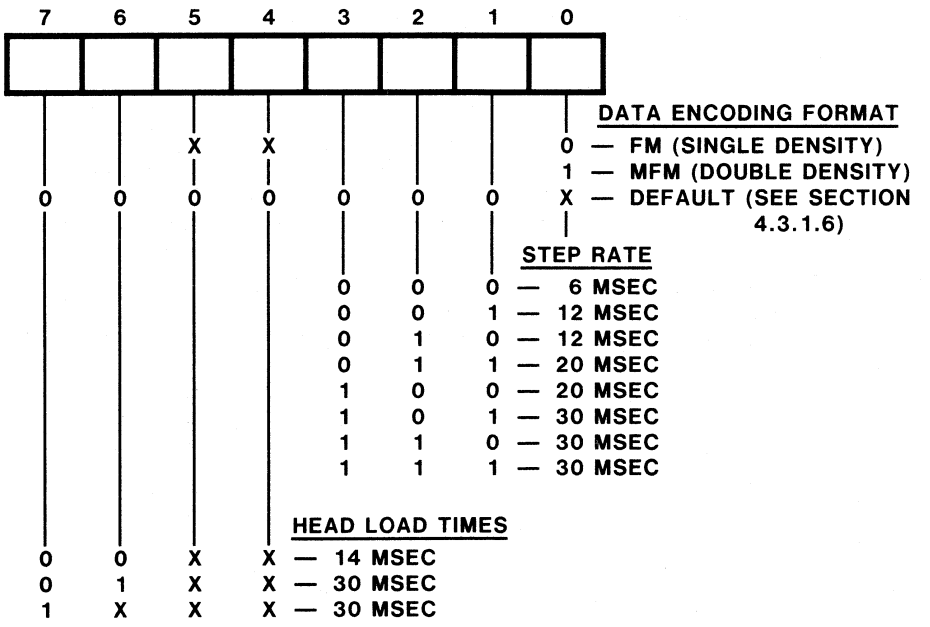


Figure 4-8. Flexible Disk Drive Initialization Byte Structure

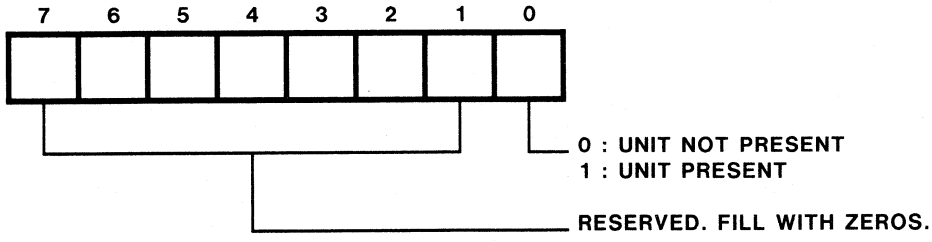


Figure 4-9. Tape Parameter Byte Structure

The host examines the status bits to determine the cause of the error. To perform this function, the host CPU establishes the following fields in the I/O Parameter Block:

Function Code	Byte 11
Modifier	Bytes 12 and 13
Data Buffer Pointer	Bytes 18 through 21

The host can request 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 0, the short-term status contents are transferred; when bit 6 is set to 1, the long-term status buffer contents are transfered. 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 can be found in Section 4.6.

4.3.3 Format (02H)

The Format function command partitions the addressed track for subsequent data recording. Figure 4-10 illustrates the format written to the Winchester drive media. Refer to Figure 4-11 for flexible disk MFM format generated by the iSBC 214 board. Figure 4-12 contains the flexible disk FM format generated by the iSBC 214 controller. The Format function writes the gaps, sector headers, and data fields on a track and reserves recording space based upon the intialization information for the specified disk drive. The command must be issued for each track that is to be formatted. The sector headers contain information used in subsequent read or write operations to locate the correct sector data area. Note that there is no alternate track, defective track map, or diagnostic track on flexible disk drives.

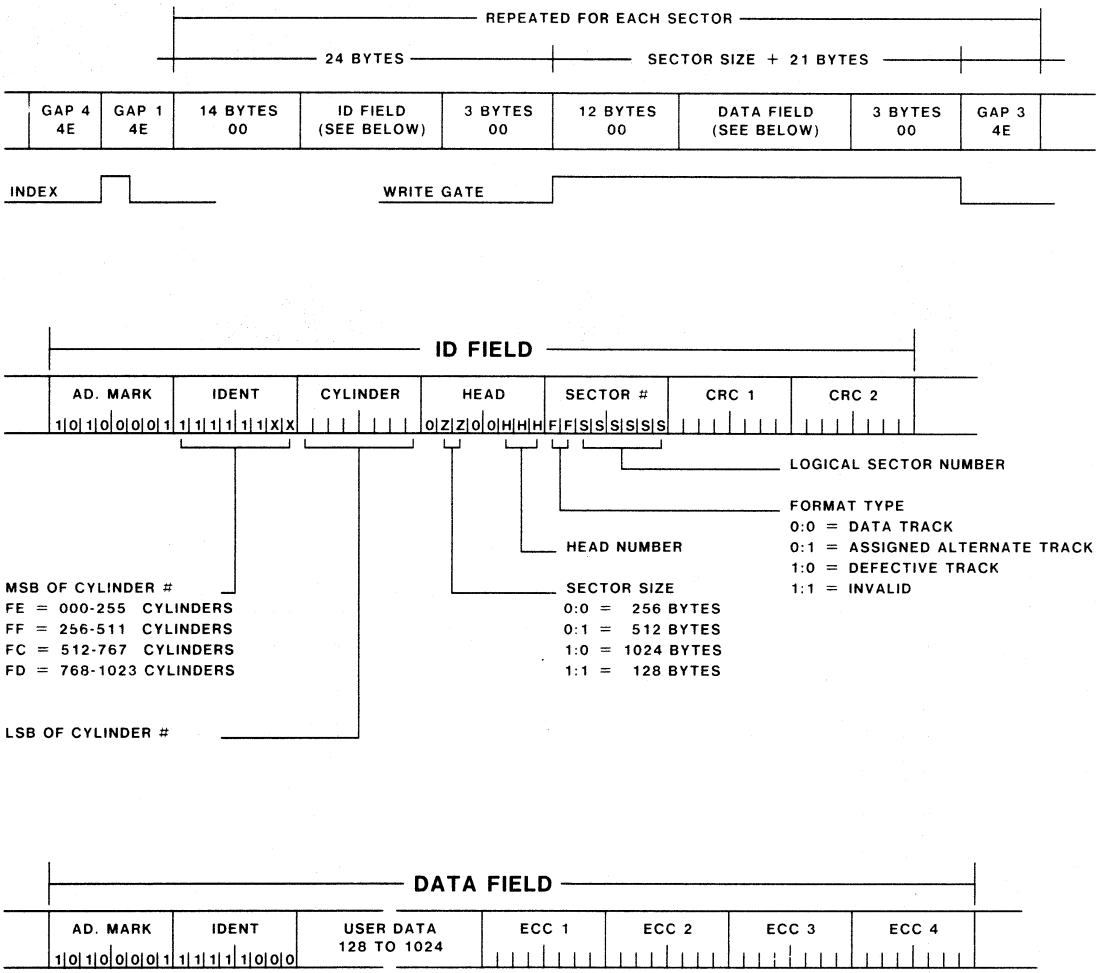
To perform this function, the host CPU sets up the following fields in the I/O Parameter Block:

Device Code	Bytes 8 and 9
Unit Number	Byte 10
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

Additional format parameters are specified in the data buffer area. Figure 4-13 shows the data buffer content for the Format function command. Byte 0 in the data buffer specifies the required format function. Most tracks are formatted as standard data tracks. When a track is determined to have media defects, the host assigns an alternate track, and reformats the original track as defective with a pointer to the assigned alternate track. The host also must format the used alternate track as an assigned alternate track. An unassigned alternate track may be formatted as a normal data track, but should not be used as one.

Bytes 1 through 4 provide a 4-byte user-specified pattern when formatting data tracks and assigned alternate tracks. The 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 media integrity. When formatting a defective track, bytes 1 and 2 specify the cylinder number and byte 3

WINCHESTER FORMAT GENERATED BY THE iSBC214 BOARD



NOTES: 1 — MOST SIGNIFICANT BIT OF ANY BYTE IS WRITTEN FIRST

2 — ADDRESS MARK = 10100001 DATA BITS WITH 00001010 CLOCK BITS

3 — ECC ACCUMULATOR PRESET VALUE = B517894A (HEX)

4 — ECC POLYNOMIAL P(X) = $X^{32} + X^{28} + X^{26} + X^{19} + X^{17} + X^{10} + X^6 + X^2 + 1$

5 — CRC POLYNOMIAL P(X) = $X^{16} + X^{12} + X^5 + 1$

6 — GAP4 SIZE = APPROXIMATELY 312 BYTES

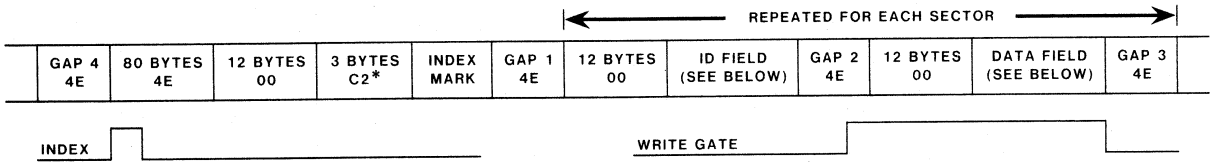
7 — GAP1 & GAP3 SIZE =

128 BYTES PER SECTOR, 54 SECTORS PER TRACK	GAP3 = 15 BYTES
256 BYTES PER SECTOR, 32 SECTORS PER TRACK	GAP3 = 15 BYTES
512 BYTES PER SECTOR, 17 SECTORS PER TRACK	GAP3 = 38 BYTES
1024 BYTES PER SECTOR, 9 SECTORS PER TRACK	GAP3 = 54 BYTES

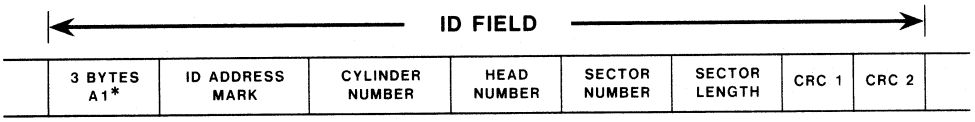
PP24

Figure 4-10. Winchester Drive Media Format

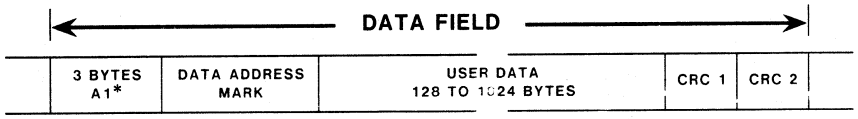
FLEXIBLE DISK DOUBLE DENSITY (MFM) FORMAT



* MISSING CLOCK PULSE BETWEEN BITS 3 AND 4



* MISSING CLOCK PULSE BETWEEN BITS 4 AND 5



* MISSING CLOCK PULSE BETWEEN BITS 4 AND 5

NOTES: 1 — INDEX MARK IS FC (HEX)

2 — ID ADDRESS MARK BYTE IS FE (HEX).

3 — NORMAL DATA ADDRESS MARK IS FB (HEX)

4 — DELETED DATA ADDRESS MARK IS F8 (HEX)

5 — CRC POLYNOMIAL $P(X) = X^{16} + X^{12} + X^5 + 1$

6 — SECTOR LENGTH FIELD IS: 01 (HEX) FOR 256 BYTE PER SECTOR
02 (HEX) FOR 512 BYTE PER SECTOR
03 (HEX) FOR 1024 BYTE PER SECTOR

7 — SECTOR SIZES vs GAP SIZES

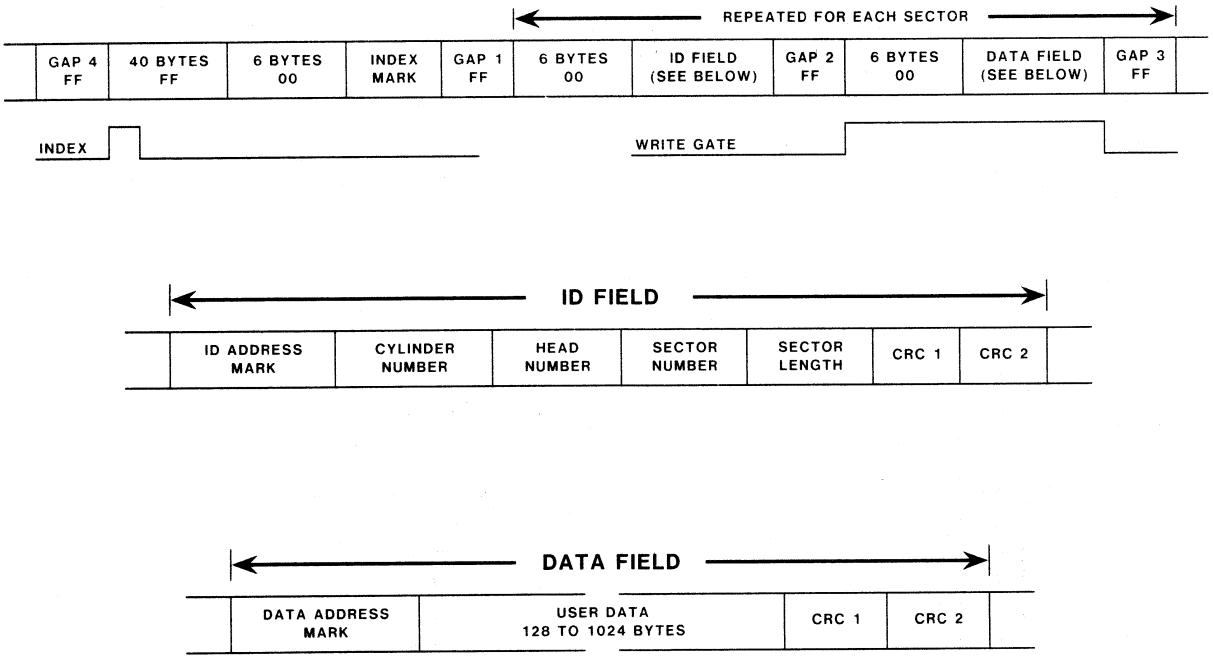
SECTOR SIZE	MAX # SECTORS PER TRACK	GAP 1 (BYTES)	GAP 2 (BYTES)	GAP 3 (BYTES)	GAP 4 (APPROX. BYTES)
256 BYTES	16	50	22	50	282
512 BYTES	9	50	22	80	310
1024 BYTES	4	50	22	240	1052

THIS TABLE IS VALID FOR 5-1/4 INCH FLEXIBLE DISKS ONLY.

PP25

Figure 4-11. Flexible Disk Media MFM Format

FLEXIBLE DISK SINGLE DENSITY (FM) FORMAT



NOTES: 1 — INDEX MARK IS 11111100 DATA BITS WITH 11010111 CLOCK BITS

2 — ID ADDRESS MARK BYTE IS 11111110 DATA BITS WITH 11000111 CLOCK BITS

3 — NORMAL DATA ADDRESS MARK IS, 11111011 DATA BITS WITH 11000111 CLOCK BITS

4 — DELETED DATA ADDRESS MARK IS, 11111000 DATA BITS WITH 11000111 CLOCK BITS

5 — CRC POLYNOMIAL $P(X) = X^{16} + X^{12} + X^5 + 1$

6 — SECTOR LENGTH FIELD IS: 00 (HEX) FOR 128 BYTE PER SECTOR
01 (HEX) FOR 256 BYTE PER SECTOR
02 (HEX) FOR 512 BYTE PER SECTOR
03 (HEX) FOR 1024 BYTE PER SECTOR

7 — SECTOR SIZES vs GAP SIZES

SECTOR SIZE	MAX # SECTORS PER TRACK	GAP 1 (BYTES)	GAP 2 (BYTES)	GAP 3 (BYTES)	GAP 4 (APPROX. BYTES)
128 BYTES	16	26	11	25	161
256 BYTES	8	26	11	48	404
512 BYTES	4	26	11	135	467
1024 BYTES	2	26	11	255	683

THIS TABLE IS VALID FOR 5-1/4 INCH FLEXIBLE DISKS ONLY.

PP26

Figure 4-12. Flexible Disk Drive FM Format

DATA TRACK:

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

ASSIGNED ALTERNATE TRACK:

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

DEFECTIVE TRACK:

	7	0 7	0
1	ALTER. CYLIN. (LOW)	80H	0
3	ALTERNATE HEAD	ALTER. CYLIN. (HIGH)	2
5	INTERLEAVE FACTOR	00H	4

Figure 4-13. Track Format Data Buffer Structure

specifies the head number for the alternate track. As with the data and alternate tracks, the content of bytes 1 through 4 are recorded into each sector data area during defective track formatting as a pointer to the assigned alternate track.

Byte 5 in the data buffer specifies the sector interleave factor for the track to be formatted. The interleave factor normally 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 logically sequential sectors.

Also note that the iSBC 214 controller reads the requested sector and all the logically sequential sectors on that track for track buffering purposes. The actual interleave factor on the media is very important for this reason. Any interleave factor other than "1" would reduce the performance significantly.

To avoid this possibility, and also to be able to use existing software, the iSBC 214 controller board always uses interleave 1 during format regardless of the interleave factor specified in byte 5.

4.3.4 Read Sector ID (03H)

The Read Sector ID function searches for the first error-free sector header on the presently selected cylinder and head. When the header is located, the function 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 10
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 214 board during function execution. Figure 4-14 illustrates the data buffer and flag byte format.

4.3.5 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

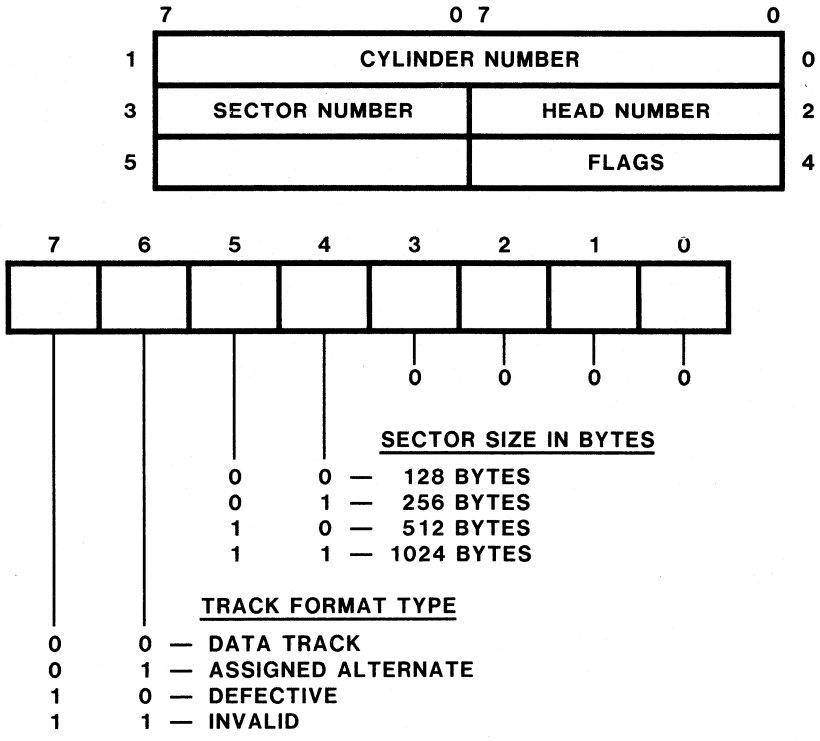


Figure 4-14. Read-Sector-ID Function Data Buffer and Flag Byte

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 10
Function Code	Byte 11
Modifier	Bytes 12 and 13
Cylinder Number	Bytes 14 and 15
Head Number	Byte 16
Starting Sector No.	Byte 17
Data Buffer Pointer	Bytes 18 through 21
Requested Transfer Count	Bytes 22 through 25

If multi-sectored data transfers are requested, the controller automatically reads the next sector. If end-of-track is reached, the operation resumes from the next head and sector-0 (sector-1 on flexible disks). If end-of-cylinder is reached, the operation resumes from the next cylinder, head-0 and sector-0 (sector-1 on flexible disks).

4.3.5.1 Winchester Disk Read Details

A Winchester read data function can be initiated to read any number of bytes. The controller board always reads a full sector for proper ECC checking and correction, but transfers only the requested number of bytes to the host memory.

When the Winchester disk Read Data function is initiated, the controller firmware scans the cache (track buffer) entries to check if the requested sector (and the consecutive sectors on that track) are in the cache already. If the requested sector is found in the cache (cache hit), the data is transferred to the host memory directly from the track buffers, without any Winchester disk access.

If the requested sector is not found in the cache, the controller firmware replaces the contents of the least recently used (LRU) cache entry with this request and compares the current location of the heads with the requested cylinder numer. If the two are not the same, the firmware initiates a seek to the requested cylinder (an implicit seek). When the requested cylinder is reached and the requested head is selected, the iSBC 214 controller board reads the requested sector and all the consecutive sectors on that track to the local cache buffer. A rotational position sensing mechanism is used to guarantee minimum read time from the Winchester disks. Worst case track read time is one revolution time (16.66 msec). Note that on a singe sector read, the requested sector data is transferred from cache buffer to the host memory as soon as that sector is read from the disk. On the multi-sector read, however, the data transferred to the host memory is delayed until the track (partial or full) is cached.

Multi-track read requests are handled track at a time, as described above, until the requested transfer count is met, or an exception is encountered.

4.3.5.2 Flexible Disk Read Details

A flexible disk read data function can be initiated to read any number of bytes, but the controller reads full sectors for proper CRC checking and transfers only the requested number of bytes to the host memory.

When the flexible disk read data function is initiated, the controller firmware allocates a temporary local storage area of sector size length and compares the current location of the heads with the requested cylinder number. If the two are not

the same, the firmware initites a seek to the requested cylinder (an implicit seek). When the requested clyinder is reached and the requested head is selected, the iSBC 214 controller reads the requested sector to its local temporary buffer. Once the sector is read and the CRC check is performed, the controller transfers the data from the temporary buffer to the host system memory.

Multi-sector read requests are handled sector at a time, as described above, until the requested transfer count is met, or an exception is encountered.

4.3.5.3 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 a complete operation. A complete tape read operation consists of one or more tape read functions that must be 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 disk or flexible disk) can be interleaved with subsequent tape read functions. Note: The only Tape function permitted prior to the termination of the Tape Read Operation 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:

— **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.

— **File Mark Termination**—The tape read operation termination results when a file mark is encountered. 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 the file mark detected.

If the iSBC 214 controller fails to maintain the data transfer rate required by the tape drive (typically 90 kilobytes per second), an over-run occurs. The iSBC 214 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.

4.3.6 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 214 board RAM buffer and checks each sector read for an error correcting code (ECC for Winchester disks, CRC for flexible disks). 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 10
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

By specifying one disk device for the read-to-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, and intervening iSBC 214 board functions cannot be executed.

When this function is performed on Winchester drives, the track accessed is cached. If the track accessed was already in cache, the controller board reads the requested sector and the consecutive sectors in that track into the same cache buffer.

4.3.7 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 10
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

4.3.7.1 Winchester Disk Write Details

A Winchester write data function can be initiated to write any number of bytes. If the requested transfer count is not an integer multiple of sector size, then the iSBC 214 controller board fills the rest of the last sector with zeros.

When the Winchester disk write data function is initiated, the controller firmware scans the cache entries for the desired track. If the sector (or sectors) to be written are in cache, the controller board will use this cache buffer as a temporary storage area, otherwise the controller board will use the least recently used cache

buffer as a temporary storage area. Once the temporary storage area is determined, the controller transfers the data from host memory to the temporary area and compares the current location of the heads with the requested cylinder number. If the two are not the same, the firmware initiates a seek to the requested clyinder (implied seek). When the requested clyinder is reached and the requested head is selected, the iSBC 214 controller board writes the sector (sectors) in the temporary storage to the media.

4.3.7.2 Flexible Disk Write Details

A flexible disk write data function can be initiated to write any number of bytes. If the requested transfer count is not an integer multiple of the sector size, the iSBC 214 controller board fills the rest of the sector with zeros.

When the flexible disk write data function is initiated, the controller firmware allocates a temporary storage area of sector size length, transfers data from the host memory to this temporary area, and compares the current location of the heads with the requested cylinder number. If the two are not the same, the firmware initiates a seek to the requested cylinder (implicit seek). When the requested cylinder is reached and the requested head is selected, the iSBC 214 controller board writes the sector in the temporary buffer area to the flexible disk media.

Multi-sector write requests are handled sector at a time, as described above, until the requested transfer count is met or an exception is encoutered.

4.3.7.3 Tape Drive Write Operation

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

The Tape Write Data function is always a complete operation. A complete Tape Write operation consists of one or more tape write functions that are 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 Winchester 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 a file mark.

If the iSBC 214 board fails to maintain the data transfer rate required by the tape drive (typically 90 kbytes per second), an underrun occurs, the iSBC 214 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 underrun indicates that tape repositioning was required with a buffer underrun/overrun error posted in the short-term status buffer.

4.3.8 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-buffer-data 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
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

NOTE

Prior to the execution of this function, a *single sector* "read to buffer and verify" (05), or "buffer I/O" (0E) function must be executed to preload the sector buffer. The use of a multi-sector "read to buffer and verify" function or any other function other than those listed above, makes the data written by the write buffer Data function indeterminate.

4.3.9 Initiate Track Seek (08H)

This function positions the Read/Write heads of the specified Winchester or flexible disk drive to the specified track without initiating a data transfer. To perform the initiate-track-seek function, the host system 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
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 214 board is unavailable to the host CPU until the selected operation is completed.

The iSBC 214 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 214 board posts the seek-complete status and sends a second interrupt to the host CPU. (The second interrupt cannot be suppressed.)

If the iSBC 214 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 214 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 214 board is limited to one active short-term function

while an off-line seek operation(s) is 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 0, and an invalid-address error is posted in the short-term status buffer.

4.3.10 Buffer I/O (0EH)

The Buffer I/O function transfers a block of data between the system memory and the iSBC 214 on-board RAM. To perform the Buffer I/O function, the host system CPU establishes the following fields in the I/O Parameter Block:

Unit Number	Byte 10
Function Code	Byte 11
Modifier	Bytes 12 and 13
On-board Memory Pointer	Bytes 14 and 15 (cylinder no.)
I/O Specifier (Input = 00H, 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 214 board RAM and a system memory buffer. This function is used for diagnostic purposes, or to fill the iSBC 214 board buffer for a subsequent write-buffer-data function.

This function redefines the I/O Parameter Block structure slightly to accommodate passing the starting controller RAM address and the transfer direction to the board. The board memory address pointer is specified in bytes 14 and 15 of the I/O parameter block and must be in the range of 4000H to 43FFH. 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 a short-term operation and includes a single interrupt at the completion of the function.

4.3.11 Diagnostic (0FH)

The diagnostic function initiates a go/no-go self test, resident in the iSBC 214 firmware. The self test verifies internal data and status logic in the peripheral devices.

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 10
Function Code	Byte 11
Modifier	Bytes 12 and 13

The diagnostic function reserves the use of the highest cylinder number on head 0. 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 additional modifiers (specified in modifier byte 13) specifically apply to the diagnostic function.

00H — **(Winchester disk function)**—Recalibrates, then initiates a seek to the highest clyinder number of head 0. At seek completion, the iSBC 214 board performs a read-sector-ID function to verify track location before performing a write and read test on sector 0 using a 55AAH data pattern.

00H — **(Flexible disk function)**—recalibrates, then initiates a seek to the highest cylinder number of head 0. At seek completion the iSBC 214 board performs a read-data function on sector 1. Note that since there is no dedicated diagnostic track on flexible disk, the diagnostic function is limited to a non-destructive read only test.

01H — Initiates a ROM check-sum test to verify the contents of the iSBC 214 board ROM.

02H — Initiates a seek to cylinder 0. At seek completion, the iSBC 214 board performs a read-sector-ID function to verify that the heads are located at cylinder 0.

4.3.12 Tape Initialize (10H)

The Tape Initialize function is the second operation in a four-step initialization process for the specified tape drive. The initialization process is starts with the initialize function (00H), followed by the tape initialize function, the tape reset function (1CH), and finally the load tape (18H) function. 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 10
Function Code	Byte 11

Initializing the tape drives attached to the iSBC 214 board is a four-step process. In the first step, the host CPU performs the initialize function (00H) to initialize the iSBC 214 board. In the second step, the iSBC 214 board initializes the Tape Drive Interface circuitry. In the third step, the tape reset, the tape drive is initialized. Finally, in the fourth step, the load-tape function is performed to position the tape to the beginning of tape (BOT) for subsequent operations. The tape-initialize function is classed as a short-term operation, and includes a single interrupt at the completion of the function.

4.3.13 Rewind (11H)

The rewind function returns the tape on the specified drive to the beginning-of-tape marker. To perform this long-term 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

Typically, the tape cartridge is rewound for one of two reasons: to return the tape to its starting point prior to removing the cartridge from the drive unit, or to position the tape to a known point before attempting a data transfer.

The iSBC 214 instructs the tape interface circuitry to perform an off-line rewind on the specified tape drive. Once the operation commences, the iSBC 214 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 214 controller board firmware sends a second interrupt to the host CPU.

To examine the 12-byte status buffer relative to the off-line rewind operation, the host processor must request the contents of the long-term status buffer. If the controller is busy with another device when the off-line rewind operation completes, it completes the current 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 controller firmware posts the rewind-complete status and sends an interrupt to the host CPU indicating completion of the off-line operation.

4.3.14 Space Forward One File Mark (12H)

This long-term function moves the tape forward until a file mark or end-of-media is reached. 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 10
Function Code	Byte 11

The iSBC 214 directs the tape interface circuitry to perform an off-line tape movement on the specified tape drive. Once the operation begins, the iSBC 214 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 reaches a file mark or detects end-of-media and completes the function, the controller firmware sends a second interrupt to the host CPU.

The host CPU must request the contents of the long-term status buffer in order to examine the 12-byte status buffer relative to the function. As with the Rewind (11H) function, if the board is busy with another device when an off-line operation completes, the controller finishes with the current device, posts the appropriate status, and sends an interrupt (if not suppressed) to the host CPU. After the host CPU services this interrupt and resets the status semaphore, the iSBC 214 controller board firmware posts the operation-complete status and sends an interrupt to the host CPU indicating function completion.

4.3.15 Write File Mark (14H)

The write-file-mark function, a short-term function, allows writing additional file marks at the end of the 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 this 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

The write-file-mark function writes a file mark on the tape at the position of the write head at the time of the function 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.

4.3.16 Erase Tape (17H)

The erase-tape function prepares the tape for subsequent recording by removing all existing recorded data. To perform this long-term 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

Whenever a write data operation is performed, the tape written is also erased automatically. However, there is some risk that this erase operation may leave some background noise in the newly written area. Also, any data from a previous recording in the area beyond the subject area are not erased. This 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 214 board directs the tape interface circuitry to perform an off-line erase operation to the designated tape drive. Once the operation begins, the controller firmware terminates the function, posts the operation-complete status and sends an interrupt (if not suppressed) to the host CPU. When the tape drive completes the function, the controller firmware sends a second interrupt to the host CPU.

The host CPU must request the contents of the long-term status buffer in order to examine the 12-byte status buffer relative to the function. As with the Rewind (11H) function, if the board is busy with another device when an off-line operation completes, it completes the current device operation, posts the appropriate status, and sends an interrupt (if not suppressed) to the host CPU. After the host CPU services this interrupt and resets the status semaphore, the controller firmware posts the operation-complete status and sends an interrupt to the host CPU indicating function completion.

4.3.17 Load Tape (18H)

The load-tape function is the fourth operation in the tape initialization sequence and positions the tape to the beginning-of-tape marker. This long-term function also initiates the on-going controller check for tape media change. 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 10
Function Code	Byte 11

The iSBC 214 board accomplishes the load-tape function by directing the tape interface circuitry to perform an off-line operation on the specified tape drive. Once the operation is started, the iSBC 214 board terminates the load-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 off-line rewind, the iSBC 214 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 load-tape operation, it must request the contents of the long term status buffer. If the iSBC 214 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 214 board sends the load-complete status and again sends an interrupt to the host CPU, this time to indicate completion of the off-line load-tape operation.

4.3.18 Tape Reset (1CH)

The tape-reset function, a long-term function, is the third function in the tape initialization sequence and initializes the tape drive. To perform this function, the host system CPU sets the following fields in the I/O parameter block:

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

4.3.19 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 back to the beginning-of-tape marker. This restacks the tape in the cartridge and ensures unobstructed tape movement. 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 10
Function Code	Byte 11

The iSBC 214 directs the tape interface circuitry to perform an off-line retensioning of the selected tape drive. Once the operation commences, the controller firmware terminates the Retension function, posts the operation-complete status, and sends an interrupt (if not suppressed) to the host CPU. When the tape drive completes the retension operation, the controller firmware sends a second interrupt to the host CPU.

The host CPU must request the contents of the long-term status buffer in order to examine the 12-byte status buffer relative to the function. As with the Rewind (11H) function, if the controller is busy with another device when an off-line operation completes, it completes the current device operation, posts the appropriate status, and sends an interrupt (if not suppressed) to the host CPU. After the host CPU services this interrupt and resets the status semaphore, the on-board firmware posts the operation-complete status and sends an interrupt to the host CPU indicating function completion. This operation is classed as a long-term function.

4.3.20 Read Tape Status (1EH)

The read-tape-status function transfers the existing status from the tape drive to the iSBC 214 board short-term status buffer. This function is only useful when the host CPU wants to know the current status of the tape drive, typically for debugging purposes. 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 10
Function Code	Byte 11

4.3.21 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

4.4 Function Modifiers

The function modifiers allow the user to change easily the default functions and tailor the iSBC 214 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 4-15 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 214 board to suppress assertion of the interupt 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 214 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.

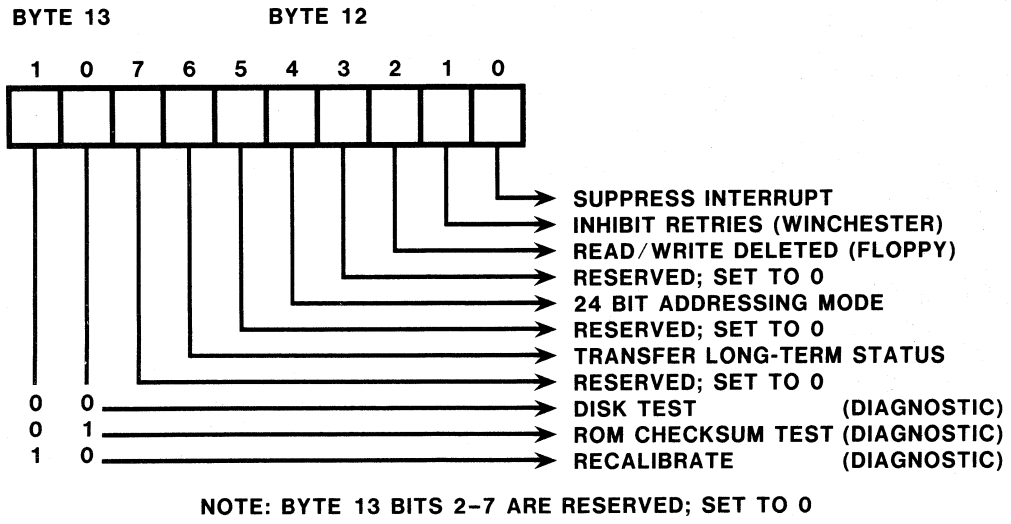


Figure 4-15. Modifier Word Format

Inhibit Retries—(Winchester Disk Only)—The inhibit-retries modifier bit, when set to 1, directs the board to attempt only once to complete a data transfer function command.

Read/Write Deleted Data—The read/write deleted data modifier bit, when set to 1, converts the flexible disk read and write functions to read and write deleted data (record) functions.

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 0H through 7H, byte 19 provides address bits 8H through FH, and byte 20 provides address bits 10H through 17H. Byte 21 is set to zero.

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 a 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 status buffer contents should be transferred first. The long-term-status-buffer is valid only for tape operations.

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 214 board ROM.

Restore to Cylinder 0—The restore-to-cylinder-0 modifier bit, when set to 1, limits the diagnotic function to restoring the read/write heads to cylinder 0.

4.5 Extended Status

At the end of each controller operation, information pertaining to the operation is stored in one of two status buffers provided in the on-board RAM. If the completed operation was short-term function, only the short-term status is posted; if the completed operation was a long-term function, both the short-term and the 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 214 board summarizes the status buffer into the operation status byte.

Figure 4-16 illustrates the format of the operation status byte. Bits 0 through 3 contain the code for a device-specific status summary. Bits 4 and 5 specify the device unit number. Bit 6 indicates (when set to 1) that a hard error occurred on the designated device and the specified function could not be executed. Bit 7 (when set to 1) indicates that some type of error has occurred. If bit 7 is set to 1 and bit 6 is set to 0, a soft (recoverable) error has occurred.

4.6 Status Operations

Whenever the operation status byte indicates that an error occurred, the host processor can request the appropriate status block, short-term or long-term, for additional error information. This information is stored in the on-board status buffer, the contents of which are described in the following paragraphs.

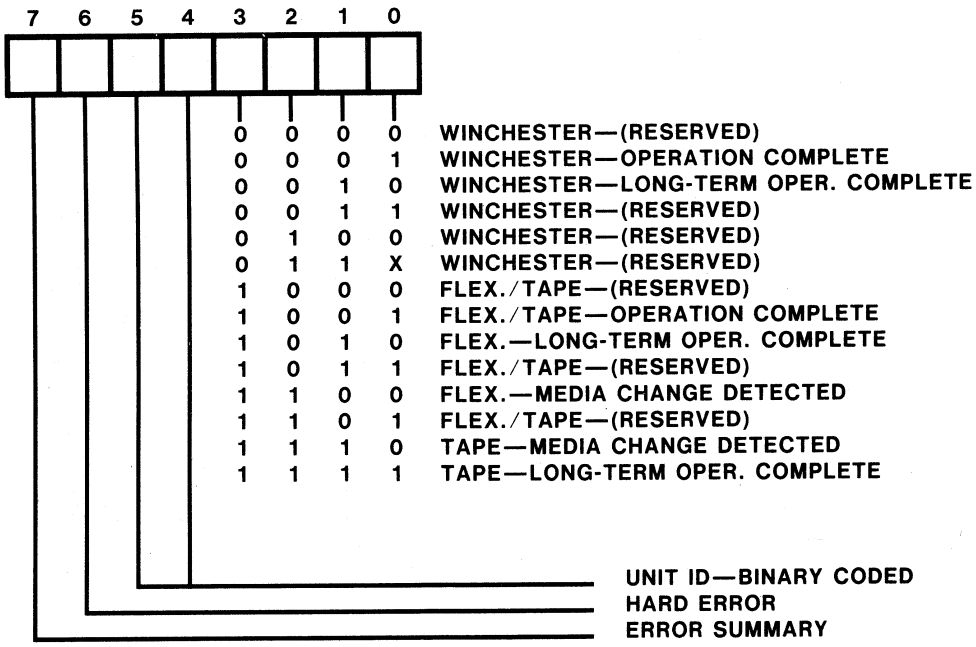


Figure 4-16. Operation Status Byte Format

4.6.1 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. Table 4-7 lists the format of the status buffer for both disk and tape operations.

Table 4-7. Error Status Buffer Format

Byte	Disk Function	Tape Function
0	Detailed Status Byte	Detailed Status Byte
1	Detailed Status Byte	Detailed Status Byte
2	Detailed Status Byte	Detailed Status Byte
3	Desired Cylinder(low)	Beginning of Tape Marker Detected
4	Desired Cylinder(high)	Logical Load Point Detected
5	Desired Head and Volume	File Mark Detected
6	Desired Sector	Logical EOT Detected
7	Actual Cylinder (low)	Not Used
8	Flags, Act. Cyl.(high)	No Data Detected (Blank Tape)
9	Actual Head	Not Used
10	Actual Sector	Not Used
11	Number of Retries	Not Used

The definitions of each byte in the status buffer are as follows:

- Bytes 0 through 2 for both disk and tape functions contain the detailed 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 tape functions), when set to 0FFH, indicates the beginning of tape marker was detected and the tape is positioned at the start of the recording area on the tape.
- Byte 4 (for tape functions), when set to 0FFH, indicates 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 tape functions), when set to 0FFH, indicates the tape drive encountered a file mark during function execution.
- Byte 6 (for tape functions), when set to 0FFH, indicates the tape drive encountered the logical end-of-tape marker during the execution of the last function.
- Bytes 7 through 10 (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 4-17 illustrates byte 8 of status buffer for disk drive operations.
- Bytes 7, 9, 10, and 11 are not used for tape drive functions and are returned as all zeros.
- Byte 8 (for tape functions), when set to 0FFH, indicates 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 214 board or the specified drive unit.

4.6.2 Detailed Error Status

The first three bytes of the status buffer provide detailed status, called error status, relating to the last function completed. If any bit in these first three status

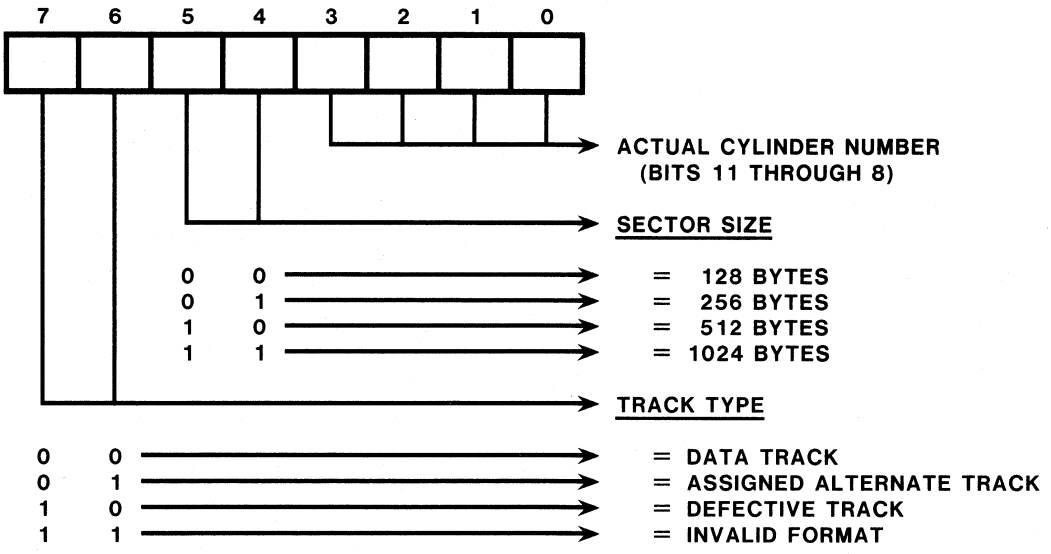


Figure 4-17. Status Buffer Format, Byte 8 (for Disk Drive)

bytes is set to 1, bit 7 in the operation status byte is set to report the occurence of an error. Bit 6 in the operation status byte reports the type: 1 = hard error, 0 = soft error. The definitions of the status bits for disk functions and tape functions are similar, not identical. In the following descriptions of the status bits, both disk and tape device definitions are included.

4.6.2.1 Status Byte 0

Figure 4-18 illustrates the format of status byte 0. A functional description of each bit appears after the figure.

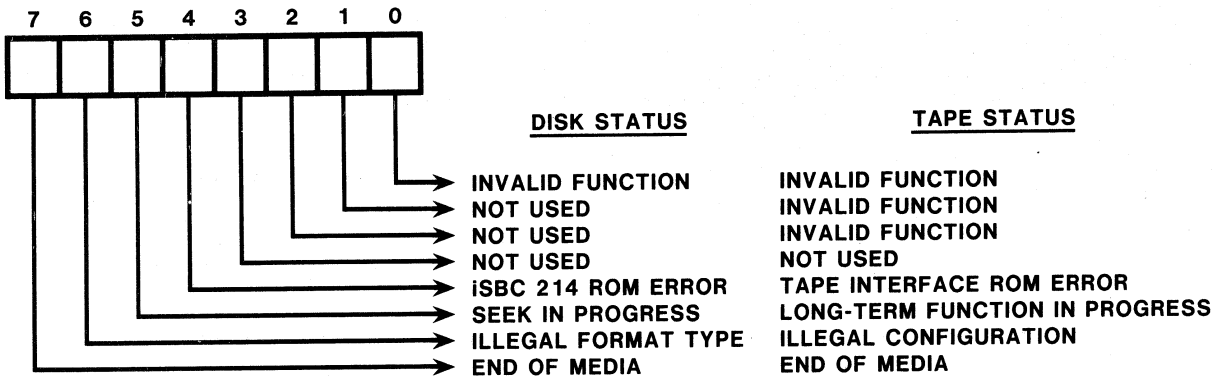


Figure 4-18. Status Buffer Format, Byte 0

Bit 0 Invalid iSBC 214 Board Function—Bit 0 set to 1 indicates the function code in byte 11 of the I/O parameter block was not one of the defined function codes.

Bit 1 Invalid Tape Interface Function—Bit 1 set to 1 indicates the tape function code in byte 11 of the I/O parameter block could not be executed by the tape interface circuitry. Bit 1 is always set to 0 when disk functions are executed.

Bit 2 Invalid Tape Drive Function—Bit 2 set to 1 indicates the tape function code in byte 11 of the I/O parameter block could not be executed by the tape drive. Bit 2 is always set to 0 when disk functions are executed.

Bit 4 iSBC 214 Board ROM Error (Disk Function)—Bit 4 set to 1 indicates the board failed the ROM checksum portion of the internal diagnostic program.

Bit 4 iSBC 214 Tape Interface ROM Error (Tape Function)—Bit 4 set to 1 indicates the iSBC 214 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 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 there was an off-line function in progress in the tape drive when the initiation of another function was attempted with the same drive.

Bit 6 Illegal Format Type (Disk Function)—Bit 6 set to 1 indicates 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 directly access an assigned alternate track as a primary data track. 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 the use of an illegal value.

Bit 6 Illegal Configuration (Tape Function)—Bit 6 set to 1 indicates 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 sets bit 6 to 1.

Bit 7 End of Media—Bit 7 set to 1 indicates the end of media marker was detected before the requested transfer count in the I/O parameter block was satisfied.

4.6.2.2 Status Byte 1

Figure 4-19 illustrates the format of status byte 1; the paragraphs that follow describe the bits in this status byte.

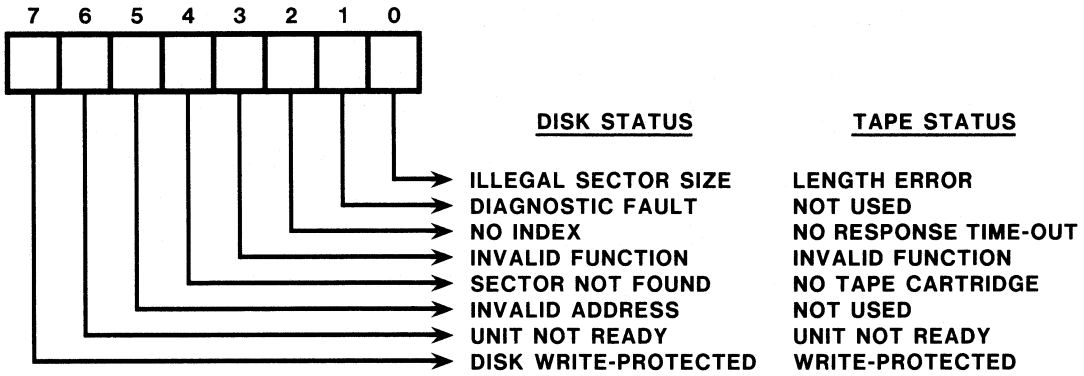


Figure 4-19. Status Buffer Format, Byte 1

Bit 0 Illegal Sector Size (Disk Function)—Bit 0 set to 1 indicates 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 0 Length Error (Tape Function)—Bit 0 set to 1 indicates one of the following conditions:

1. The data transfer function specified a requested transfer count of 0.
2. A file mark was detected before the requested transfer count was satisfied.
3. The data transfer function was terminated by the iSBC 214 board.

Bit 1 Diagnostic Fault (Disk Function)—Bit 1 set to 1 indicates the iSBC 214 controller and the disk drive failed execution of the internal diagnostic program. (Although not strictly a disk function, the diagnostic fault function is so classified.) Bit 1 is always set to 0 when tape functions are executed. This bit, when set to 1, also indicates a possible hardware malfunction on the Winchester and/or flexible disk controller subsection of the iSBC 214 board, detected by the firmware as a response time-out from these hardware subsections.

Bit 2 No Index (Disk Function)—Bit 2 set to 1 indicates the controller did not receive an index pulse from the disk drive. This error indicates typically that the specified drive is not attached to the iSBC 214 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 the tape interface circuitry 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 0 was set to 1.

Bit 4 Sector not Found (Disk Function)—Bit 4 set to 1 indicates that the iSBC 214 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 214 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 media, installed in the selected device, that was mechanically write-protected.

4.6.2.3 Status Byte 2

Figure 4-20 illustrates the format of status byte 2; the following paragraphs describe the bits of this status byte.

Bit 0 Not used; set to 0.

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 iSBC 214 board was successfully completed, but that one or more retries 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 0 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 iSBC 214 during a data transfer. Bit 2 is always set to 0 when disk functions are executed.

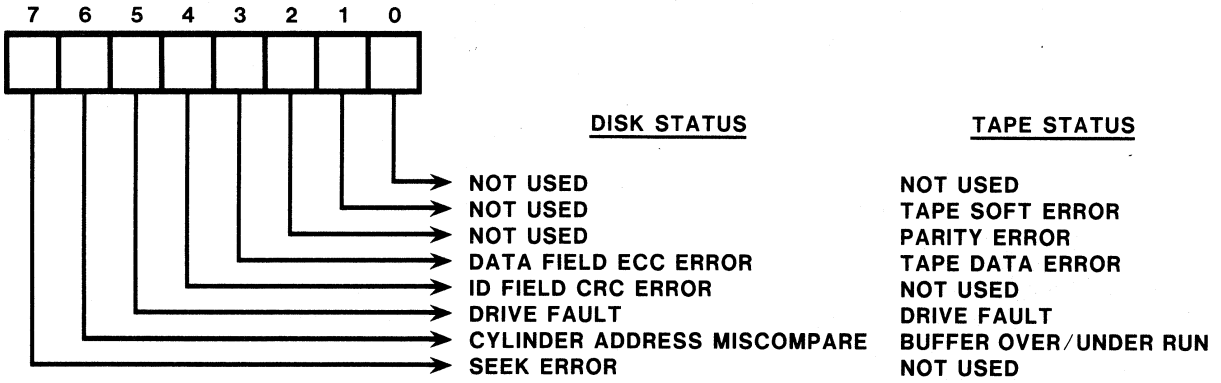


Figure 4-20. Status Buffer Format, Byte 2

Bit 3 Data Field ECC error (Winchester Disk function)—bit 3 set to 1 indicates that the iSBC 214 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 0, the error was soft and recoverable (by either retries or correction algorithm); if bit 6 is set to 1, the error was hard and not recoverable.

Bit 3 Data Field CRC error (Flexible Disk function)—Bit 3 set to 1 indicates that the iSBC 214 board detected an error in the data field during a Cyclic Redundency Check (CRC) in one of the sectors transferred. If bit 6 of the operation status byte is set to 0, the error was soft and recoverable; if bit 6 is set to 1, the error was hard and is not recoverable.

Bit 3 Tape Data Error (Tape Function)—Bit 3 set to 1 indicates the controller detected a data error in the file that was transferred and that the error was hard and unrecoverable.

Bit 4 ID Field CRC error (Disk Function)—Bit 4 set to 1 indicates that the iSBC 214 board detected an error in the ID field during a Cyclic Redundency Check (CRC) in one of the sectors transferred. If bit 6 of the operation status byte is set to 0, the error is soft and is recoverable by retries; if bit 6 is set to 1, the error is hard and not recoverable. Bit 4 is always set to 0 when tape functions are executed.

Bit 5 Drive Fault—Bit 5 set to 1 indicates there is a hardware problem in the selected drive.

Bit 6 Cylinder Address Miscompare (Disk Function)—Bit 6 set to 1 indicates that the read/write heads were positioned to the incorrect cylinder. The recovery process for this error is to perform a seek to track 0 and then attempt to seek to the desired track.

Bit 6 Buffer Over/Under Run (Tape Function)—Bit 6 set to 1 indicates that the data transfers from the iSBC 214 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 recalibrate procedure and initiates a seek unless retries are inhibited. Bit 7 is always set to 0 when tape functions are executed.

4.7 Interrupts

The iSBC 214 board generates interrupts to alert the host CPU of significant changes in mass storage system status by asserting any of the eight MULTIBUS interface interrupt lines (INT0 through INT7). The iSBC 214 board posts interrupts to the host CPU for three conditions: operation-complete, seek-complete, media-change.

The interrupt on operation-complete is disabled by entering a 1 in bit 0 of the modifier word in bytes 12 and 13. The seek-complete and media-change interrupts cannot be disabled. Once an interrupt is asserted, it can be cancelled in one of three ways:

- Host CPU generates a Clear I/O command (00H) to the controller's Wake-Up port.
- Assertion of the MULTIBUS interface INIT signal by the system.
- Power-on reset to the system.

Interrupt priority level selection in the range of 0 through 7 is user selectable through 12 jumperable stake pins. The default is interrupt 5. Detailed jumper information is contained in Chapter 3.

CHAPTER 5
INTERFACING INFORMATION

5.1 Introduction

This chapter provides detailed information about the peripheral device and system interfaces supported by the iSBC 214 board. The descriptions of the interfaces include signal mnemonic identification and relative pin assignments. The iSBC 214 board has four independent interfaces that follow industry standards and adhere to the electrical and mechanical specifications of each interface. The four interfaces are:

- MULTIBUS System Interface (with optional 24-bit addressing)
- ST506/412 Winchester 5.25-inch disk drive
- SA450/460 5.25-inch flexible disk drive
- QIC-2 ¼-inch streaming tape drive

5.2 MULTIBUS® Interface

The MULTIBUS interface subsystem consists of the MULTIBUS interrupt port, iSBC 214 controller Wake-Up Port, and the address and data interface. The iSBC 214 board provides two standard PCB edge connnectors, P1 and P2, that plug into female MULTIBUS card edge connectors on the backplane in the system bus. Tables 5-1 and 5-2 list the pin assignments and signal mnemonic for connectors P1 and P2, respectively. Tables 5-3 and 5-4 provide a description of the signals listed for the connector pins. With reference to the information in Tables 5-1 through 5-4, the following considerations are important.

1. All odd-numbered pins (1, 3, 5, etc.) are on the component side of the board; even-numbered pins (2, 4, etc.) are on the solder side of the board. 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. A asterisk (*) following a signal mnemonic denotes that the signal is active when in the low state.

Table 5-1. MULTIBUS® Connector P1 Pin Assignments

Component Side			Circuit Side		
Pin	**Mnemonic**	**Function**	**Pin**	**Mnemonic**	**Function**
Power Supplies					
1	GND	Logic Ground	2	GND	Logic Ground
3	+5V	Logic Supply	4	+5V	Logic Supply
5	+5V	Logic Supply	6	+5V	Logic Supply
7	+12V	Logic Supply	8	+12V	Logic Supply
9	N.C.		10	N.C.	
11	GND	Logic Ground	12	GND	Logic Ground

Table 5-1. MULTIBUS® Connector P1 Pin Assignments (Cont'd)

Component Side			Circuit Side		
Pin	Mnemonic	Function	Pin	Mnemonic	Function
Bus Controls					
13	BCLK*	Bus Clock	14	INIT*	Sys Init.
15	BPRN*	Bus Prior In	16	BPRO*	Bus Prior Out
17	BUSY*	Bus Busy	18	BREQ*	Bus Request
19	MRDC*	Memory Read	20	MWTC*	Memory Write
21	IORD*	I/O Read	22	IOWC*	I/O Write
23	XACK*	Transfer Ack	24	N.C.	
Bus Controls and Address					
25	N.C.		26	N.C	
27	BHEN*	High Byte Enable	28	AD10*	Address
29	CBRQ*	Common Bus Req	30	AD11*	Address
31	CCLK*	System Clock	32	AD12*	Address
33	N.C.		34	AD13*	Address
Parallel Interrupt Requests					
35	INT6*	Interrupt 6	36	INT7*	Interrupt 7
37	INT4*	Interrupt 4	38	INT5*	Interrupt 5
39	INT2*	Interrupt 2	40	INT3*	Interrupt 3
41	INT0*	Interrupt 0	42	INT1*	Interrupt 1
Address Bus					
43	ADRE*	Address	44	ADRF*	Address
45	ADRC*	Address	46	ADRD*	Address
47	ADRA*	Address	48	ADRB*	Address
49	ADR8*	Address	50	ADR9*	Address
51	ADR6*	Address	52	ADR7*	Address
53	ADR4*	Address	54	ADR5*	Address
55	ADR2*	Address	56	ADR3*	Address
57	ADR0*	Address	58	ADR1*	Address
Data Bus					
59	DATE*	Data	60	DATF*	Data
61	DATC*	Data	62	DATD*	Data
63	DATA*	Data	64	DATB*	Data
65	DAT8*	Data	66	DAT9*	Data
67	DAT6*	Data	68	DAT7*	Data
69	DAT4*	Data	70	DAT5*	Data
71	DAT2*	Data	72	DAT3*	Data
73	DAT0*	Data	74	DAT1*	Data
Power Supplies					
75	GND	Logic Ground	76	GND	Logic Ground
77	N.C.		78	N.C.	
79	N.C.		80	N.C.	
81	+5V	Logic Supply	82	+5V	Logic Supply
83	+5V	Logic Supply	84	+5V	Logic Supply
85	GND	Logic Ground	86	GND	Logic Ground

Note: N.C. denotes line is not connected.

Because the iSBC 214 controller is limited to 8 of the 60 lines available on the P2 connector, Table 5-2 provides the signal mnemonics only for the lines actively connected to the MULTIBUS interface.

Table 5-2. MULTIBUS® Connector P2 Pin Assignments

Component Side			Circuit Side		
Pin	**Mnemonic**	**Function**	**Pin**	**Mnemonic**	**Function**
Power Supplies					
1 38	GND RST*	Logic Ground Auxiliary	2 18	GND ACLO	Logic Ground P/S Fail Signal
Address Bus					
55 57	ADR16* ADR14*	Add Line 23 Add Line 21	56 58	ADR17* ADR15*	Add Line 24 Add Line 22

Table 5-3. MULTIBUS® Connector P1 I/O Signal Description

Signal Mnemonic	Description
ADR0*–ADR13*	**Address Bits:** Specify part of memory address or I/O Port address to be accessed. (ADR10*–ADR13 are 0 for I/O addresses.)
BHEN*	**Byte High Enable:** Determines, in conjunction with ADR0*, byte bank data to be transferred.
BCLK*	**Bus Clock:** Synchronizes bus contention logic on all bus masters.
CCLK*	**System Clock:** Provides for synchronization of all devices using MULTIBUS interface. Master clock signal. (Not used by iSBC 214 board).
BPRN*	**Bus Priority Input:** Indicates to particular bus master that no higher priority master is requesting the bus. BPRN* is synchronized with BCLK*.
BPRO*	**Bus Priority Output:** 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 the bus. Not used in parallel arbitration scheme.
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.
DAT0*–DATF*	**Data Lines:** Provides for transmission/ reception of 16 parallel bits of data to or from selected memory address or I/O Port. For byte data operations, bits DAT0*–DAT7* make up the even byte; bits DAT8*–DATF* make up the odd byte.
INIT*	**Initialize:** Resets system to a known state.
INT0*–INT7*	**Interrupt Requests:** Provides for transmission of 8 interrupt requests to assigned interrupt handlers.
IORC*	**I/O Read Command:** Indicates that I/O Port address is on MULTIBUS system address lines and port output is to be placed on MULTIBUS system data lines.
IOWC*	**I/O Write Command:** Indicates that I/O port address is on MULTIBUS system address lines and data on MULTIBUS system data lines is to be accepted by the addressed port.

Table 5-3. MULTIBUS® Connector P1 I/O Signal Description (Cont'd)

Signal Mnemonic	Description
MRDC*	**Memory Read Command:** Indicates that memory address is on MULTIBUS system address lines and address contents are to be placed on MULTIBUS system data lines.
MWTC*	**Memory Write Command:** Indicates that memory address is on MULTIBUS system address lines and that data on MULTIBUS system 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 has been placed on or accepted from MULTIBUS data lines).

Table 5-4. MULTIBUS® Connector P2 I/O Signal Description

Signal Mnemonic	Description
ADR14*–ADR17*	**Address Bits:** Specify high-order bits of memory address to be accessed. Used in conjunction with address ADR0*.
RST*	**Reset:** Resets the board.
ACLO*	**AC Power Low:** Signal generated by the power supply goes high when the AC line voltage drops enough to indicate a power failure.

5.3 ST506/412 Winchester Disk Drive Interface

The iSBC 214 controller can interface to one or two ST506/412 Winchester 5.25-inch disk drives. The Winchester physical interface consists of one 34-pin, right angle connector which handles a single control cable and two 20-pin, right angle connectors that mate with the Winchester data cables (one for each drive). The single control cable is connected in daisy chain manner to both drives. Tables 5-5 and 5-6 list the pin assignments for the control signals and data signals, respectively. Pin 1 on each of the right angle headers is oriented on the component side of the board, and designated on the header by a triangle. Table 5-7 provides a description of the signals listed in both Tables 5-5 and 5-6. As with the MULTIBUS interfaces and all peripheral devices, a signal mnemonic followed by an asterisk (*) is active when the line is low.

Table 5-5. Winchester Control Interface Connections

Ground Return	Signal Pin	Signal Name
1	2	Reduced Write Current* or Head Select 23* as selected by jumper E79-E81
3	4	Head Select 22*
5	6	Write Gate*
7	8	Seek Complete*
9	10	Track 0*
11	12	Write Fault*
13	14	Head Select 20*
15	16	Not used
17	18	Head Select 21*
19	20	Index*
21	22	Ready*
23	24	Step*

Table 5-5. Winchester Control Interface Connections (Cont'd)

Ground Return	Signal Pin	Signal Name
25	26	Drive Select 1*
27	28	Drive Select 2*
29	30	Drive Select 3*
31	32	Drive Select 4*
33	34	Direction*

Table 5-6. Winchester Data Interface Connections

Ground Return	Signal Pin	Signal Name
2	1	Drive Selected*
4	3	Reserved
6	5	Reserved
8	7	Reserved
—	9, 10	Reserved
12	11	Ground
—	13	+ MFM Write Data
—	14	− MFM Write Data
16	15	Ground
—	17	+ MFM Read Data
—	18	− MFM Read Data
20	19	Ground

Note: Differential Data Drivers follow the EIA RS-422 Electrical Interface Specifications.

Table 5-7. Winchester Control/Data Signal Description

Control Signal	Description
Write Gate*	This signal, when active, allows write data to be written on the disk. When the signal is inactive, data can be transferred from the drive and step pulses are enabled to step the R/W head positioner.
Seek Complete*	This line goes true when the R/W heads have settled on the final track at the end of a seek. Reading or writing should not be attempted when the signal is false (high).
Track 0*	This signal indicates a true state only when the R/W heads are positioned at track 0 (the outermost data track).
Drive Select 1* Drive Select 2* Drive Select 3* Drive Select 4*	This signal connects the designated drive interface to the control lines.
Write Fault*	This signal indicates an existing condition at the drive which may causes improper writing on the disk. When active, this signal inhibits further write and stepping operations until the condition is corrected.
Head Select 2^0* Head Select 2^1* Head Select 2^2* Head Select 2^3*	These signal lines select each individual read/write head in a binary coded sequence. Head Select 2^0 is the least significant line. Heads are numbered 0 through 3. When all four Head Select lines are false (high), head 0 is selected.
Recovery Mode*	This signal is not used by the iSBC 214 controller.
Index*	The drive provides this interface signal once per disk revolution to indicate beginning of the track. The transition from the normal high state to the low state only is valid and indicates index.

Table 5-7. Winchester Control/Data Signal Description (Cont'd)

Control Signal	Description
Ready*	This interface signal when low combined with a low Seek Complete indicates the drive is ready to read, write, or seek and that all I/O signals are valid. When this line is high, all writing and seeking is inhibited.
Step*	This signal causes the R/W head to move with the direction of motion defined by the Direction In signal.
Direction*	This signal defines motion direction of the R/W head when the Step signal is pulsed. A logical one defines direction as "out" causing the R/W heads to move away from the disk center. A logical zero (low) defines direction as "in" causing the heads to move toward the center of the disk.
Reduced Write Current*	This signal reduces the recording write current on the drive's innermost track.
Drive Selected*	This signal provides status at the J2 connector to inform the host controller of the selection status of each drive.
+ MFM Write Data − MFM Write Data	This is a differential pair that defines the transitions to be written on the track. The transition of the + MFM Write Data going more positive than the − MFM Write Data signal causes flux reversal on the track, provided the Write Gate signal is active (low). When the host system is in a read mode, the signal is driven to an inactive state (+ MFM Write Data becoming more negative than the − MFM Write Data signal).
+ MFM Read Data − MFM Read Data	The data recovered by reading a prerecorded track is transmitted to the host system via this differential pair of lines. The transition of the + MFM Read Data line going more positive than the − MFM Read Data line represents a flux reversal on the designated head's track.

5.4 SA450/460 5.25-inch Flexible Disk Interface

The iSBC 214 controller can interface a maximum of four 5.25-inch flexible disk drives. To select 5.25-inch flexible disk drives, the user may jumper certain stake pins on the board. Refer to Chapter 3 for more detailed information. The flexible disk physical interface consists of a single 34-pin right-angle connector which mates to the drive's data/control cable. Note that flexible diskettes formatted using an iSBC 215G controller with an iSBX 218A MULTIMODULE are compatible with the iSBC 214 controller as long as the data encoding technique is the same (FM or MFM). Pin 1 on the right-angle connector is oriented on the component side of the board, and designated on the header by a triangle. Table 5-8 lists the pin assignments and signals on the SA450/460 flexible disk interface. Table 5-9 provides a description of the signals listed in Table 5-8. Signal mnemonics followed by an asterisk (*) are active when they are in the low state.

Table 5-8. Flexible Disk Interface Connections

Ground Return	Signal Pin	Signal Name
1	2	Reduced Write Current* (not used)
3	4	Head Load*
5	6	Drive Select 4*
7	8	Index*
9	10	Drive Select 1*

Table 5-8. Flexible Disk Interface Connections (Cont'd)

Ground Return	Signal Pin	Signal Name
11	12	Drive Select 2*
13	14	Drive Select 3*
15	16	Motor On*
17	18	Direction*
19	20	Step*
21	22	Write Data*
23	24	Write Gate*
25	26	Track 0*
27	28	Write Protect*
29	30	Read Data*
31	32	Side Select*
33	34	Ready*

Table 5-9. Flexible Disk Interface Signal Descriptions

Signal Name	Description
Reduced Write Current*	This signal, when active, goes to the drive and indicates that the track number is greater than 43 (decimal). It also indicates to the drive that the head write current needs to be reduced. (Typically not used by drives.)
Head Load*	This signal, when active, enables the activity LED on the selected drive, on all drives in a daisy chain, or separately in a radial configuration.
Drive Select 1* Drive Select 2* Drive Select 3* Drive Select 4*	An active signal on one of these lines allows the user system to select which drive on the interface is to be used.
Index*	The drive provides this signal to indicate each time an index or sector hole is sensed at the Index/Sector photo detector. The signal is normally high and makes the transition to the low state each time a hole is sensed.
Motor On*	When active, this signal turns the drive motor on which enables diskette read/write operations.
Direction*	This interface line defines the motion the read/write heads will take when the Step line is pulsed. A logical one (high) defines "out" and causes the heads to move away from the cylinder center. A logical zero (low) defines "in" and causes the heads to move toward the center of the cylinder.
Step*	This interface control signal causes the read/write heads to move with the direction of the motion defined by the Direction signal. The access motion is initiated on the low-to-high transition.
Write Data*	This interface signal provides the data to be written on the diskette, The high-to-low state transition causes current through the read/write heads to reverse, thus writing a data bit. This line is enabled by active Write Gate signal.
Write Gate*	Active, this signal enables the Write Data signal to write data to the disk. The inactive state of this signal enables the read data logic and head stepper logic.
Track 0*	This interface signal, when active, indicates when the drive's read/write heads are positioned at track zero (the outermost track).
Write Protect*	The drive provides this signal to the controller to indicate when a write-protected diskette is installed in the drive. The signal is active when the diskette is write-protected.
Read Data*	This interface line provides the "raw data" (clock and data together) as detected by the drive's electronics.

Table 5-9. Flexible Disk Interface Signal Descriptions (Cont'd)

Signal Name	Description
Side Select*	This signal defines which side of the two-sided diskette is to be written to or read from. A logical one selects the side 0 head.
Ready*	The drive provides this signal to indicate that the drive is ready to proceed with read/write operations. (Some drives do not support this line.)

5.5 QIC-2 Tape Drive Interface

The iSBC 214 board can interface up to four QIC-2 standard streaming tape drives. All standard commands in the QIC-2 interface are supported by the iSBC 214 board. The QIC-2 physical interface on the board consists of a single 50-pin right-angle connector. The QIC-2 control signal cable mates with this header; additional drives are daisy-chained off the control cable. Table 5-10 lists the pin assignments for the QIC-2 interface to the iSBC 214 controller. Table 5-11 provides a brief description of the signals listed in Table 5-10. Pin 1 on the right-angle header is oriented on the component side of the board, and designated on the header by a triangle. Note, that signals followed by an asterisk (*) in both tables are active when they are in the low state.

Table 5-10. QIC-2 Control Signal Interface Connections

Ground Return	Signal Pin	Signal Name
1	2	Spare
3	4	Spare
5	6	Spare
7	8	Spare
9	10	Host Bus Odd Parity
11	12	Host Data Bus 7* MSB
13	14	Host Data Bus 6* MSB
15	16	Host Data Bus 5* MSB
17	18	Host Data Bus 4* MSB
19	20	Host Data Bus 3* MSB
21	22	Host Data Bus 2* MSB
23	24	Host Data Bus 1* MSB
25	26	Host Data Bus 0* LSB
27	28	On Line*
29	30	Request*
31	32	Reset*
33	34	Transfer*
35	36	Acknowledge*
37	38	Ready*
39	40	Exception*
41	42	Direction*
43	44	Spare
45	46	Spare
47	48	Spare
49	50	Spare

Table 5-11. QIC-2 Tape Interface Signal Descriptions

Signal Name	Description
Host Bus Odd Parity	This signal is generated from the controller to check odd parity on the data transmission lines.
Host Data Bus 7*	This is the most significant bit of the 8-bit host, bi-directional data bus.
Host Data Bus 0*	This is the least significant bit of the 8-bit host, bi-directional data bus.
On Line*	This signal is generated from the controller prior to transferring a read or write command. The signal is deactivated to terminate the read or write command.
Request*	This signal comes from the iSBC 214 controller to indicate that command data has been placed on the data bus in Command Mode or that status has been taken from the data bus in Status Input Mode. This signal is generated only when the Ready or Exception signals are asserted by the tape device.
Reset*	This controller generated signal causes the tape device initialization to be performed. The default drive selection is drive unit 0. The Exception signal must also be asserted.
Transfer*	This controller-generated signal indicates that data has been placed on the data bus in Write Mode or that data has been taken from the data bus in Read Mode.
Acknowledge*	The tape drive provides this signal to indicate that data has been taken from the data bus in Write Mode or that data has been placed on the data bus in Read Mode.
Ready*	The tape drive provides this signal to indicate one of the following: 1. data has been taken from the data bus in Command Transfer Mode 2. data has been placed on the data bus in Status Input Mode 3. a BOT, Cartridge Intialization or Erase command is completed following issuance 4. the device is ready to receive the next block or ready to receive a Write or WFM (Write File Mark) Command from the host system in Write Mode 5. a Write File Mark Command is completed in Write File Mark Mode 6. the device is ready to transmit the next block of data to the host or ready to receive a Read or Read File Mark Command from the host in Read Mode 7. device is ready to receive a new command
Exception*	The tape drive generates this signal to indicate that an exception condition exists in the device. The host must issue a Status Command and perform a Status Input to determine the cause.
Direction*	This tape drive-generated signal when high, causes host data bus drivers to assert their data bus levels and device data bus drivers to assume high impedence states. When the signal is low, it causes host data bus drivers to assume high impedence states and device data bus drivers to assert their data bus levels.

CHAPTER 6
SERVICE INFORMATION

6.1 Introduction

This chapter contains specific service and repair instructions, and the schematic diagram for the iSBC 214 controller board.

6.2 Service Diagrams

Table 6-1 locates and identifies the factory default configuration of jumpered stake pins. Figure 6-1 shows the location of the jumpers on the iSBC 214 board. Figure 6-2 is the schematic diagram of the iSBC 214 controller board. Available configuration options for all operations are described in Chapter 3.

NOTE

It is a good idea to consult the schematics shipped with the iSBC 214 controller board for any recent changes.

6.3 Service and Repair Assistance

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

Prior to calling the Product Service Marketing Administration you should have the following information available:

1. Date that you received the product.
2. Complete part number of the product (including 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:

Northwest (Wash, Ore, No. Calif, Colo)	602-869-4951
Southwest (So. Calif, Az, New Mexico)	602-869-4023
Midwest	602-869-4392
New England	602-869-4045
East	602-869-4950
International	602-869-4862
TWX Number:	910-951-1330

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 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) to pack 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, N.J.). Enclose the product in a heavy-duty corrugated shipping carton and mark the carton FRAGILE to ensure careful handling. Ship only to the address specified by the Service Center personnel.

Table 6-1. iSBC® 214 Default Jumper Configuration

Default Jumper	Function	Board Location
E2 - E3	Tape Parity Disabled	1
E4 - E5	Hard Reset Disabled	2
E12 - E13	Timeout Enabled	4
E15 - E16	Reserved	5
E17 - E18	Reserved	5
E20 - E21	Reserved	16
E22 - E23	Reserved	17
E31 - E33	FLP Drive READY always ready	3
E36 - E37	Reserved	3
E38 - E39	Reserved	3
E57 - E58	Wake-Up Port Address 100H	11
E75 - E76	Megabyte Page 0 Selected	14
E77 - E78	16-Bit Data Bus Selected	10
E79 - E81	1-16 Wini Heads Selected	8
E82 - E83	NMI Enabled	8
E86 - E87	27128 EPROMs Supported	6
E89 - E90	Reserved	9
E91 - E92	Reserved	9
E93 - E94	Reserved	9
E98 - E99	Relinquish bus on CBRQ*	15
E101 - E102	Drive CBRQ*	15
E103 - E104	Reserved	15
E105 - E106	Reserved	12
E111 - E112	Reserved	12
E113 - E114	FLP drive polling mode 0	12
E115 - E116	Reserved	12
E117 - E118	FLP drive polling mode	12
E119 - E120	16-bit I/O address	12
E121 - E122	Wini buffered seeks Enabled	12
E126 - E129	MULTIBUS Interrupt 5	13
E135 - E138	Reserved	7
E137 - E140	Reserved	7
E142 - E143	ACLO Disabled	7
E144 - E145	Reserved	3

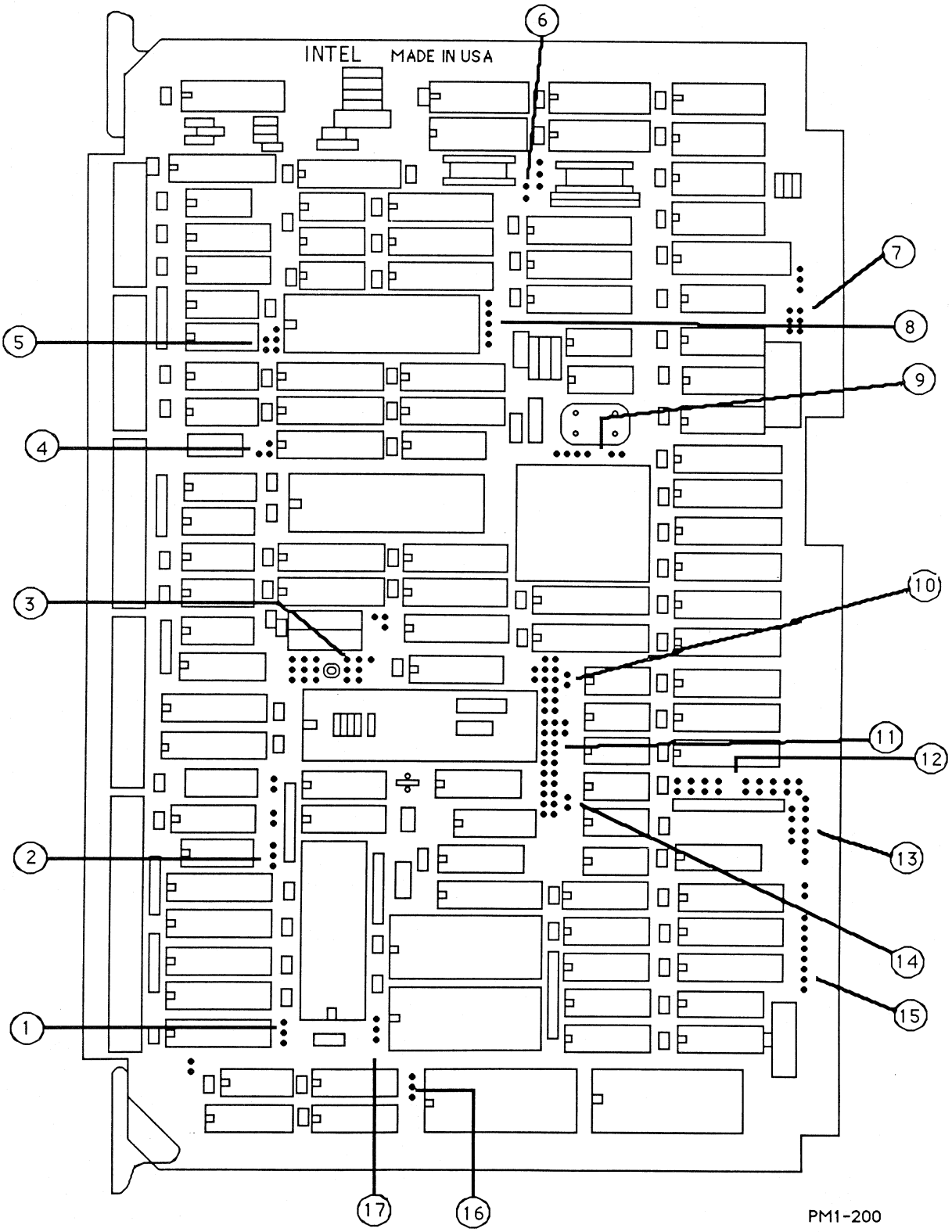


Figure 6-1. iSBC® 214 Controller Board Jumper Locations

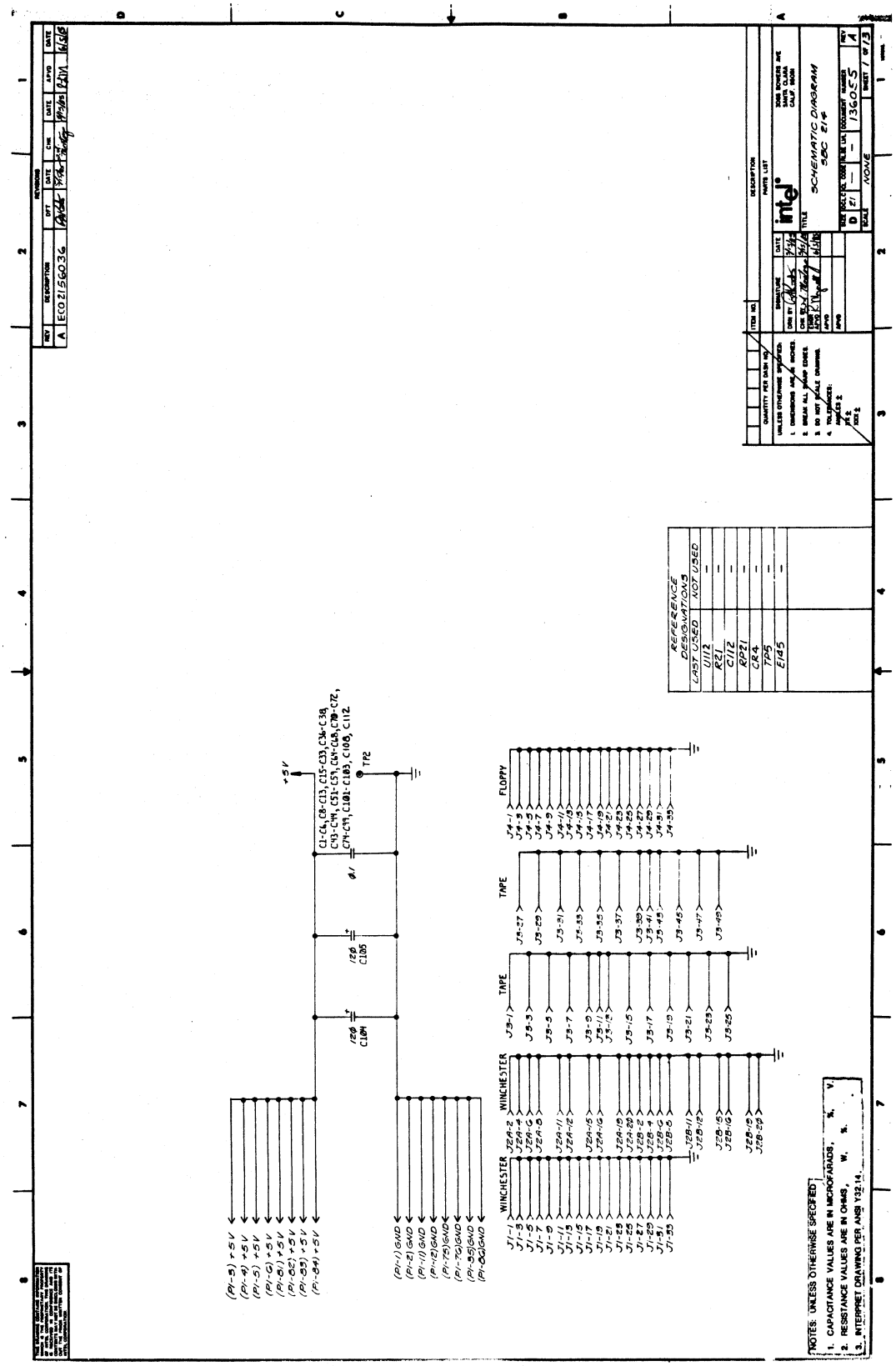


Figure 6-2. iSBC® 214 Controller Board Schematic Diagrams (sheet 1 of 13)

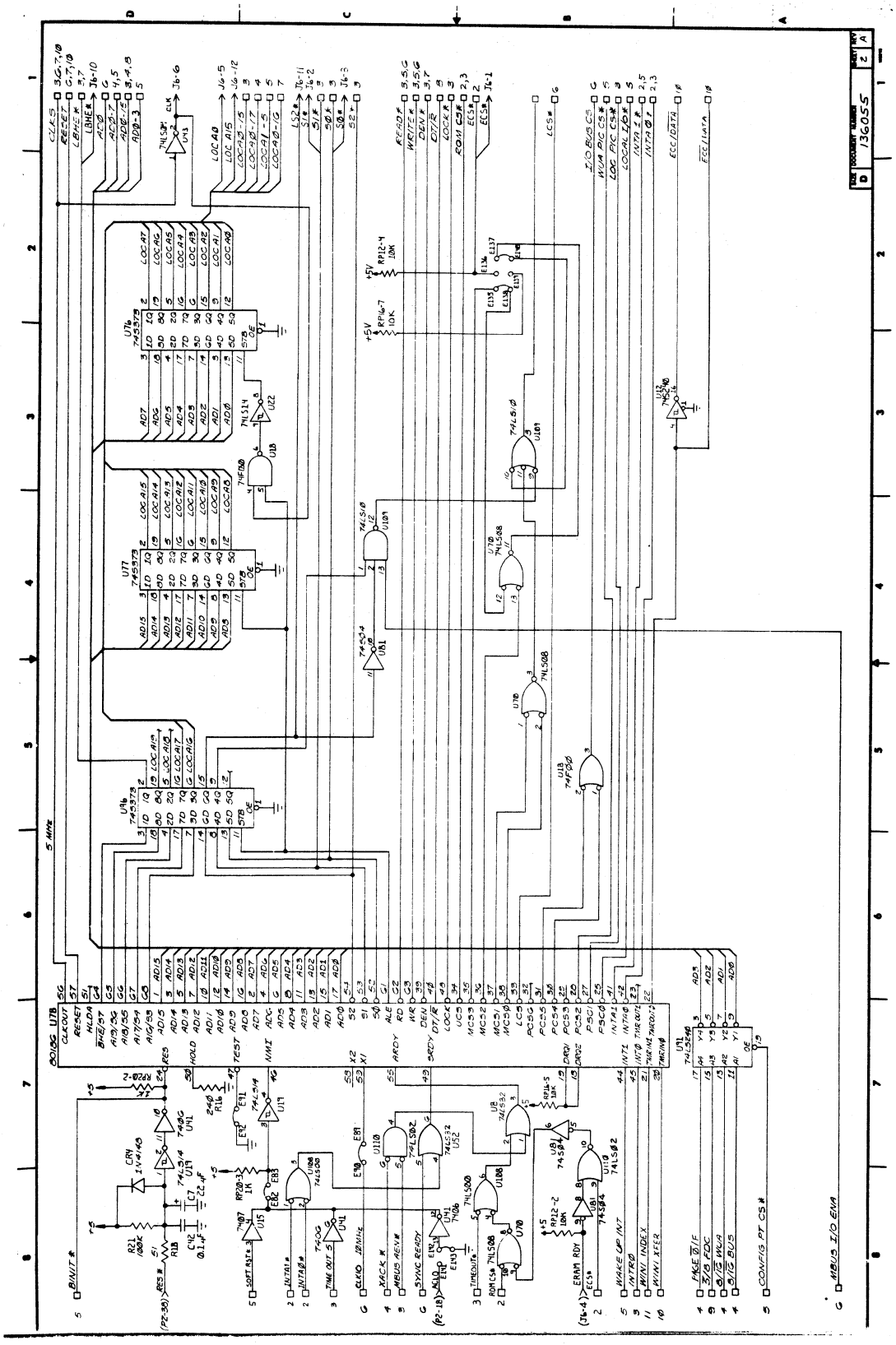


Figure 6-2. iSBC® 214 Controller Board Schematic Diagrams (sheet 2 of 13)

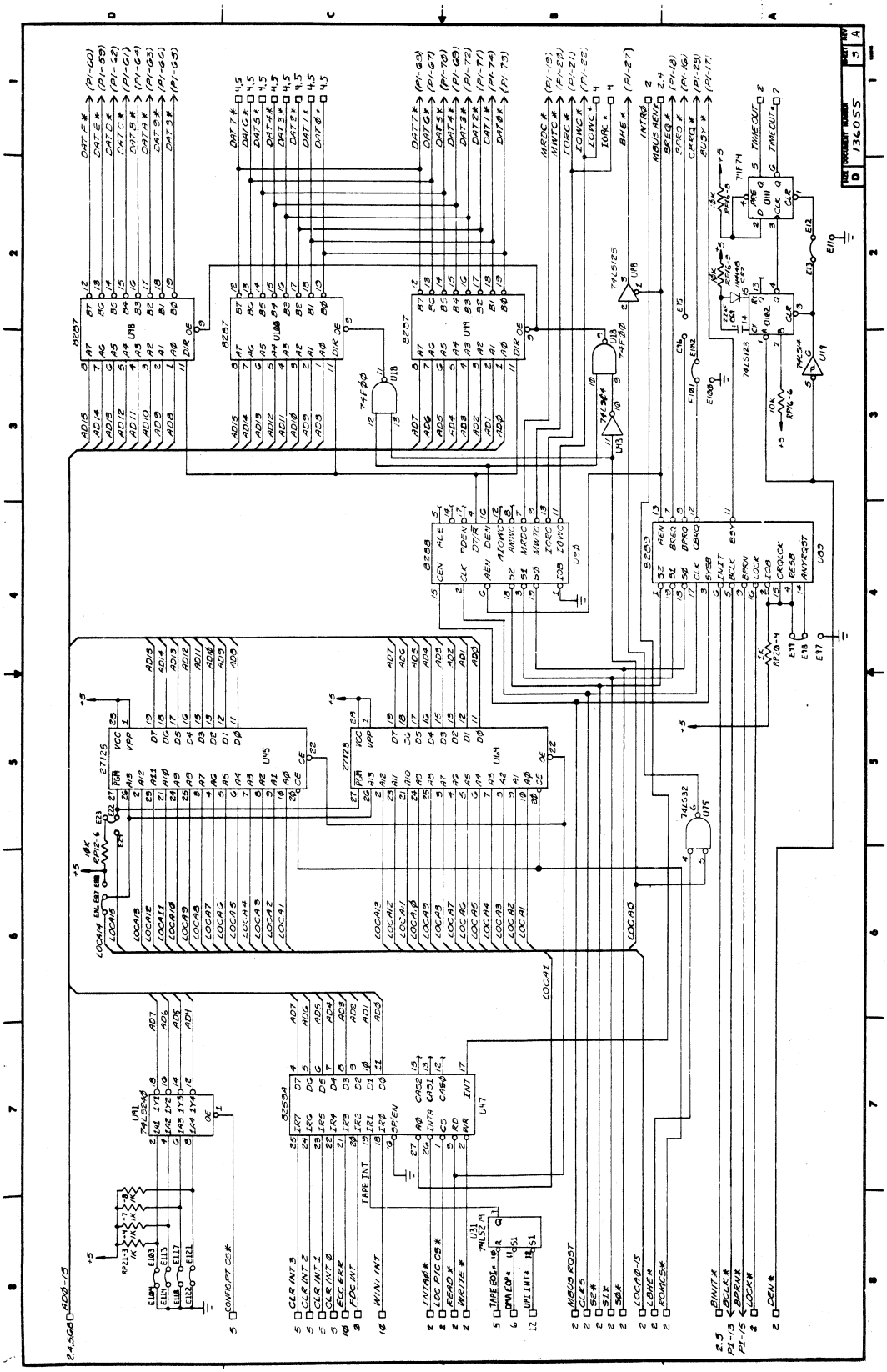


Figure 6-2. iSBC® 214 Controller Board Schematic Diagrams (sheet 3 of 13)

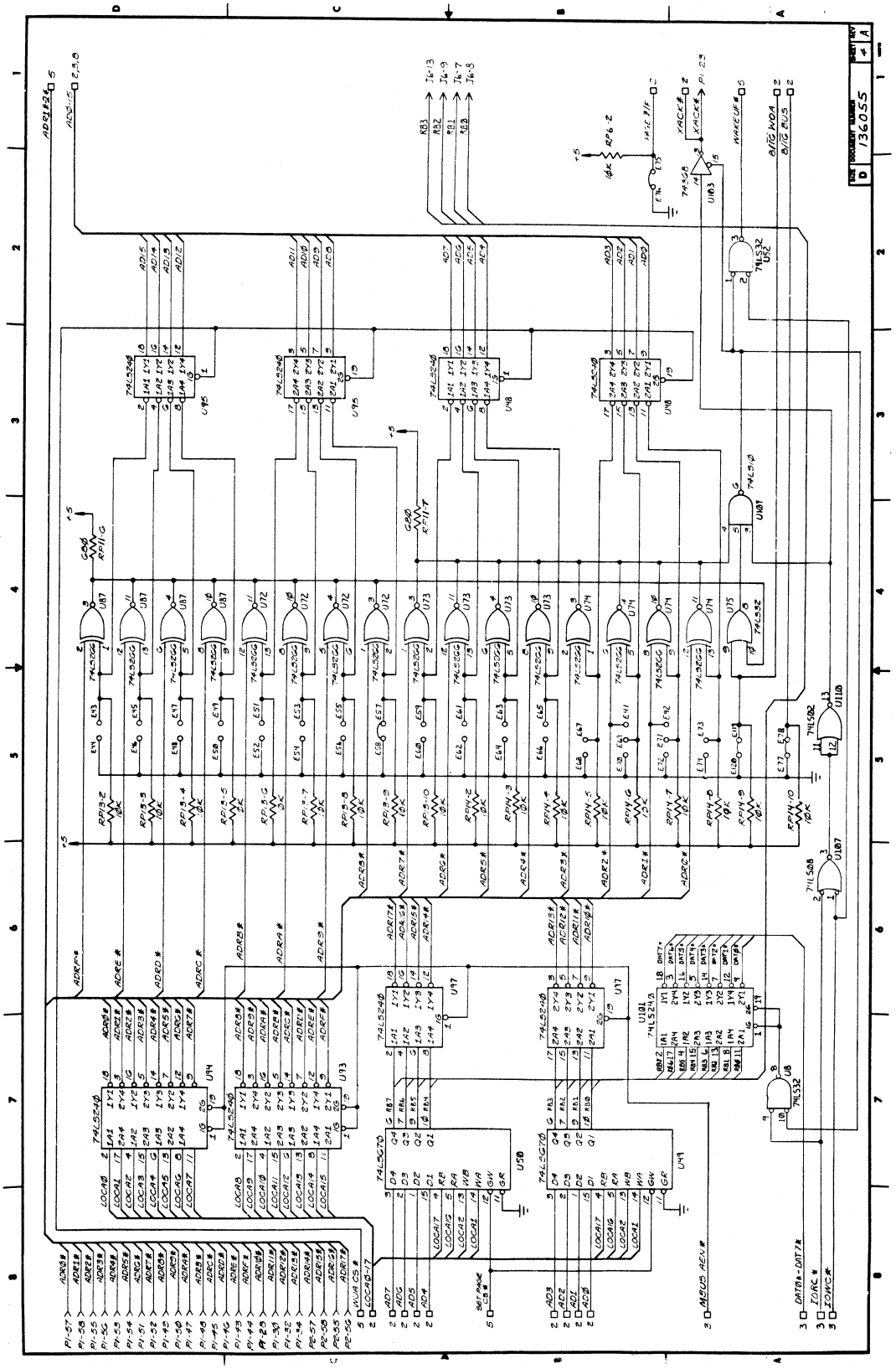


Figure 6-2. iSBC® 214 Controller Board Schematic Diagrams (sheet 4 of 13)

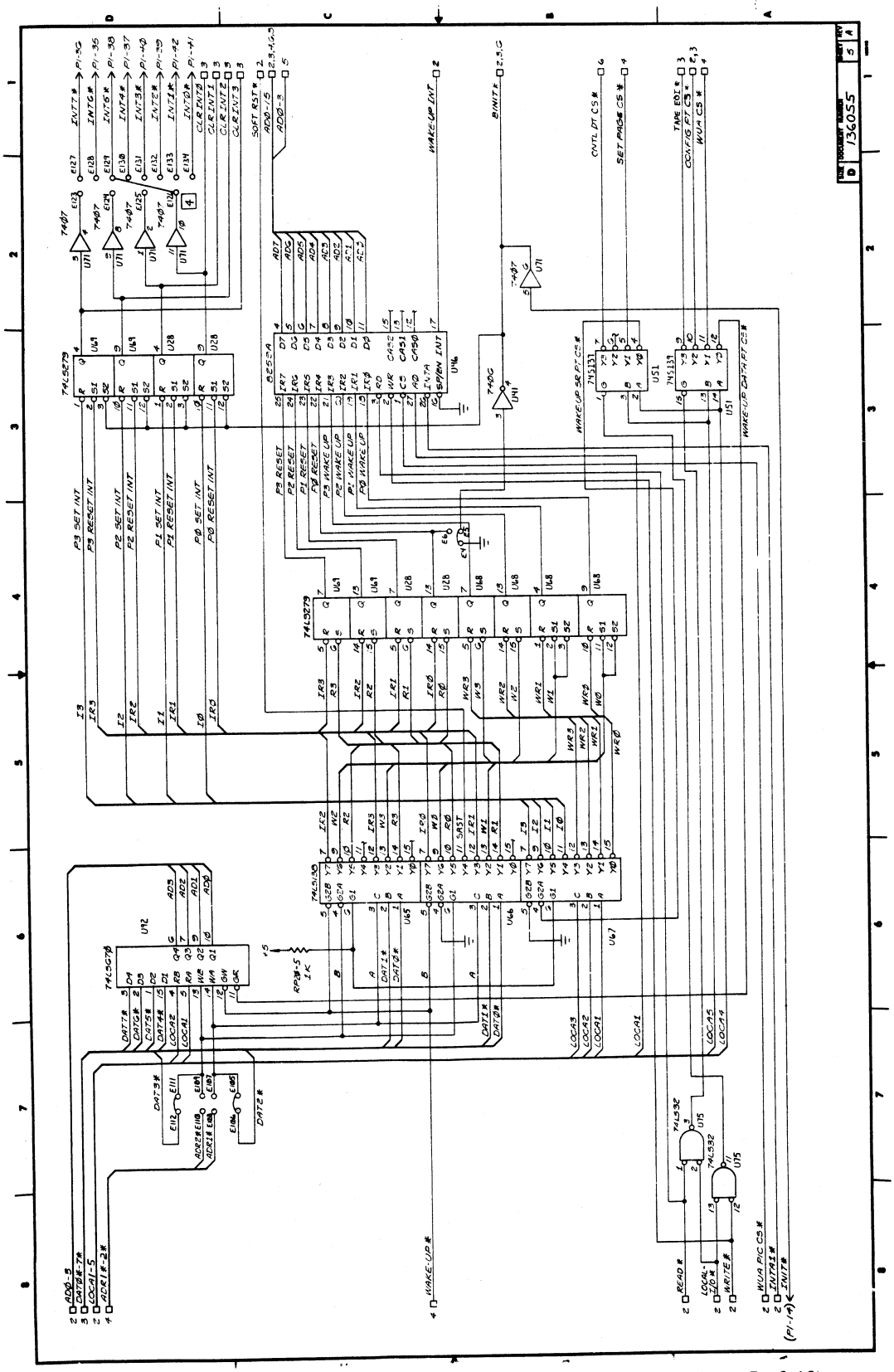


Figure 6-2. iSBC® 214 Controller Board Schematic Diagrams (sheet 5 of 13)

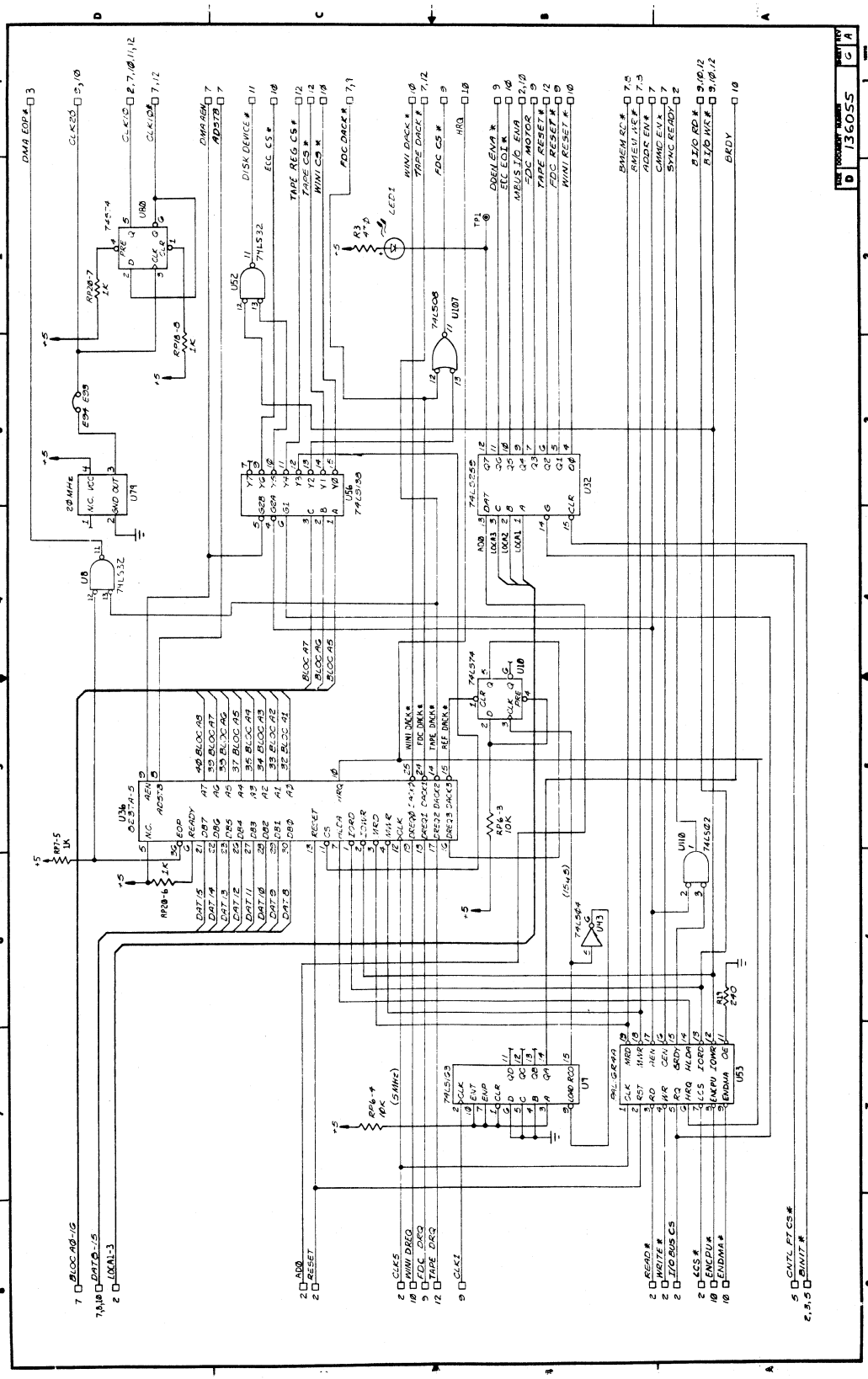


Figure 6-2. iSBC® 214 Controller Board Schematic Diagrams (sheet 6 of 13)

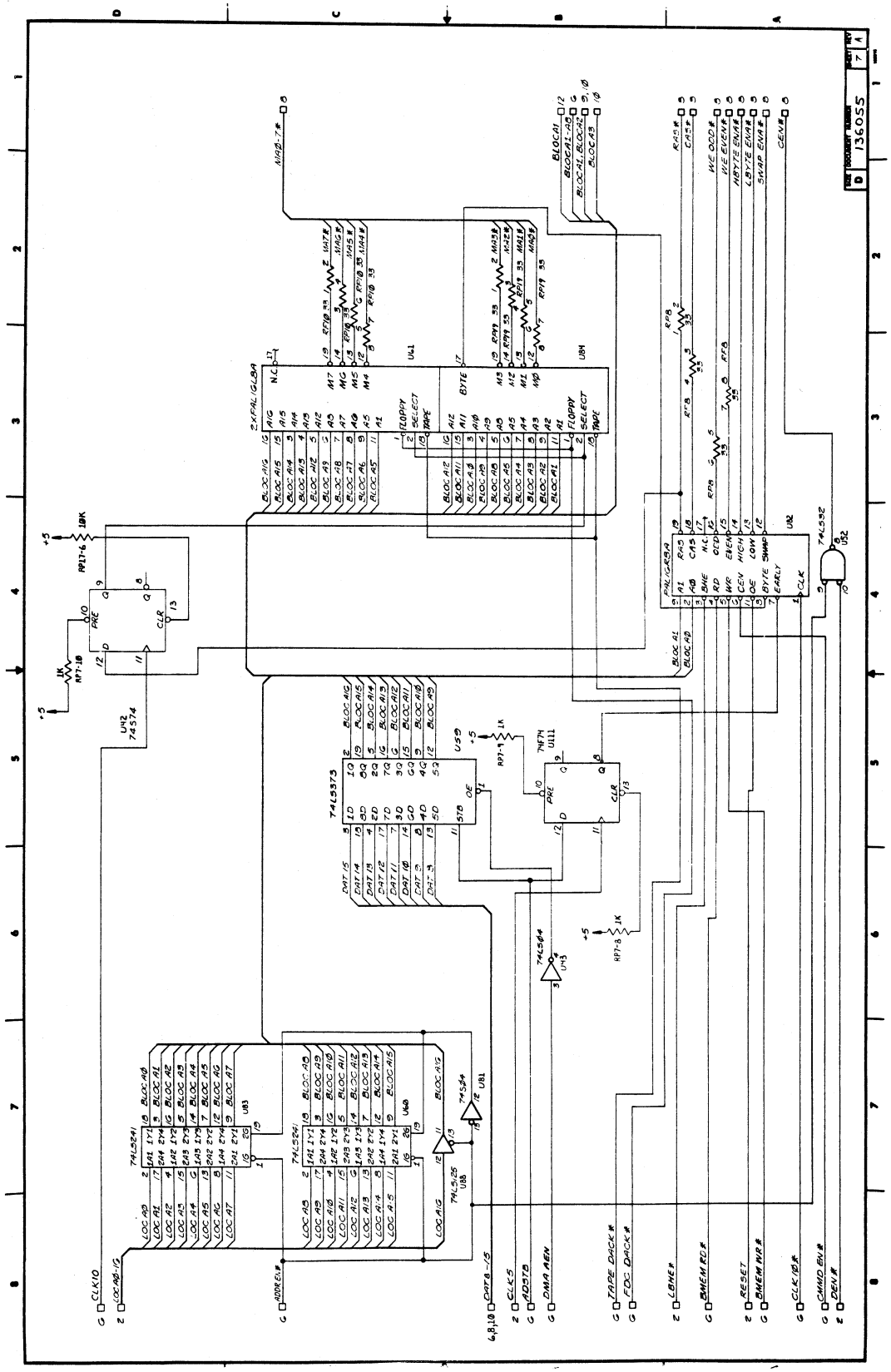


Figure 6-2. iSBC® 214 Controller Board Schematic Diagrams (sheet 7 of 13)

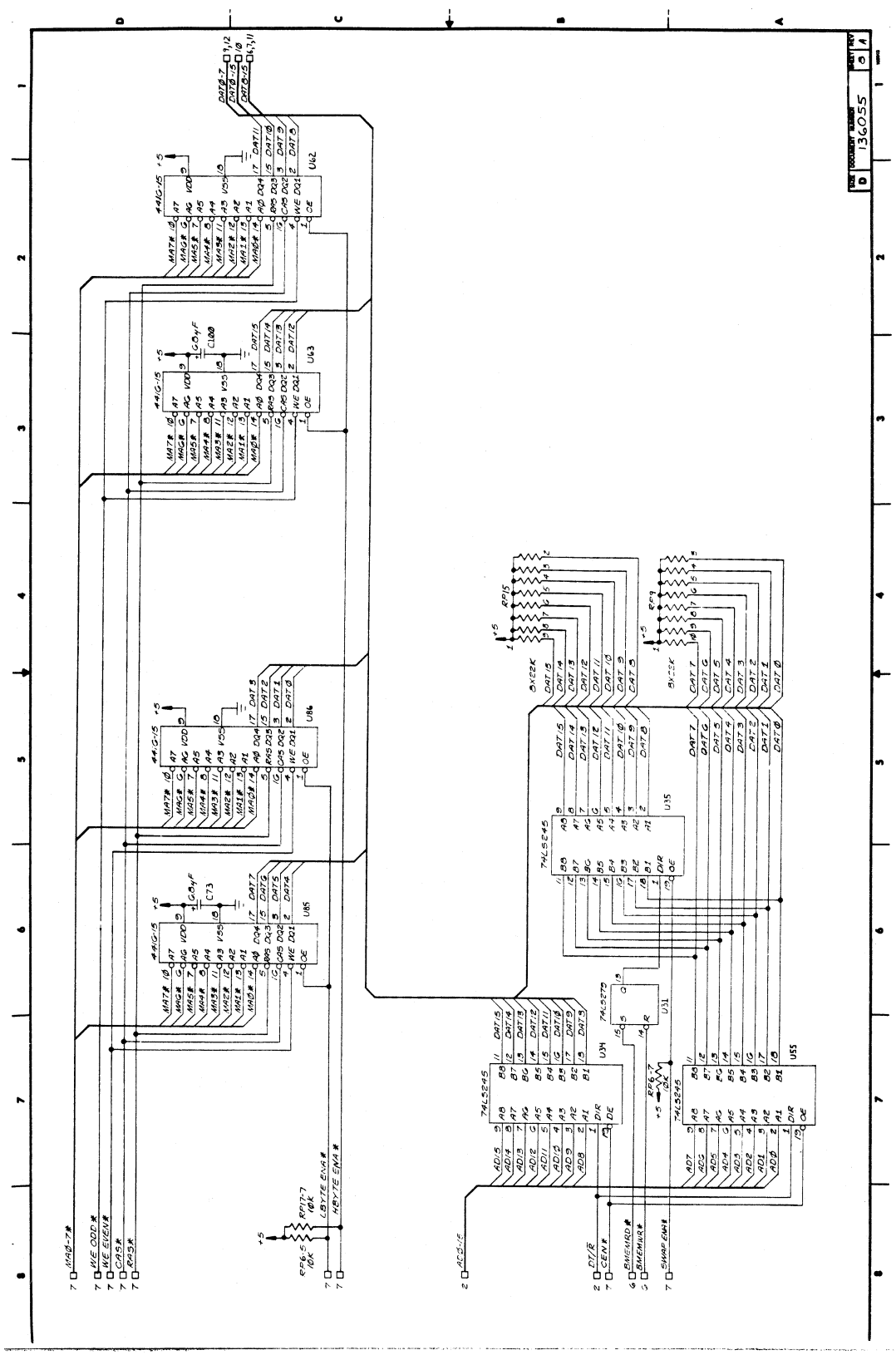


Figure 6-2. iSBC® 214 Controller Board Schematic Diagrams (sheet 8 of 13)

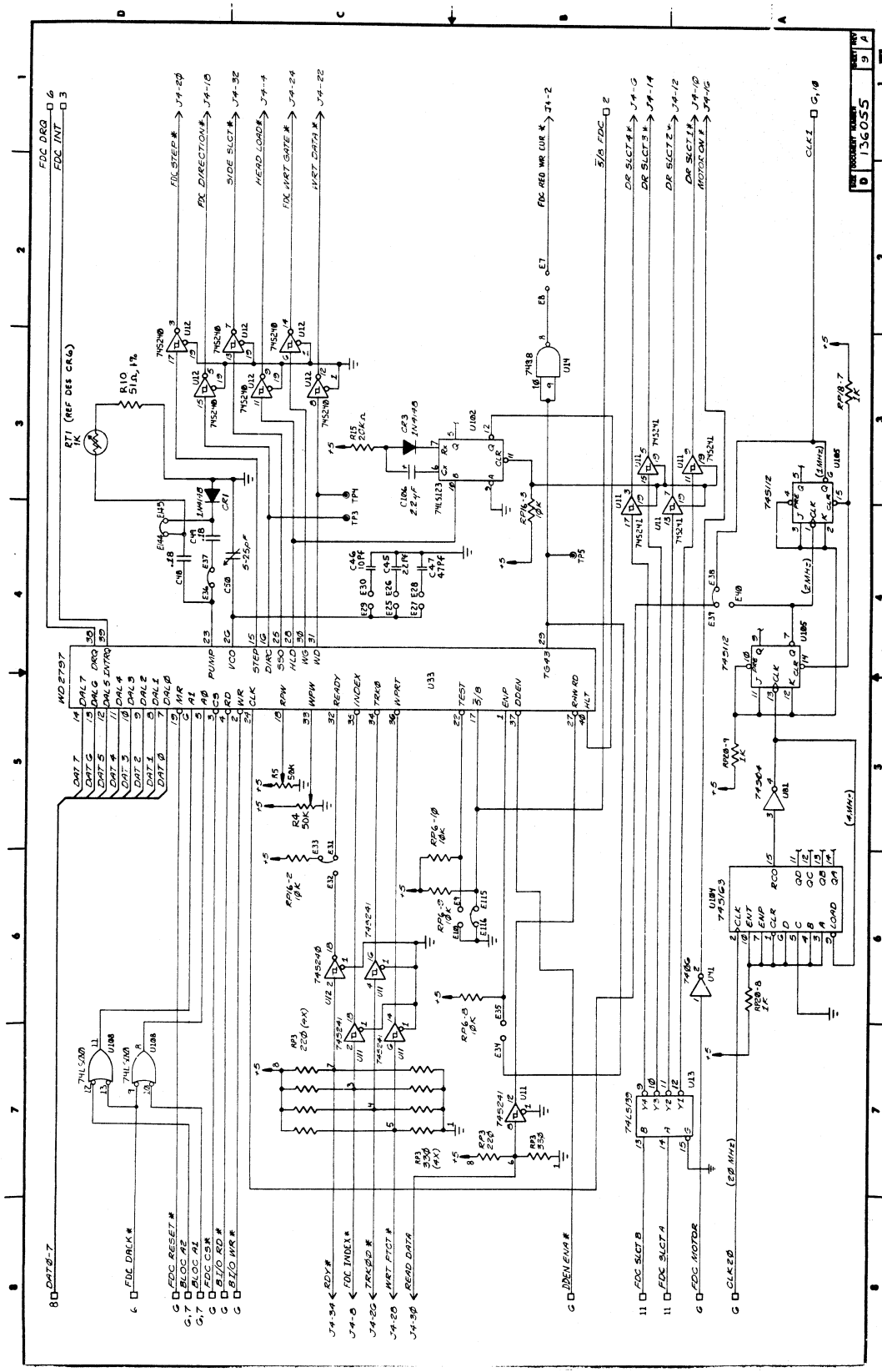


Figure 6-2. iSBC® 214 Controller Board Schematic Diagrams (sheet 9 of 13)

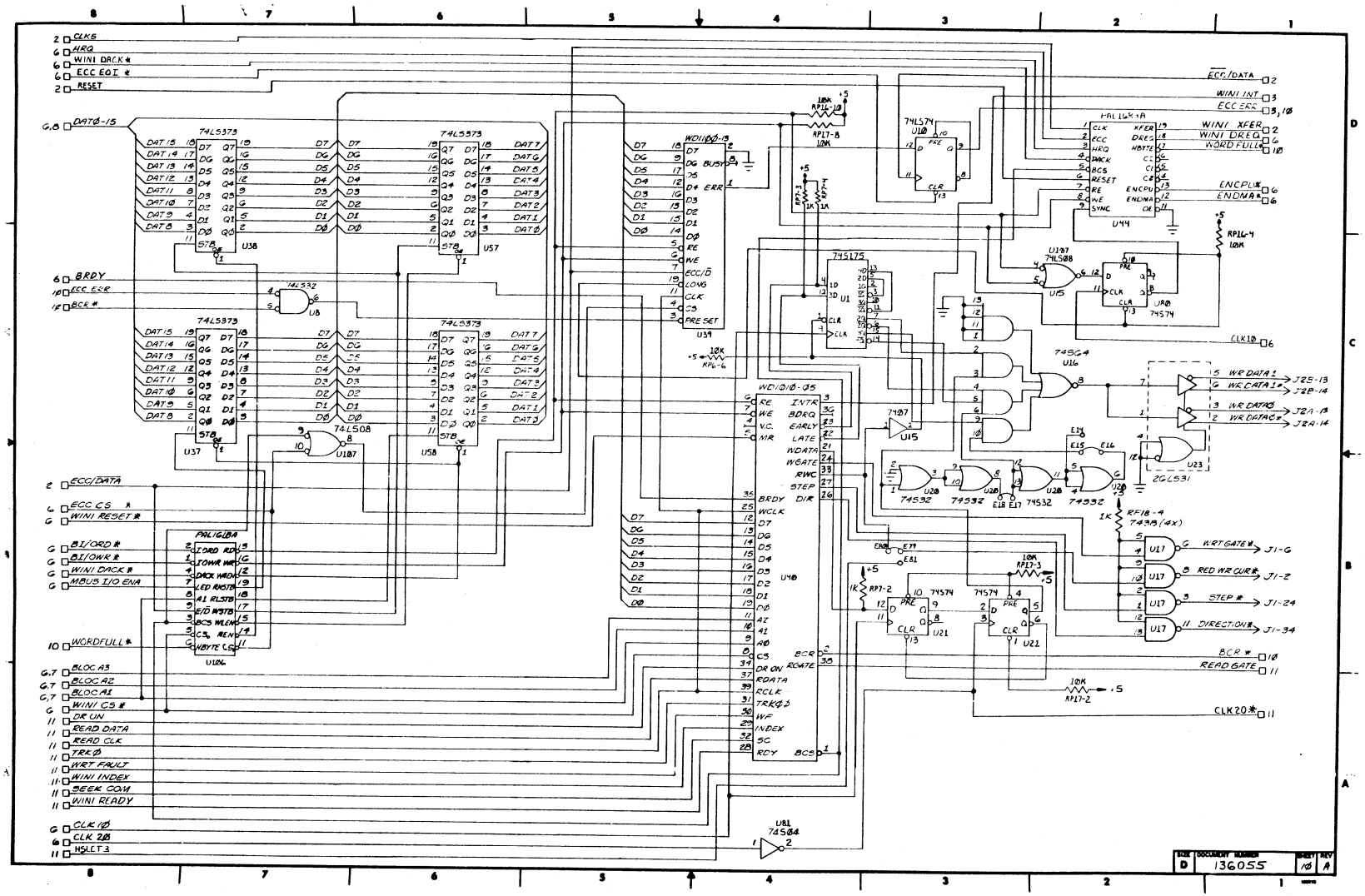


Figure 6-2. iSBC® 214 Controller Board Schematic Diagrams (sheet 10 of 13)

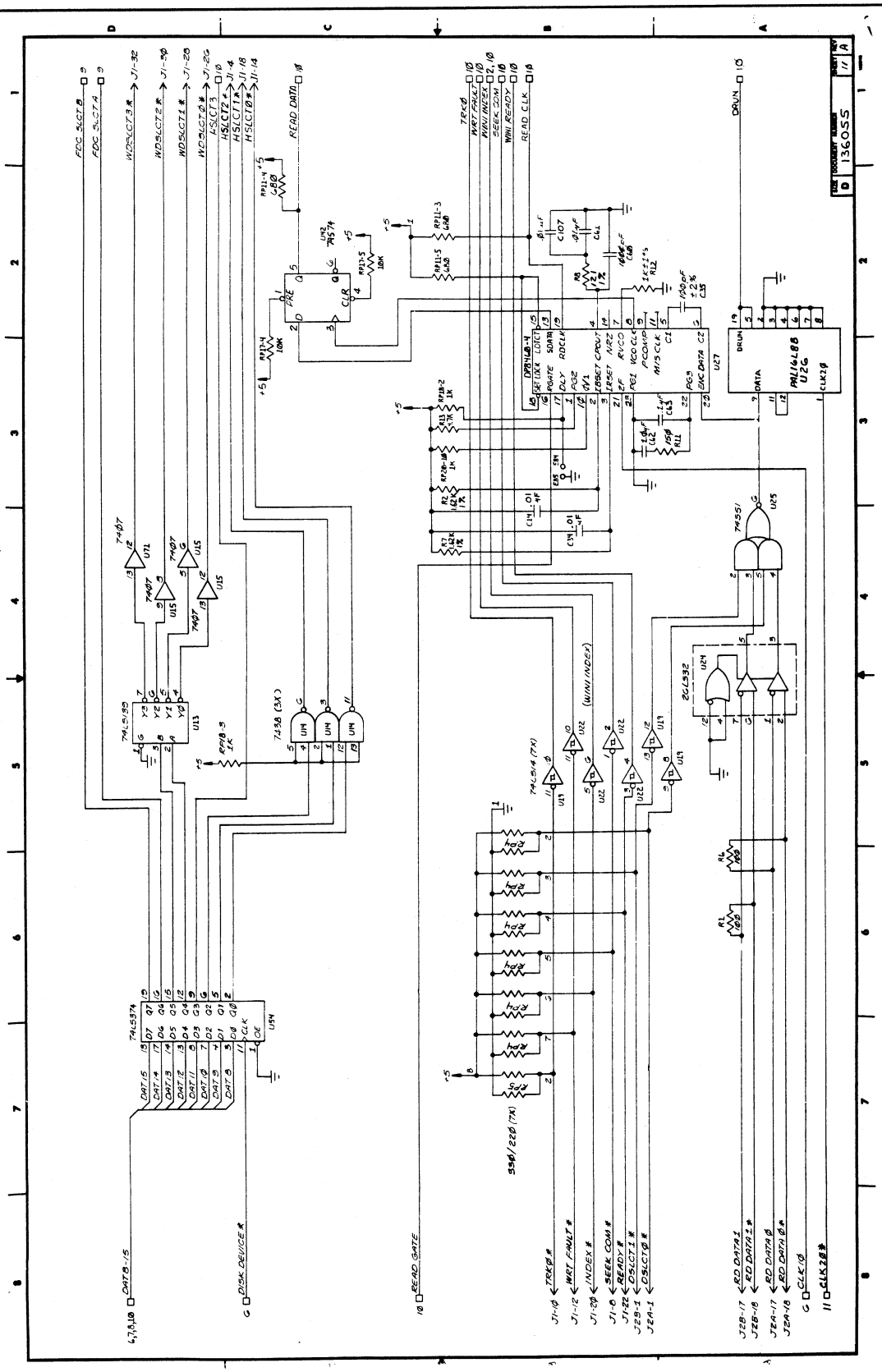


Figure 6-2. iSBC® 214 Controller Board Schematic Diagrams (sheet 11 of 13)

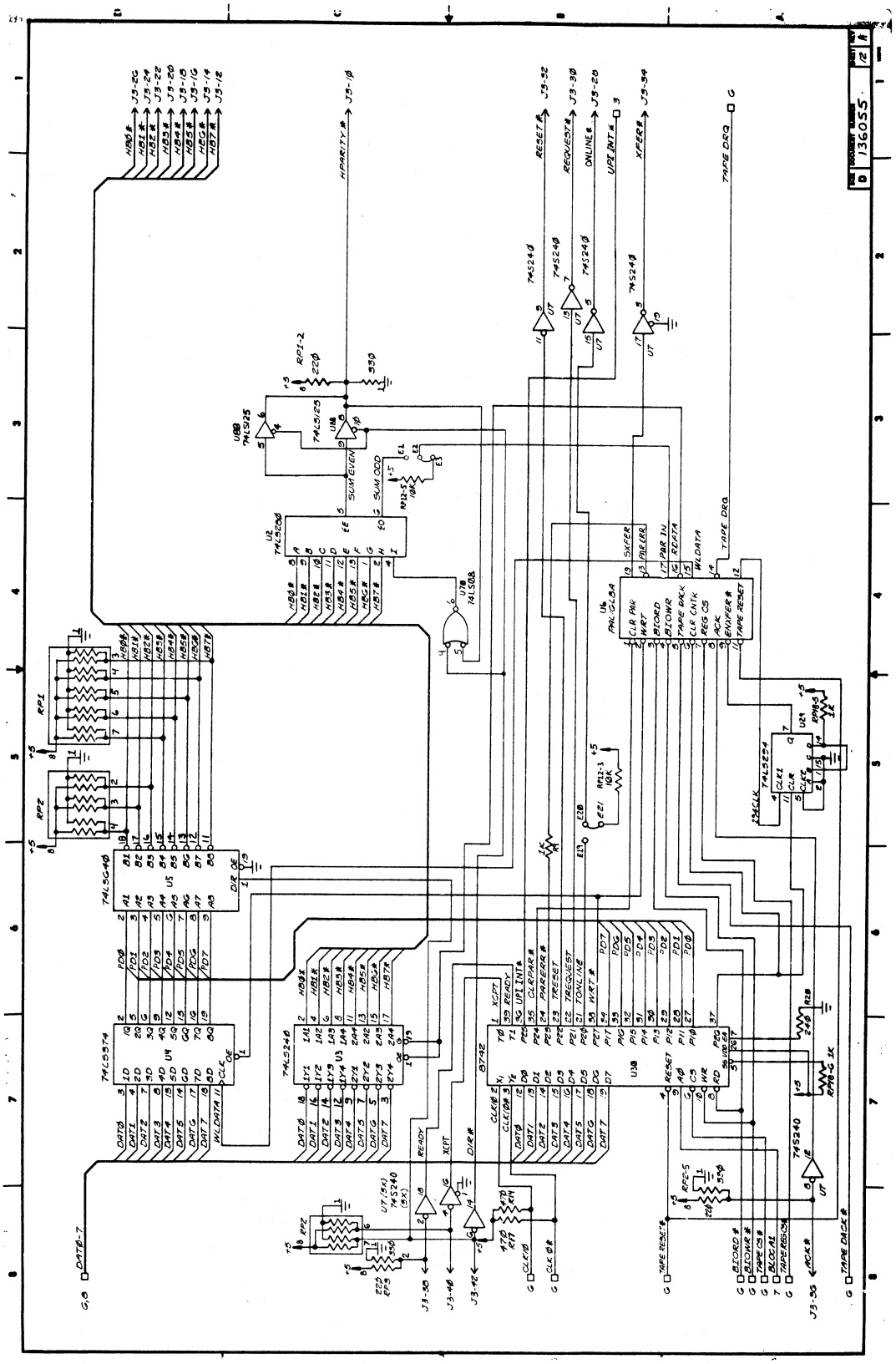
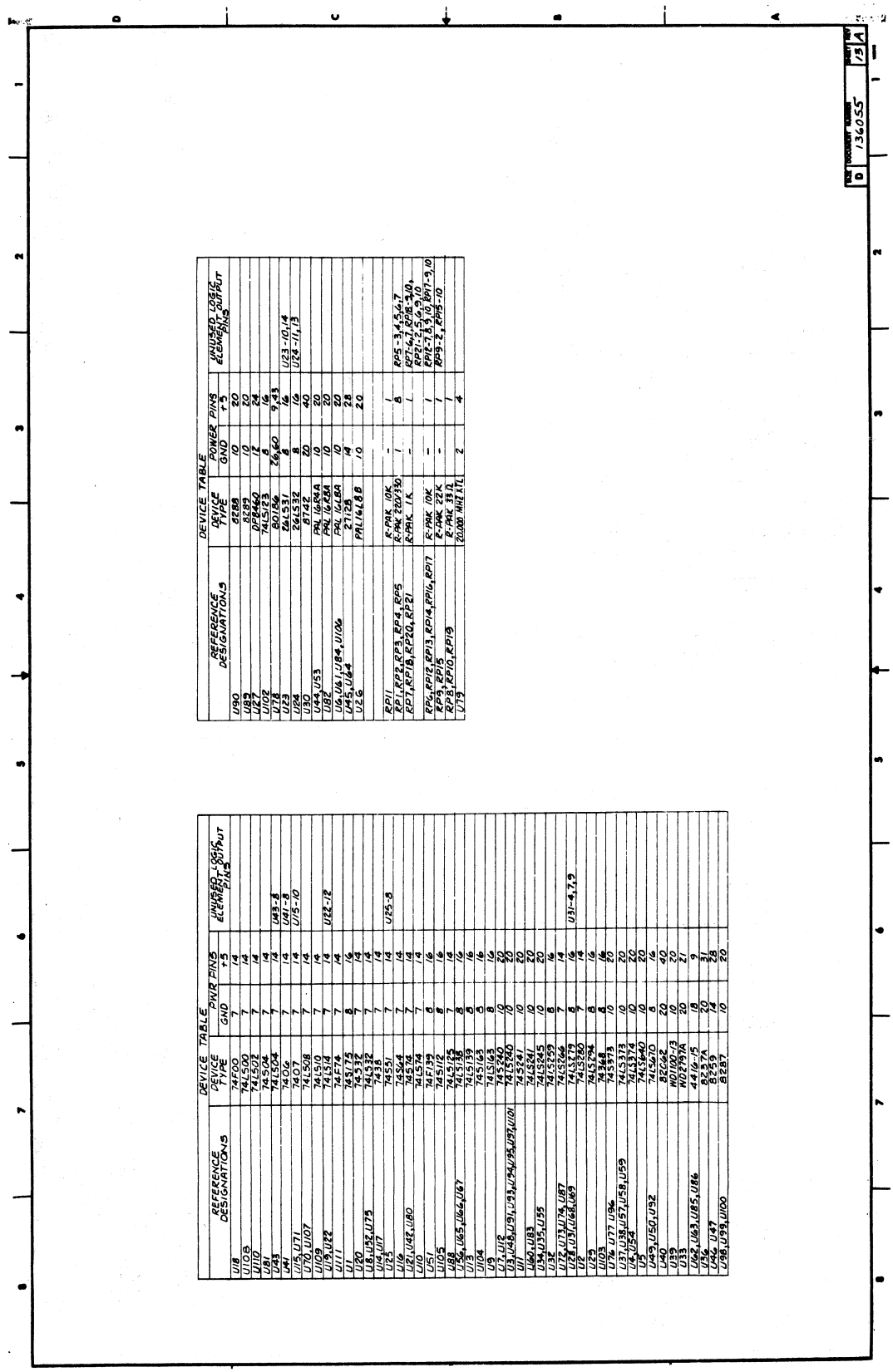


Figure 6-2. iSBC® 214 Controller Board Schematic Diagrams (sheet 12 of 13)



DEVICE TABLE

REFERENCE DESIGNATIONS	DEVICE TYPE	POWER PINS GND	POWER PINS +5	UNUSED LOGIC ELEMENT OUTPUT PINS
U90	8288	10	20	
U89	8289	10	20	
U27	DP8460	12	24	
U102	74LS123	8	16	
U78	80130	26,40	9,33	
U23	74LS32	7	14	U23-10,14
U24	74LS32	7	14	U24-11,13
U30	8742	20	40	
U44,U53	PAL 16R4A	10	20	
U82	PAL 16R8A	10	20	
U6,U41,U84,U106	PAL 16L8A	10	20	
U45,U64	27128	14	28	
U26	PAL16L8B	10	20	
RP11	R-PAK 10K	-	1	
RP1,RP2,RP3,RP4,RP5	R-PAK 220/330	1	8	RP5-3,4,5,6,7
RP7,RP18,RP20,RP21	R-PAK 1K	-	1	RP7-6,7,8,9,10 RP8-3,10, RP21-2,5,6,9,10, RP18-7,8,9,10 RP17-9,10
RP6,RP12,RP13,RP14,RP16,RP17	R-PAK 10K	-	1	
RP19,RP15	R-PAK 22K	-	1	RP19-2, RP15-10
RP8,RP10,RP15	R-PAK 33K	-	1	
U119	20.000 MHZ TTL	2	4	

DEVICE TABLE

REFERENCE DESIGNATIONS	DEVICE TYPE	PWR PINS GND	PWR PINS +5	UNUSED LOGIC ELEMENT OUTPUT PINS
U18	74F00	7	14	
U108	74LS00	7	14	
U110	74LS02	7	14	
U81	74S04	7	14	
U43	74LS04	7	14	U43-8
U41	7406	7	14	U41-8
U72,U71	7407	7	14	U15-10
U70,U107	74LS08	7	14	
U109	74LS10	7	14	
U15,U22	74LS14	7	14	U22-12
U111	74F74	7	14	
U1	74S175	8	16	
U20	74S32	7	14	
U8,U58,U75	74LS32	7	14	
U14,U17	7438	7	14	
U25	74S51	7	14	U25-8
U16	74S64	7	14	
U21,U42,U80	74S74	7	14	
U10	74LS74	7	14	
U51	74F139	8	16	
U105	74S112	8	16	
U88	74LS123	8	16	
U3,U65,U66,U67	74LS138	8	16	
U13	74LS139	8	16	
U104	74S163	8	16	
U9	74LS163	8	16	
U7,U12	74S240	10	20	
U3,U48,U91,U93,U94,U95,U97,U101	74LS240	10	20	
U11	74S241	10	20	
U62,U83	74LS241	10	20	
U34,U35,U55	74LS245	10	20	
U32	74LS259	8	16	
U12,U13,U14,U87	74LS266	7	14	
U28,U31,U68,U69	74LS279	8	16	U31-4,7,9
U2	74LS280	7	14	
U29	74LS294	8	16	
U103	74S373	8	16	
U76,U77,U96	74LS373	8	16	
U4,U38,U57,U58,U59	74LS374	10	20	
U54	74LS540	10	20	
U5	74LS640	10	20	
U49,U50,U92	74LS670	8	16	
U40	82C62	20	40	
U39	W01100-13	10	20	
U33	W02797A	20	21	
U82,U83,U85,U86	4416-15	18	9	
U46	8237A	10	11	
U46,U47	8259	14	28	
U98,U99,U100	8287	10	20	

Figure 6-2. iSBC® 214 Controller Board Schematic Diagrams (sheet 13 of 13)

REQUEST FOR READER'S COMMENTS

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

Please restrict your comments to the usability, accuracy, readibility, organization, and completeness of this publication. If you have any comments on the product that this publication describes, please contact your Intel representative. If you wish to order publications, contact the Intel Literature Department (see 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 for improvement.

3. Is this the right type of publication for your needs? Is it at the right level? What other types of 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 _____

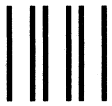
ADDRESS _____

CITY _____ STATE _____ ZIP CODE _____
(COUNTRY)

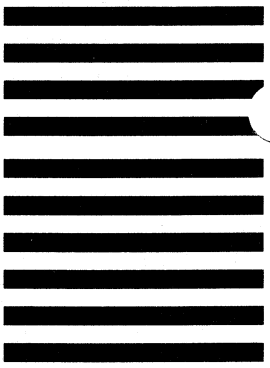
Please check here if you require a written reply. ☐

WE'D LIKE YOUR COMMENTS . . .

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



NO POSTAGE
NECESSARY
IF MAILED
IN U.S.A.



BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1040 SANTA CLARA, CA

POSTAGE WILL BE PAID BY ADDRESSEE

Intel Corporation
Attn: Technical Publications M/S DV2-82
2402 West Beardsley Road
Phoenix, Arizona 85027