POINT 4™

MARK 3
4

PERIPHERALS INTERFACE MANUAL

Not Up to Date

POINT 4
DATA CORPORATION

POINT 4 DATA CORPORATION
2569 McCabe Way / Irvine, California 92714

POINT 4™
MARK 3
PERIPHERALS INTERFACE MANUAL

NOTICE

Every attempt has been made to make this reference manual complete, accurate and up-to-date. However, all information herein is subject to change due to updates. All inquiries concerning this manual should be directed to POINT 4 Data Corporation.

POINT 4 Data Corporation
2569 McCabe Way
Irvine, CA 92714
(714) 754-4114

REVISION RECORD

PUBLICATION NUMBER: HM-081-0027

Revision	Description	Date
A	Initial Release	08/06/82

LIST OF EFFECTIVE PAGES

Changes, additions, and deletions to information in this manual are indicated by vertical bars in the margins or by a dot near the page number if the entire page is affected. A vertical bar by the page number indicates pagination rather than content has changed.

Page	Rev	Page	Rev	Page	Rev
Cover	-				
Title	-				
ii thru xii	A				
1-1 thru 1-20	A				
2-1 thru 2-32	A				
3-1 thru 3-49	A				
4-1 thru 4-37	A				
Appendix Title	-				
A-1	A				
B-1 thru B-8	A				
Comment Sheet	A				
Mailer	-				
Back Cover	-				

PREFACE

This manual describes the interface functions of the POINT 4 MARK 3 Peripherals Interface Board (PIB). The PIB is a peripheral controller that interfaces the MARK 3 CPU to three groups of peripherals: asynchronous communications lines, disc drive units, and tape drive units.

The introduction provides information regarding the characteristics and internal architecture of the POINT 4 MARK 3 PIB as well as input/output interface handling.

Separate sections are devoted to the multiplexer, disc drive interface, and tape drive interface. The appendices provide a table of ASCII codes and tape handling flowcharts.

A separate manual describes the MARK 3 Computer System and provides step-by-step instructions for installation, upgrade and operating procedures.

Related manuals include:

Title	Pub. Number
POINT 4 MARK 3 Computer System Manual	HM-081-0019
POINT 4 MARK 3 Diagnostics Manual	

CONTENTS

APPENDICES

FIGURES

TABLES

Section 1
INTRODUCTION

1.1 GENERAL DESCRIPTION

The Peripheral Interface Board (PIB) is a peripheral controller that interfaces the MARK 3 CPU to three groups of peripherals: asynchronous communications lines, disc drive units, and tape drive units. The PIB interfaces to the MARK 3 CPU via the backplane, providing high-speed DMA transfer between peripherals and the CPU. Control for all PIB functions is provided by the CPU as needed via I/O instructions. The PIB board itself has no on-board intelligence.

PIB features include:

- Separate Software Programmed I/O Control
- DMA Data Transfer
- Simultaneous Operation of Three Groups of Peripherals
- Ease of Software Interface

1.1.1 SYSTEM DESCRIPTION

The PIB contains three main control modules:

- Asynchronous Communications Control Module
- CMD/SMD/LMD Disc Drive Control Module
- Tape Drive Control Module

Utilizing these modules, the MARK 3 system can control up to three types of peripherals simultaneously.

Figure 1-1 shows the basic system block diagram for the PIB. Communication is via the backplane DMA facility, utilizing I/O control blocks (IOCBs) for access control. The PIB includes device control and instruction decode logic. This allows the CPU to access each control module individually.

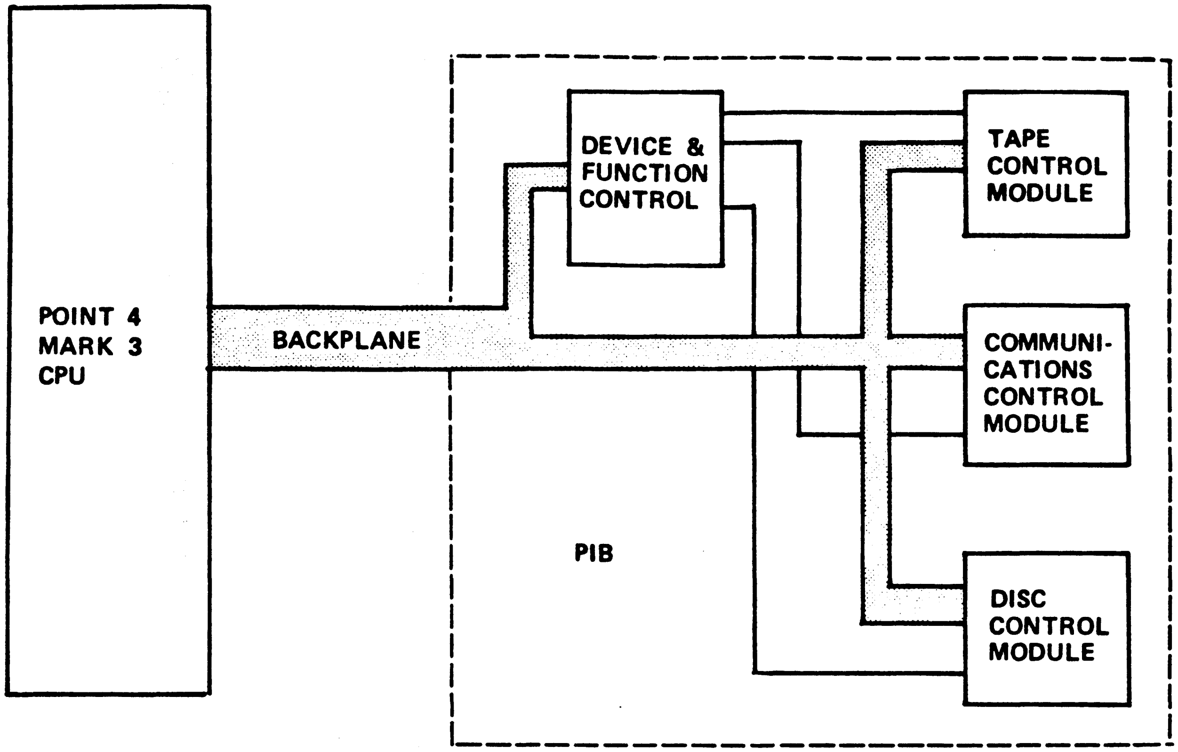


Figure 1-1. Peripheral Interface Board System Block Diagram

1.1.2 EQUIPMENT CHARACTERISTICS

The following subsections give performance and equipment characteristics for the MARK 3 Peripheral Interface Board.

1.1.2.1 Performance Characteristics

COMMUNICATIONS CONTROL

Transmission Type:
Asynchronous

Line Types:
Half or Full Duplex

Line Interface:
RS-232C

Number of Ports:
Four
(Seven with Port Expansion Board option)

Modes of Operation:
Programmed I/O
Port 0 - Device Code 10/11
Port 1 - Device Code 12/13
Port 2 - Device Code 14/15
Port 3 - Device Code 16/17
Port 4 - Device Code 20/21
Port 5 - Device Code 22/23
Port 6 - Device Code 24/25
(Ports 4 thru 6 are Optional Expansion Ports)
DMA Operation - Device Code 77

Line Printer:
May be connected to any port

Baud Rates (Hardware Strappable):
110, 150, 300, 600, 1200, 2400, 4800, 9600

DISC DRIVE INTERFACE

Drives per Controller:
Two

Drive Type:
SMD/CMD/LMD drives supporting the sector mark interface signal

DMA Transfer Rate:
1.25 megabytes per second

Sector Size:
Header - 4 words
Data - 256 words

TAPE DRIVE INTERFACE

Drives per Controller:
Four

Drive Type:
Archive Streamer Tape Units

DMA Transfer Rate:
90 kilobytes/second

Tape Speed:
90 inches/second

Tape Capacity:
20 megabytes

INPUT/OUTPUT INSTRUCTIONS

Input - DIA
Output - DOA

Device Codes:
10-17 - MUX Ports 0-3
20-25 - Expansion Ports 4-6
50-55 - Disc 0&1
60-62 - Tape 0-4

DMA (Device Code 77):
DOBS/DIBS - MUX
DOBC/DIBC - Disc
DOBP/DIBP - Tape

CPU Functions (Device Code 77):
SKPBN - Skip if Interrupts Enabled
SKPDN - Skip if Power-Fail Detected
SKPBZ - Skip if Interrupts Disabled
SKPDZ - Skip if no Power-Fail Detected

1.1.2.2 Equipment Specifications

Power Supply:

Nominal	Rated
Voltage	Current
+5V dc	18 amps
-5V dc	500 milliamps
-12V dc	100 milliamps
+12V dc	1.5 amps

Environmental Requirements:

Temperature:
0 to +50°C (32 to 122° F) operating
-20 to +100°C nonoperating

Relative Humidity: 5 to 90% noncondensing

Dimensions: 14.5 in. x 12 in. (36.8 cm x 30.5 cm)

1.1.3 PIB INTERNAL ARCHITECTURE

Figure 1-2 shows the internal architecture of the Peripheral Interface Board (PIB). There are four areas of control performed on the PIB:

- General Control Functions
- Multiplexer Control Functions
- Disc Control Functions
- Tape Control Functions

General Control Functions include data reception/transmission, command decoding, device decoding, and status recording.

Multiplexer Control Functions include baud rate selection, data transfer synchronization, port selection, data format manipulation, and the real-time clock.

Disc Control Functions include disc selection, bus control for DMA transfer, word count, CRC generation, and tag line control.

Tape Control Functions include function decode, strobe generation, command and status, and data transfer control.

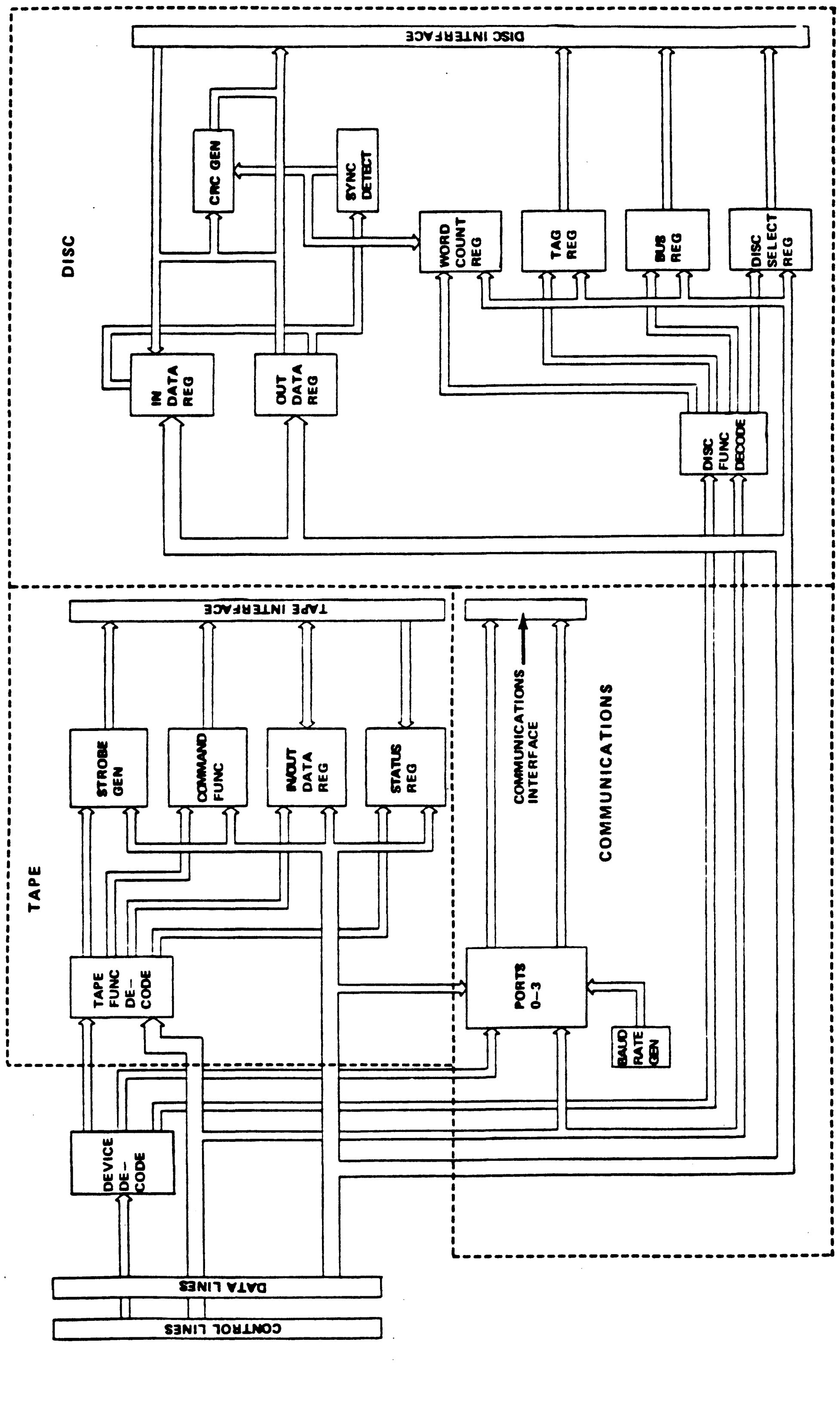


Figure 1-2. PIB Internal Architecture

1.2 INPUT/OUTPUT INTERFACE HANDLING

This section provides information about basic operating principles and programming methods for input/output devices. Two types of I/O devices are compatible with the POINT 4 MARK 3 Computer:

- those transferring data via I/O programmed instructions only
- those using Direct Memory Access (DMA) for input/output transfers

The following subsections outline interrupt handling and priority scheme, programmed transfer handling and DMA channel transfer handling.

1.2.1 PROGRAM INTERRUPT AND PRIORITY SCHEME

Many I/O devices require service within a short time of request; however, service is needed infrequently (relative to the processor speed) and only a small amount of time is required to service them. Failure to service within the specified time (which varies among devices) causes operation below the device's maximum speed and can result in loss of information.

The use of interrupts in the current program sequence facilitates concurrent operation of the main program and servicing of a number of peripheral devices. The program interrupt scheme allows an I/O device to gain control of the processor. When a device raises an interrupt-request flag, the processor suspends normal program execution and starts a device service routine. When the routine is completed, the processor returns to the interrupted program.

1.2.1.1 Interrupt Sequence

When a device needs service, it sets its interrupt-request flag. The processor begins servicing interrupts if all four of the following conditions exist:

- The processor has just completed an instruction fetch or a data channel transfer
- At least one device has an interrupt-request flag set
- Interrupts are enabled (i.e., ION is set)
- No device is waiting for a DMA transfer

The processor responds to the interrupt request by storing the value of the program counter in memory location 0 and jumping to the instruction addressed by memory location 1. Location 1 must contain the address of the interrupt handling routine. Interrupts are disabled at the start of the interrupt service cycle and must be re-enabled by the software at the end of the interrupt service.

The POINT 4 MARK 3 has only one priority level for interrupts. The mask-out (MSKO) instruction may be used to mask out interrupts and still allow use of programming routines using INTDS/INTEN instructions. MSKO is also used to turn DMA operation on and off. The single priority level is reflected in the least significant bit (LSB) of the accumulator specified in the MSKO instruction. The least significant bit of the accumulator can be interpreted as follows:

Accumulator Value	Function
Odd (≠ 1)	Masks out (disables) all interrupts
Even (≠ 0)	Masks in (enables if ION=1) all interrupts
= 1	Disables DMA to controllers
= 0	Enables DMA to controllers

1.2.1.2 Programming Polling and Interrupts

Pending interrupts are flagged in the most significant bit (MSB) of the pointer to the device controller input/output control block (IOCB). The DIB instruction with a device code of 77 (CPU) is used to read the IOCB pointers (accumulators 13-15). There are three forms of DIB instructions, one for each of the device controllers. The control field of the I/O instruction is used to designate multiplexer (S), disc controller (C), or tape controller (P). The resulting I/O instructions are:

- DIBS ac,77 read MUX pointer
- DIBC ac,77 read Disc pointer
- DIBP ac,77 read Tape pointer

Two software procedures are used to service I/O devices using the DIB instruction:

1. Polling - Disable interrupts using an INTDS or a MSKO instruction. The DIB with S, C or P must be used to constantly poll the IOCB pointers to check for MSB = 1. Upon detecting a pending interrupt, the program jumps to a device service routine. After servicing of the first pending interrupt encountered, the other IOCB pointers should be checked, since more than one interrupt may be pending at one time.

2. Interrupting - Enable interrupts using an INTEN instruction. The system will be in automatic-transfer mode. When an interrupt is generated by a device requesting service, the processor resets the interrupt-request flag and jumps to a software interrupt handling routine. There the DIB instruction is used to read the IOCB pointers to check for MSB = 1. The controller causing the interrupt is serviced immediately as described in Section 1.2.1.1. Note that if more than one controller is requesting service, each should be serviced at this time. If not, a pending interrupt will be ignored.

   In the automatic-transfer mode, interrupts may also be generated by a controller when either the block transfer is finished or when a status change or error has occurred. In either case the controller sets the interrupt-request flag and the processor responds by resetting the interrupt-request flag and jumping to a software read-status handling routine, as described above.

When using either method, the software must first reset the interrupt-pending bit in the IOCB pointer before jumping to the interrupt servicing routine.

Resetting the interrupt-pending bit at that time ensures the detection of a second interrupt-pending condition while the first one is being serviced.

The IOCB for each multiplexer port contains an operation-complete bit (bit 0 of word 0 or 1). The operation-complete bit is set by the multiplexer when an interrupt is requested. This bit is used by the software to confirm that the pending-interrupt was caused by the IOCB for this device. The operation-complete bit is essential to multiplexer operation, since there is only one multiplexer IOCB pointer and thus one pending-interrupt flag for the four multiplexer ports. Each port is identified by an IOCB. Bit 0 of word 0 (input operations) and word 1 (output operations) designates operation-complete status for that port. When software detects an interrupt pending for the multiplexer, the software must scan the four IOCBs for multiplexer ports to determine which port needs service (bit 0 set to 1). There may be more than one port completing a transfer at any one time. Care must be taken to scan all operation-complete bits, even if one set operation-complete bit has been detected. The operation-complete bit must be reset to 0 by the software in order to prevent redundancy in servicing of I/O ports.

In addition to an operation-complete bit (bit 0, word 7), the disc controller IOCB also contains a summary-error bit in the termination status (bit 1, word 7) which is used to flag the existence of an error condition in disc transfer. Like the operation-complete bit, the summary-error bit is set by the disc controller when an error has been detected and is read by the software to determine the cause of the interrupt. This bit must be reset to 0 by software in order to prevent redundancy in interrupt servicing.

Figure 1-3 is a flowchart showing the sequence of steps that must be taken into account when programming polling and interrupt processing procedures. Note that interrupts other than those caused by the Peripheral Interface Board (i.e., power-fail interrupt) are not covered in this flowchart. Figure 1-3 begins with the polling of the interrupt-pending bit. Arrival at this point in the software procedure may have been from a constant polling sequence or from an interrupt that caused a jump to a poll routine.

Figures 1-4 and 1-5 are flowcharts of parts of the I/O service routines for the multiplexer and for disc or tape. The part of the service routines shown only takes into account testing of operation-complete and summary-error bits. The remainder of the service routine is determined by the programmer.

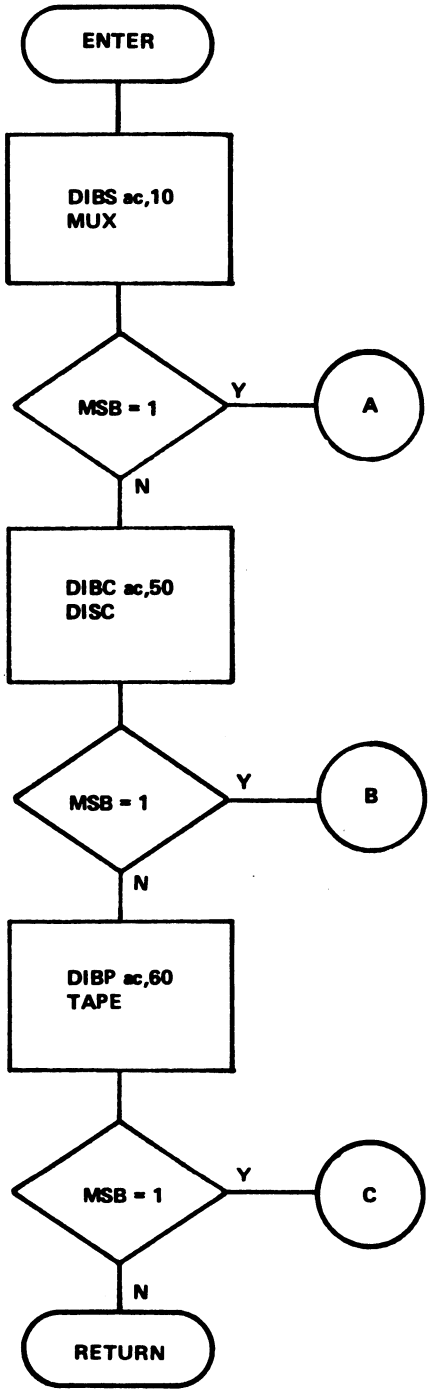


Figure 1-3. Software Polling/Interrupt Handling Flowchart (1 of 2)

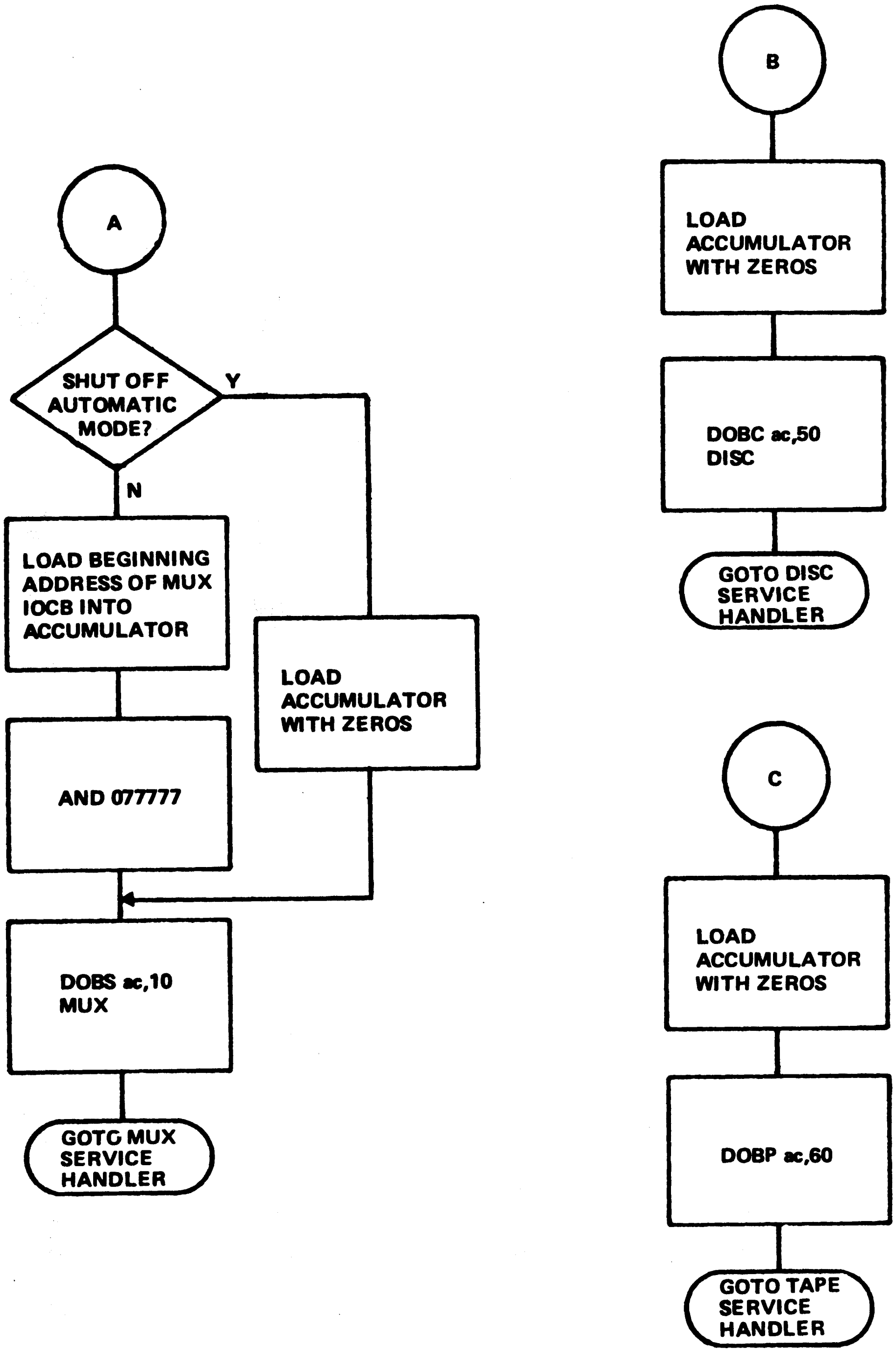

A
SHUT OFF AUTOMATIC MODE?
Y
N
LOAD BEGINNING ADDRESS OF MUX IOCB INTO ACCUMULATOR
AND 077777
LOAD ACCUMULATOR WITH ZEROS
DOBS ac,10 MUX
GOTO MUX SERVICE HANDLER
B
LOAD ACCUMULATOR WITH ZEROS
DOBC ac,50 DISC
GOTO DISC SERVICE HANDLER
C
LOAD ACCUMULATOR WITH ZEROS
DOBP ac,60
GOTO TAPE SERVICE HANDLER

Figure 1-3. Software Polling/Interrupt Handling Flowchart (2 of 2)

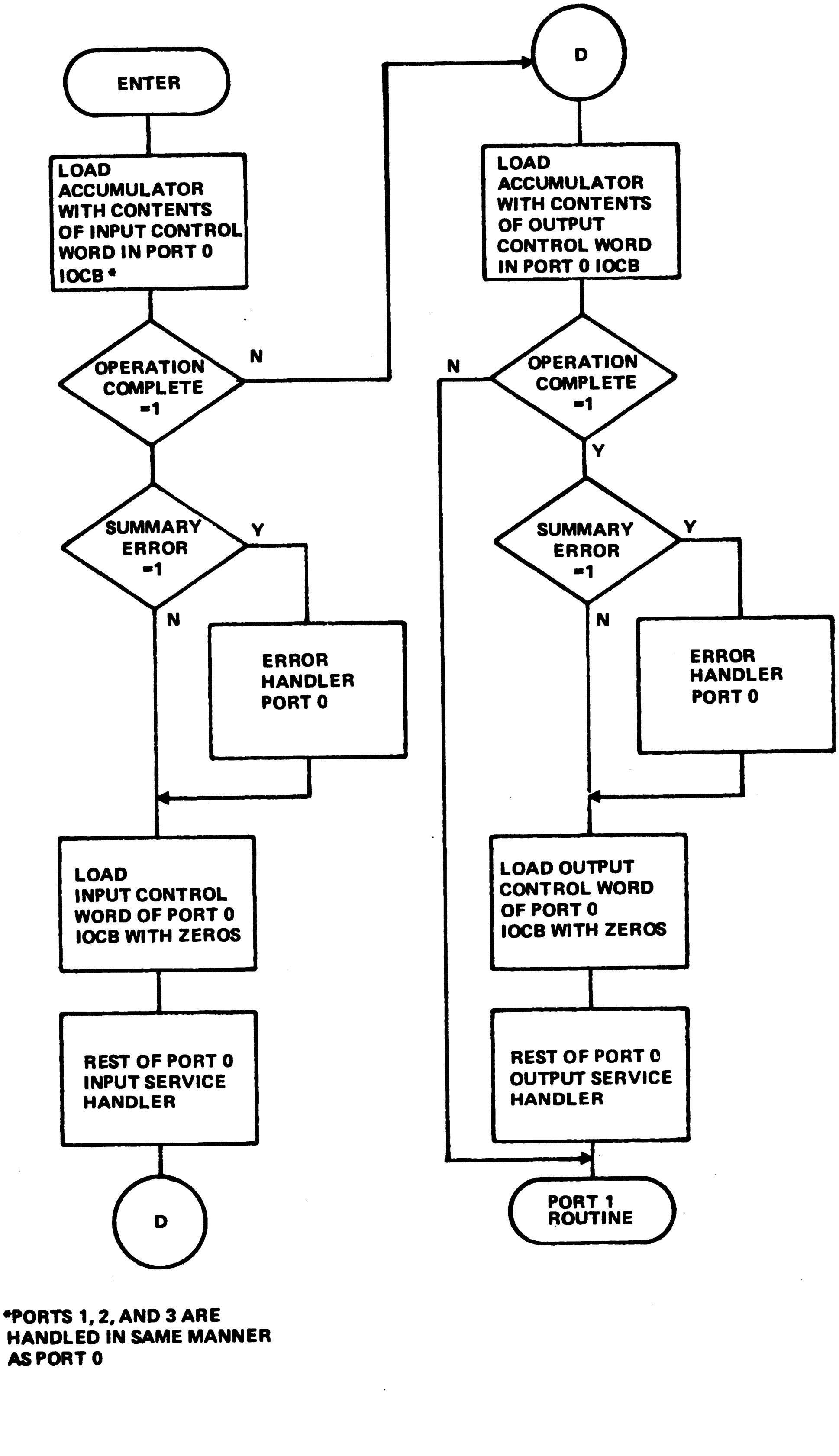

ENTER
LOAD ACCUMULATOR WITH CONTENTS OF INPUT CONTROL WORD IN PORT 0 IOCB *
OPERATION COMPLETE =1
N
SUMMARY ERROR =1
Y
N
ERROR HANDLER PORT 0
LOAD INPUT CONTROL WORD OF PORT 0 IOCB WITH ZEROS
REST OF PORT 0 INPUT SERVICE HANDLER
D
D
LOAD ACCUMULATOR WITH CONTENTS OF OUTPUT CONTROL WORD IN PORT 0 IOCB
N
OPERATION COMPLETE =1
Y
SUMMARY ERROR =1
Y
N
ERROR HANDLER PORT 0
LOAD OUTPUT CONTROL WORD OF PORT 0 IOCB WITH ZEROS
REST OF PORT C OUTPUT SERVICE HANDLER
PORT 1 ROUTINE
*PORTS 1, 2, AND 3 ARE HANDLED IN SAME MANNER AS PORT 0

Figure 1-4. I/O Service Routine Multiplexer Handling Procedures

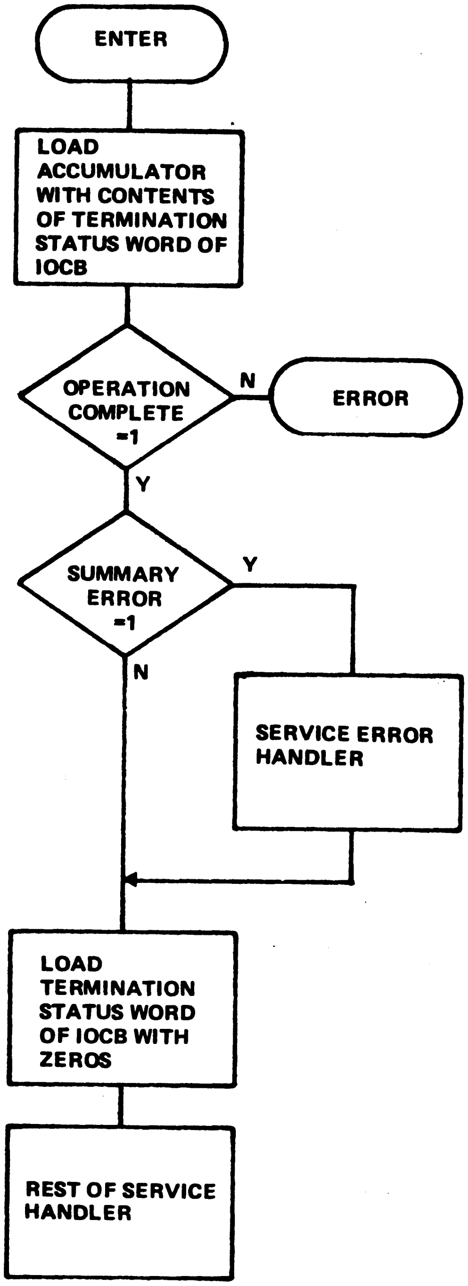

ENTER
LOAD ACCUMULATOR WITH CONTENTS OF TERMINATION STATUS WORD OF IOCB
OPERATION COMPLETE =1
N
ERROR
Y
SUMMARY ERROR =1
Y
SERVICE ERROR HANDLER
N
LOAD TERMINATION STATUS WORD OF IOCB WITH ZEROS
REST OF SERVICE HANDLER

Figure 1-5. I/O Service Routine Disc/Tape Handling Procedures

1.2.2 PROGRAMMED I/O TRANSFERS

For programmed input/output the program directly controls the data transfer between the CPU and the I/O device. Programmed I/O is used by the multiplexer in communications-byte mode. Each data word is transferred between an accumulator specified in the instruction and an I/O device register (see MARK 3 Computer System Manual, Section 5.6).

Programmed I/O uses DIA and DOA instructions to program transfer of data and status information between the CPU and peripheral device controllers. The two programmed I/O instructions are used as follows:

- DIA - Transfers (reads) contents of either the status or data register in the specified port's UART into the specified CPU accumulator. If the device code is even, contents of the status register are transferred. If device code is odd, the contents of the data register are transferred.

- DOA - Transfers (writes) contents of the specified CPU accumulator to either the command or data register in the specified port's UART. If the device code is even, contents of the accumulator are transferred to the command register. If the device code is odd, contents of the accumulator are transferred to the data register.

See Section 2.3 for detailed description of byte-mode operations.

1.2.2.1 Master Terminal Interface

All alphanumeric and control characters are represented by standard ASCII code (see Appendix A) consisting of eight bits, the most significant of which is usually an even parity bit.

The following are programming conventions for handling CRT terminals:

- Instruction Formats - The data-transfer-output instruction transmits bits 8-15 from the specified accumulator to the output-interface register. The input-interface instruction loads the contents of the input-interface register into bits 8-15 and resets bits 0-7 of the specified accumulator.

- Terminal Data Word Output - To transfer a data character from an accumulator to the Port 0 UART, the following instruction is used:

  ```
  DOA ac,11
  ```

  where ac represents one of the four processor accumulators.

- Terminal Command Output - To transfer a command from an accumulator to the Port 0 UART, the following instruction is used:

  ```
  DOA ac,10
  ```

  This instruction causes bits 8-15 of the accumulator to be transferred to the command register of Port 0. Bit 0 of the accumulator is also used for enabling/disabling the real-time clock.

- Terminal Data Word Input - When a key is pressed on the keyboard, the data word is placed in the data register. If interrupts are enabled an interrupt is requested. The program then reads the data word from the data register into bits 8-15 of the specified accumulator with the following instruction:

  ```
  DIA ac,11
  ```

- Terminal Status Input - A change in terminal status causes an interrupt request, if interrupts are enabled. The program must read the status of Port 0 from the status register into bits 8-15 of the specified accumulator using the following instruction:

  ```
  DIA ac,10
  ```

1.2.3 DIRECT MEMORY ACCESS TRANSFERS

Mass storage devices, such as tape drives and discs, can transfer blocks of data at high speeds directly into memory, without requiring programmed I/O instructions for each word transferred, by using the DMA (direct memory access). DMA device interface logic contains both conventional device registers and special data channel logic.

At the start of each instruction cycle, the processor checks to see if a device is requesting DMA service. If a device is requesting service, the DMA transfer is performed before going on with the instruction. The program initiates a DMA transfer by supplying certain parameters to the device registers and starting the device. The device automatically transfers one or more data words to or from memory. When the DMA transfer is completed, the device generates an interrupt (if so enabled). The processor acknowledges the interrupt, checks the operation-complete status and resets the interrupt-request flag.

See Section 1.2.1.2 for interrupt handling procedures.

1.2.3.1 Enabling/Disabling DMA Transfers

The mask-out (MSKO) instruction is used to enable and disable DMA transfers. To enable/disable DMA, special values must be loaded into the CPU accumulator specified in the MSKO instruction. These values are:

Accumulator Value	Function
1	Disables DMA to controllers
0	Enables DMA to controllers

An alternate way of expressing the MSKO instruction is to use a DOB instruction with a device code of 77 octal as follows:

Instruction	Accumulator Value	Function
DOB ac,77	1	Disables DMA to controllers
DOB ac,77	0	Enables DMA to controllers

Alternate methods of disabling DMA transfer include:

- Issue an IORST instruction (clears processor's ION flag)
- Press the Reset switch (gives control to the Virtual Control Panel-MANIP)
- Power-up the system (gives control to the Virtual Control Panel-MANIP)
- Issue a halt instruction (gives control to the Virtual Control Panel-MANIP)

The longer the period of time DMA is disabled, the greater the risk of having an overrun of data in the multiplexer or of having the tape streamer come to a stop. Thus, DMA should be disabled the minimum time possible.

1.2.3.2 Initiating DMA Transfer

Three configurations of the DOB instructions are used to initiate DMA transfers:

```
DOBS ac,77    set Multiplexer IOCB
DOBC ac,77    set Disc IOCB
DOBP ac,77    set Tape IOCB
```

Accumulator ac contains the memory address of the input/output control block (IOCB) for the device specified. Accumulators 13 through 15 are used for IOCB pointers as follows:

Accumulator	Device Controller IOCB
13	Multiplexer
14	Tape
15	Disc

If an IOCB pointer contains a value of 0, the device controller is not being used in the system. The most significant bit (MSB) of this accumulator is set to 1 when an interrupt is pending. For DMA transfers the MSB must be reset to 0 after interrupt processing to avoid erroneous interrupt generation.

When issued to the disc or tape controllers, the DOB instruction signals the corresponding controller. The controller responds by performing an automatic block transfer of data as specified in the IOCB for that device. This data transfer may be one byte or an entire 256-byte block of data, depending on the starting and ending parameters supplied in the IOCB. An example for disc DMA operation would be:

```
IOCB Address:  075000

Instruction:  DOBC 1,50

Accumulator 1 Value:

    A1 = 0111101000000000

Memory address:
    Must be set to 0
    (Reserved for pending interrupt)
```

When issued to the multiplexer, the DOBS instruction sets the software pointer to the multiplexer IOCB pointer but does not initiate any action on the part of the multiplexer. When the DOBS is complete, the software routine must issue DOA instructions to the port whose IOCB address was in the accumulator of the DOBS instruction. The DOA instruction will initiate DMA transfer between the CPU and the peripheral device.

1.2.3.3 Turning Off Automatic Block Transfer Mode For Disc or Tape

DOB instructions with device code 77 may be used to turn off automatic block transfer mode to disc and tape and/or reset the interrupt-pending bit in the IOCB pointer. Both of these functions must be performed by the software for disc and tape transfers. To turn off automatic-block-transfer mode and reset the interrupt-pending bit for disc or tape, issue a DOBC or DOBP with the accumulator set to zero:

Instruction: DOBC 1,50

Device Code: 50 = DISC

Accumulator 1 Value: Al = 0000000000000000

This sequence turns off automatic transfer mode and resets the interrupt-pending bit.

NOTE

> It is important that the most significant bit (MSB) be set to 0. If a DOBC or DOBP is issued with a value of 1 in the MSB, it will initiate processing of an interrupt which may or may not be pending for the controller whose IOCB address is in the accumulator.

1.2.3.4 Turning Off Automatic Block Transfer Mode For Multiplexer

Caution must be taken when using a DOBS to turn off automatic-block-transfer mode to the multiplexer and/or reset the interrupt-pending bit for the multiplexer. If the accumulator contains a value of 0 when the DOBS is issued, the automatic mode is turned off for all ports, regardless of whether a port is in the process of performing a block transfer.

For a majority of operations, the DOBS is used only to reset the interrupt-pending bit. In this case, the contents of the accumulator specified in the DOBS instruction should be set to the address of the multiplexer IOCB pointer with the most significant bit set to 0.

Section 2
MULTIPLEXER

2.1 INTRODUCTION

The multiplexer (MUX) portion of the Peripheral Interface Board supports four asynchronous ports which use the EIA-RS232C interface. Each port is independently software programmable. This allows selection of the number of bits per character, number of stop bits, and odd, even, or no parity. Baud rates are hardware jumper selectable in a range of 110 to 9600 bits per second.

Multiplexer operation is in either byte mode, using programmed I/O instructions, or in automatic-block-transfer mode using direct memory access (DMA) procedures based on information contained in an input/output control block (IOCB) for each port. Receiver and transmitter sections of the port are operated independently of one another. Associated with Port 0 is a real-time clock that, when enabled, causes an interrupt every 10 milliseconds.

2.2 MUX HARDWARE CONFIGURATION

This section describes MARK 3 baud-rate selection.

2.2.1 BAUD-RATE SELECTION

The following baud rates may be selected for MARK 3 communications ports:

110	1200
150	2400
300	4800
600	9600

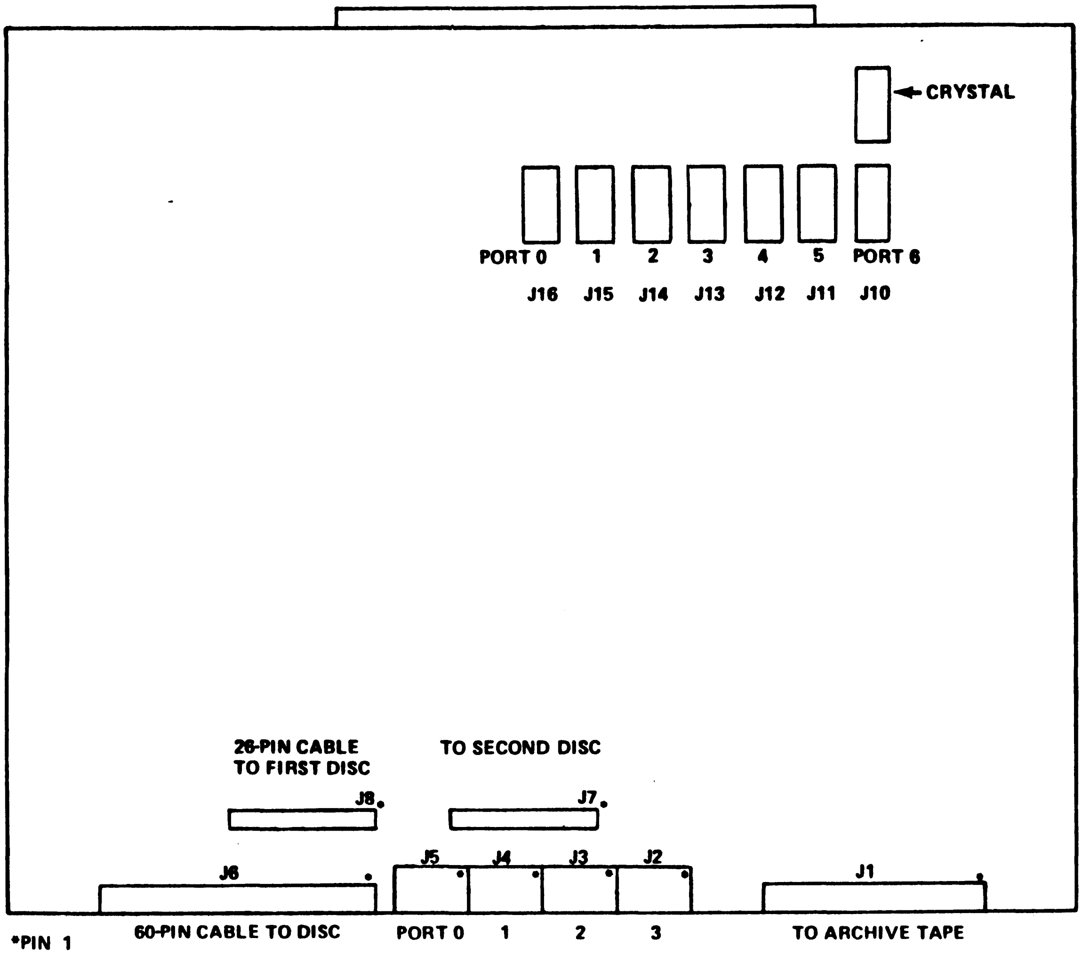
Table 2-1 shows jumpering for each baud rate. Table 2-2 indicates port assignments for baud-rate headers. Figure 2-1 shows PIB board locations for jumpering and header selection.

TABLE 2-1. BAUD RATE JUMPERING

Baud Rate	Jumper	Baud Rate	Jumper
110	16 to 1	1200	12 to 5
150	15 to 2	2400	11 to 6
300	14 to 3	4800	10 to 7
600	13 to 4	9600	9 to 8

TABLE 2-2. PORT ASSIGNMENTS FOR BAUD RATE HEADERS

Standard Ports		Expansion Ports	
Header No.	Port No.	Header No.	Port No.
J16	0	J12	4
J15	1	J11	5
J14	2	J10	6
J13	3		



EXTERNAL CABLES WHICH PLUG INTO THE PIB BOARD:

STREAMER TAPE CABLE	J1
PRINTER OR CRT CABLES	J2 J3 J4
MASTER CRT CABLE	J5
DISC CABLE A	J6
DISC CABLE B – NO. 2	J7
DISC CABLE B – NO. 1	J8

Figure 2-1. PIB Multiplexer Baud Rate Selection and Port Selection Locations

2.3 COMMUNICATIONS BYTE MODE

This section outlines the communications-byte mode programming protocols for the multiplexer controller on the Peripheral Interface Board (PIB). Communications-byte mode can operate in both DMA channel and programmed I/O modes. Section topics include: programmed I/O instructions, UART asynchronous communications, command and status registers, and software polling.

2.3.1 PROGRAMMED I/O INSTRUCTIONS

When the communications controller is used in the byte mode, command, status and data transfer are accomplished through programmed I/O (see Section 1.2.2), utilizing DOA and DIA instructions (see Section 5.6.1). Valid DOA and DIA device codes for each port and their functions are shown in Table 2-3.

TABLE 2-3. DOA, DIA DEVICE CODES

Device Code (Octal)	Read (DIA)	Write (DOA)
	PORT 0	
10	Status Register	Command Register
11	Data Register	Data Register
	PORT 1	
12	Status Register	Command Register
13	Data Register	Data Register
	PORT 2	
14	Status Register	Command Register
15	Data Register	Data Register
	PORT 3	
16	Status Register	Command Register
17	Data Register	Data Register
	PORT 4 (optional)	
20	Status Register	Command Register
21	Data Register	Data Register
	PORT 5 (optional)	
22	Status Register	Command Register
23	Data Register	Data Register
	PORT 6 (optional)	
24	Status Register	Command Register
25	Data Register	Data Register

2.3.2 UART ASYNCHRONOUS COMMUNICATIONS

The UART is an asynchronous communications chip with four internal registers. Two registers are read-only; two are write-only. The UART is programmable, and can select port characteristics: number of bits per character; number of stop bits; and odd, even, or no parity. In byte-mode operation, programming is directed by programmed I/O software procedures. An example of the UART programming process is provided below, using Port 0 and its associated device codes (10 and 11 octal).

2.3.3 COMMAND REGISTER

The following instruction is used to write to the command register (Port 0):

```
DOA ac,10
```

Only the least significant byte of the accumulator (ac) is used. The accumulator bits 8-15 will contain the command information to be loaded into the multiplexer's command register (in this example, Port 0). The command information is used to establish operational parameters for this port. The most significant byte (bits 0-7) must be set to zero (0).

Figure 2-2 shows the format of the command register. Bits, settings, and their functions are shown in Table 2-4.

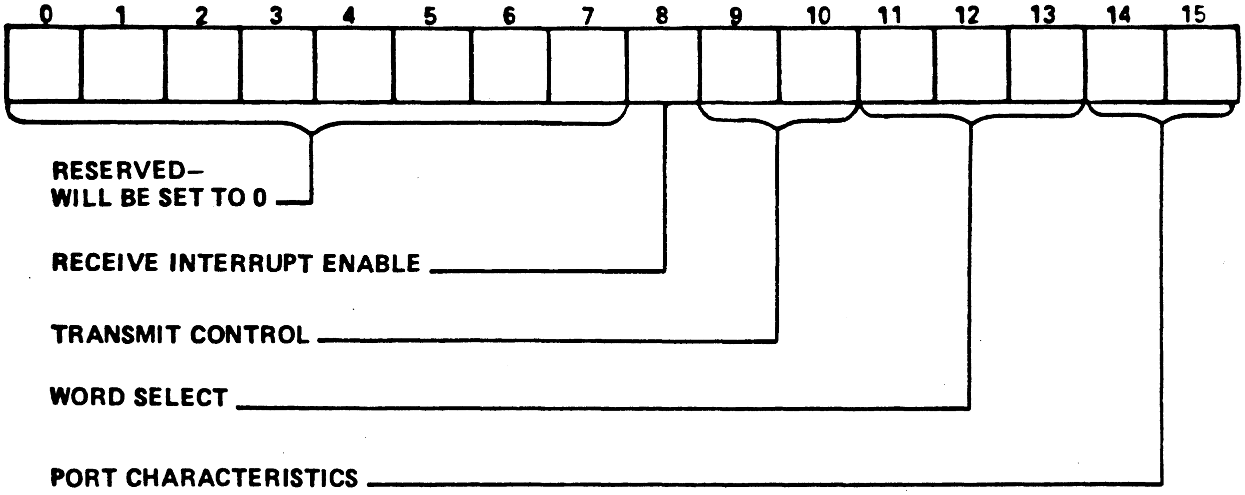


Figure 2-2. Command Register

TABLE 2-4. COMMAND REGISTER BIT FUNCTIONS

Setting			Function
Bits 0-7*			Reserved - Must be set to zero
Bit 8 (MSB)			Receiver Interrupts
0			Receiver interrupts disabled
1			Receiver interrupts enabled
Bit 9	Bit 10		Transmit Control
0	0		Transmit interrupts disabled (RTS low)
0	1		Transmit interrupts enabled (RTS low)
1	0		Transmit interrupts disabled (RTS high)
1	1		Transmit interrupts disabled (RTS low, transmits a break and transmits data output).
Bit 11	Bit 12	Bit 13	Word Select
0	0	0	7 bits + even parity + 2 stop bits
0	0	1	7 bits + odd parity + 2 stop bits
0	1	0	7 bits + even parity + 1 stop bit
0	1	1	7 bits + odd parity + 1 stop bit
1	0	0	8 bits + 2 stop bits
1	0	1	8 bits + 1 stop bit
1	1	0	8 bits + even parity + 1 stop bit
1	1	1	8 bits + odd parity + 1 stop bit
Bit 14	Bit 15		Counter Divide
0	0		Reserved
0	1		Divide-by-16
1	0		Reserved
1	1		Master reset

*Bit 0 for Port 0 controls RTC.

2.3.3.1 Standard Mode Of Operation

During power-up initialization the UART is reset, then programmed to a standard mode of operation. The POINT 4 MARK 3 is delivered with a standard mode of operation established for all ports. This standard mode is as follows:

- Divide-by-16 clock
- 7 bits + even parity + 1 stop bit
- RTS = Low, transmitting interrupt disabled
- Receiving interrupt disabled

2.3.3.2 Operational Mode Modifications

For certain devices, it may be necessary to change the mode of operation set in the UART. This may be done by issuing the following two instructions:

1. Master Reset

```
        DOA ac,10  Contents of ac = 0000000000000011
```

2. Redefine Command Register

```
        DOA ac,10  Contents of ac = 00000000XXXXXX01
            (X = user-defined command register bits)
```

If there is a need to change the mode of operation, two important points should be considered:

1. The UART uses a divide-by-16 clock. If bits 14 and 15 are not set to master reset (1,1) or to divide-by-16 (0,1), erroneous data will result.

2. During the byte mode of systems operation, interrupts may be utilized. If interrupts are used, bits 8 and 10 must be programmed accordingly. If interrupts are not used, the status register may be polled to determine if the transmitter is empty, or if the receiver is full.

2.3.4 STATUS REGISTER

Software utilizes the status register during a polling sequence to determine the following:

1. Receive data register full
2. Transmit data register empty
3. Error condition

The following instruction is used to read the status register for Port 0:

```
    DIA  ac,10
```

The status will be read from the status register into the least significant byte of the accumulator (ac). The PIB sets the most significant byte to zero.

Figure 2-3 shows the status register. Status register bits, instructions, and operational descriptions are shown in Table 2-5.

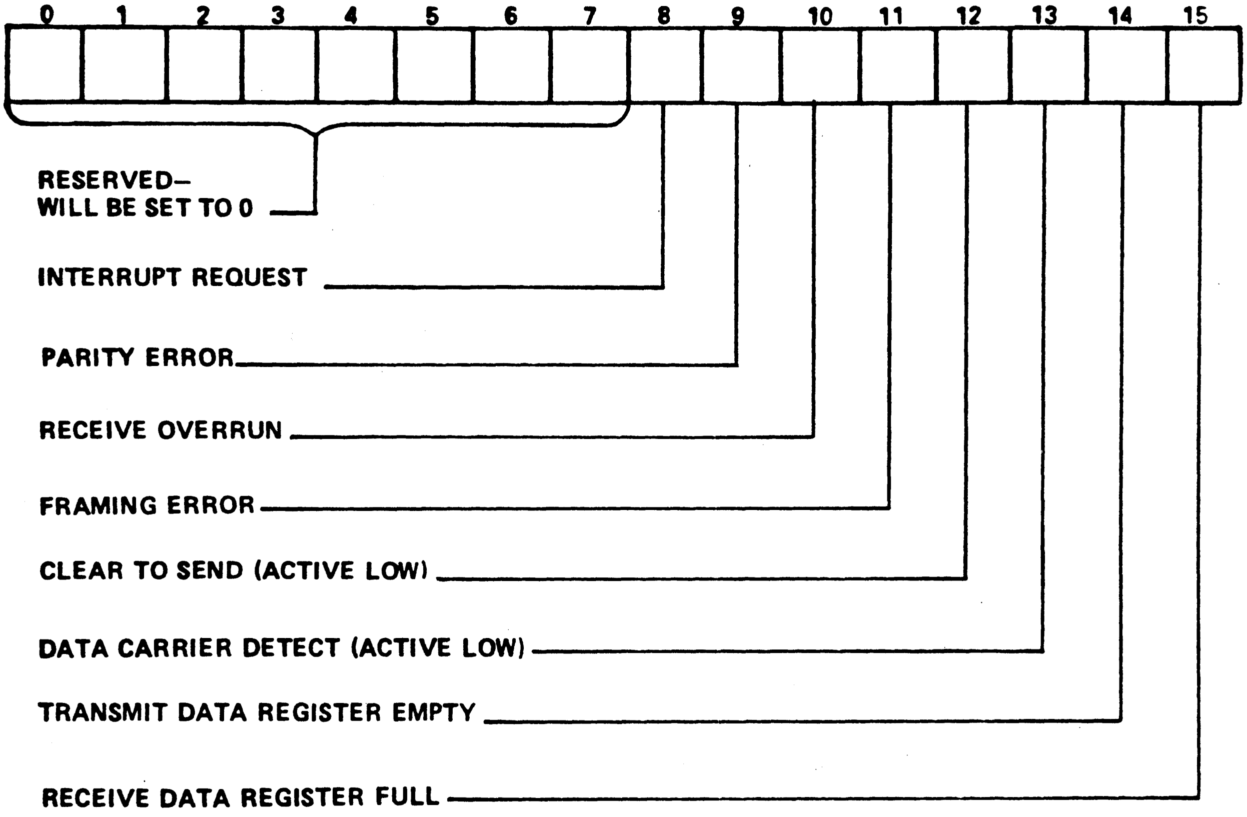

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
RESERVED–
WILL BE SET TO 0
INTERRUPT REQUEST
PARITY ERROR
RECEIVE OVERRUN
FRAMING ERROR
CLEAR TO SEND (ACTIVE LOW)
DATA CARRIER DETECT (ACTIVE LOW)
TRANSMIT DATA REGISTER EMPTY
RECEIVE DATA REGISTER FULL

Figure 2-3. Status Register

TABLE 2-5. STATUS REGISTER BIT OPERATIONS

Bit	Instruction	Operation
0-7	Reserved	Will be set to 0.
8	Interrupt Request (IRQ)	In byte mode, transmit and receive interrupts should be disabled to prevent use of this bit.
9	Parity Error (PE)	Valid as long as a data character with parity error is in the receive-data register and parity is enabled. If no parity is selected, both transmitter parity generator output and receiver parity check results are inhibited.
10	Receiver Overrun (OVRN)	Error flag, which indicates that one or more characters in the data stream have been lost due to failure to read data in the receive-data register. Flag does not occur until a valid character prior to the overrun has been read. Reset after reading the data from the receive-data register, or by a master reset.
11	Framing Error (FE)	When set high, indicates that received character is improperly framed by a start and stop bit. Indicates the existence of an error as long as character is available.
12	Clear to Send (CTS-)	The CTS- input is normally low. If it goes high, it inhibits the transmit-data-register-empty status bit.

(Table continues on next page)

TABLE 2-5. STATUS REGISTER BIT OPERATIONS (Cont)

Bit	Instruction	Operation
13	Data Carrier Detect (DCD-)	Set high when a loss of carrier occurs. Generation depends upon port connection to data terminal equipment (DTE). In a point-to-point connection, this signal is generated from the data terminal ready (DTR-) of the DTE. In a modem connection, it is generated from the modem's carrier detect. Inhibits the receive-data-register-full status bit. Once set, this bit will remain on until reset by reading the status register and the data register, or by a master reset. If it remains high after attempts to clear, it indicates that the DCD- input remains in the high state.
14	Transmit Data Register Empty (TDRE)	Set high when transmit-data register contents have been transferred, and new data may be entered. Low state indicates either that transmit-data register is full, or a high on clear-to-send is inhibiting data transfer.
15	Receive Data Register Full (RDRF)	Set high when received data has been transferred to the receive-data register. Cleared by reading the receive-data register (DIA ac,11) or by a master reset (DOA ac,10 with ac=3). Cleared state indicates that the receive-data register has not yet received new data, or data carrier detect is high.

2.3.5 SOFTWARE POLLING

Section 1.2.1.2 describes the polling procedures used to determine if a port needs service. When the multiplexer is in byte mode, it polls the MUX ports to ensure that each port's data register is empty and may receive another byte of data. Figure 2-4 illustrates this procedure.

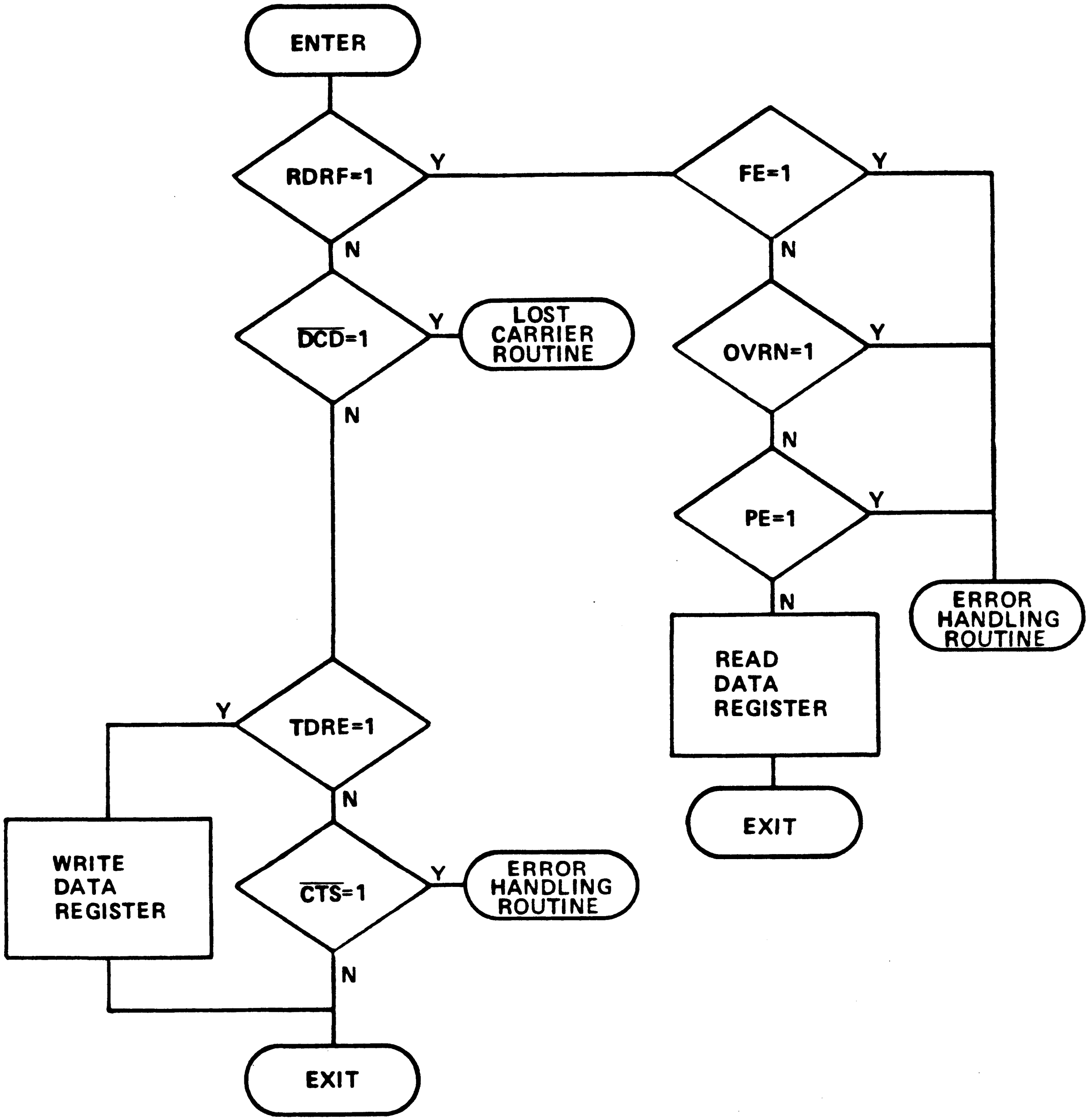


Figure 2-4. Software Polling Flowchart For Output Operations

2.4 COMMUNICATIONS AUTO MODE

This section outlines the communications auto mode programming protocol for the multiplexer controller on the Peripheral Interface Board (PIB). Communications auto mode operates only in DMA channel mode.

Auto mode topics covered include: MUX I/O control block (IOCB), control word definitions, termination status descriptions, and initialization procedures.

2.4.1 MUX I/O CONTROL BLOCK

The MUX Controller supplies software control for the communications ports and their parameters (excluding baud rate) through programmed I/O. Software control includes enabling and disabling the receiver and transmitter, setting character length, determining parity, and selecting the number of stop bits. A real-time clock is also enabled or disabled under software control.

Each port has an I/O control block (IOCB) in system memory; the MUX Controller utilizes four IOCBs (seven IOCBs if expansion board is used). The MUX IOCBs contain information which the MUX uses for automatic data transfers. They do not contain data for selecting port parameters or enabling and disabling ports. All IOCBs must be contiguous in main memory, offset from one another by a multiple of 40 octal. Each IOCB contains input and output control words and buffer pointers for each port.

2.4.2 CONTROL WORD DEFINITIONS

Each IOCB consists of eight words. Six words are used for MUX control: 0, 1, 4, 5, 6, and 7. The three even-numbered words (0, 4, 6) control input; the three odd-numbered words (1, 5, 7) control output. Two words (2, 3) are not used.

The MUX IOCB is shown in Figure 2-5.

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ICW	0	IOC	IN MODE		SC	ERROR STATUS			CAR	INPUT CHARACTER							
OCW	1	OOC	OUT MODE		SC	0	0	0	RTC	OUTPUT CHARACTER							
	2	NOT USED															
	3	NOT USED															
IBP	4	INPUT BYTE POINTER															BIN
OBP	5	OUTPUT BYTE POINTER															BIN
LIB	6	LAST INPUT BYTE POINTER															BIN
LOB	7	LAST OUTPUT BYTE POINTER															BIN

Figure 2-5. MUX Input/Output Control Block

2.4.3 INPUT OPERATIONS

The three even-numbered words of each IOCB (0, 4, 6) control input. Word 0 is the input control word (ICW); word 4 is the input byte pointer (IBP); word 6 is the last input byte pointer (LIB).

2.4.3.1 Input Control Word

The structure of word 0 is shown in Figure 2-6. Table 2-6 defines the input control word, indicating each bit, the appropriate function, and the resultant operation.

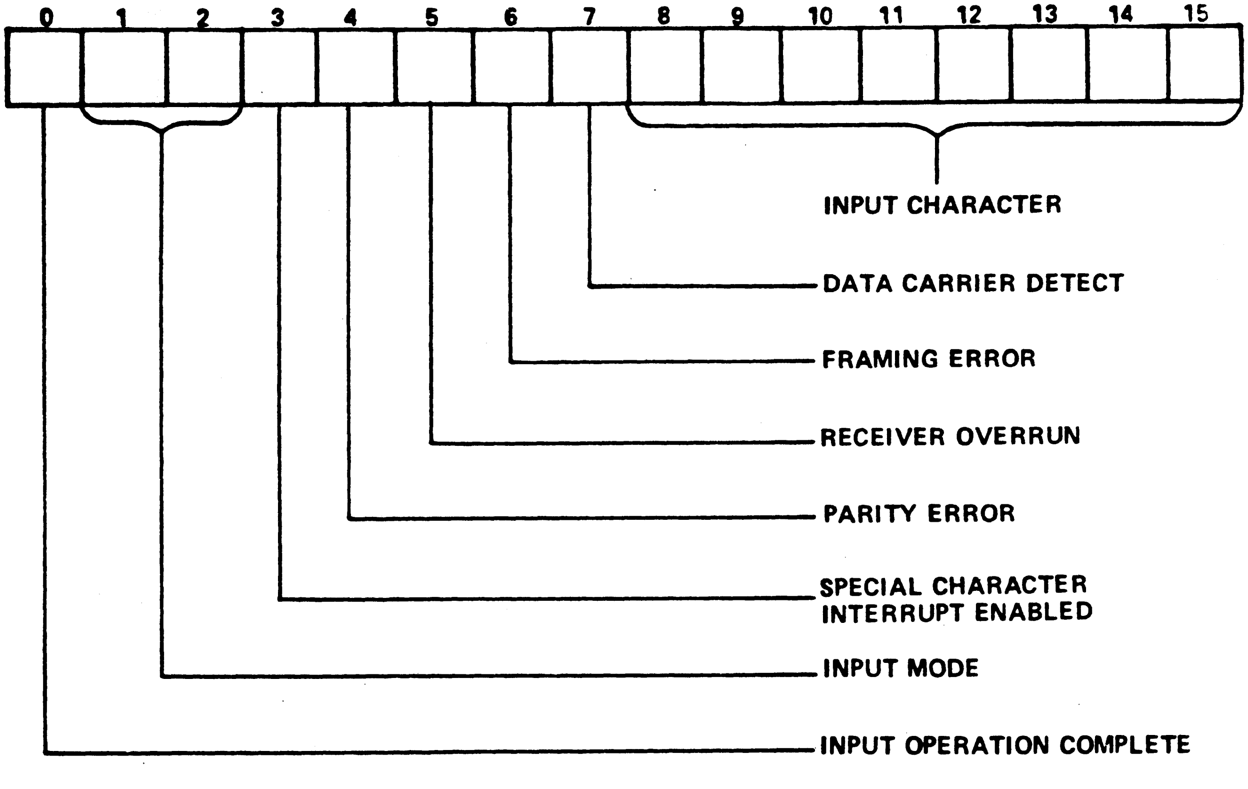


Figure 2-6. Input Control Word - Word 0

TABLE 2-6. INPUT CONTROL WORD DEFINITIONS (WORD 0 OF EACH CONTROL BLOCK)

Bit	Instruction	Operation
0	IOC (Input Operation Complete)	Set to one (1) if any of the following conditions occurs: 1. Input buffer is full 2. The receiver detects: a. Parity error (if parity is not inhibited) b. Framing error (break detect) c. Receiver overrun (DMA is disabled for an extended period, or another controller uses DMA cycles for an unreasonably extended period, i.e., greater than one character time, approximately 1 millisecond at 9600 baud). 3. IOCB is in single-character-input mode, and an incoming character is received and stored in bits 8-15 of ICW. 4. Special-character-interrupt request is enabled and a special character is received (<40 octal or 174 through 177 octal). 5. A loss of carrier occurs (carrier detect, which is normally active low, goes high) during a receive operation. At the same time that IOC is set to one (1), the controller sets the interrupt-pending bit and the interrupt-request flag. The IOC must be cleared by the MUX interrupt service program. There is approximately one (1) character time to accomplish this. If another input character is received before the IOC is cleared, the original character will be overwritten in bits 8-15.

TABLE 2-6. INPUT CONTROL WORD DEFINITIONS (Cont)
(WORD 0 OF EACH CONTROL BLOCK)

Bit	Instruction	Operation
1 & 2	Input Mode	The input mode is determined as follows:
Value		
0 0		Single Character Input Mode - each incoming character is placed in bits 8-15 of ICW. IOC is set, interrupt-pending bit is set, and an interrupt generated.
0 1		Not used (illegal)
1 0		Automatic Input - incoming characters are placed in input buffer defined by input byte pointer (IBP) and last input byte (LIB), until the buffer is filled, if no special interrupt conditions are encountered.
1 1		Automatic Input With Echo - same as automatic input, except that each character placed in input buffer is also automatically echoed (output). Note that any character producing an interrupt (other than buffer full) is not automatically echoed. **NOTE** When using automatic-input-with-echo mode, software must ensure that the transmitter interrupt section is disabled for a specified port. If interrupts are not disabled, the controller will both echo characters and transmit characters from the output buffer. With transmitter interrupts disabled, the controller will ignore the output section of IOCB for a specified port and characters will be echoed correctly.

TABLE 2-6. INPUT CONTROL WORD DEFINITIONS (Cont)
(WORD 0 OF EACH CONTROL BLOCK)

Bit	Instruction	Operation
3	Special Character Interrupt Enabled	If this bit is set to one (1) by the programmer, the controller examines each incoming character to determine if it is a special ASCII character (<40 octal or 174 through 177 octal). If the incoming character is a special character, the controller stores the character in bits 8-15 of the input control word, sets the interrupt-pending bit, and generates an interrupt request. The character will not be stored in the input buffer. In addition, the controller sets the most significant bit of all incoming characters to one (1).
4	Parity Error	Set to one (1) by hardware when the incoming character has a parity error. When this condition occurs, the IOC bit is set to one (1), and the character is stored in ICW.
5	Receiver Overrun	Set to one (1) when the controller cannot service the receiving section of the specified port before a character is overwritten. This can occur if DMA is disabled for an extended period, or if another controller uses an excessive number of DMA cycles. In this condition, the IOC bit is set to one (1), and data in receive-data register is stored in ICW.
6	Framing Error	Set to one (1) by hardware when an incoming character has a framing error (a zero bit at the point where a stop bit was expected). Sets IOC bit to one (1) and stores the incoming character in ICW.

(Table continues on next page.)

TABLE 2-6. INPUT CONTROL WORD DEFINITIONS (Cont)
(WORD 0 OF EACH CONTROL BLOCK)

Bit	Instruction	Operation
7	Carrier Detect	Set to one (1) when a loss of carrier occurs (data carrier detect, which is normally active low, goes high). When this condition occurs, receiver section is disabled. Sets IOC bit to one (1) and stores data in the receive-data register in ICW.
8-15	Input Character	Each incoming character is stored in bits 8-15. Characters that cause the IOC bit to be set are: 1. The last character stored in input buffer, causing an input buffer-full condition. **EXCEPTION** If an input buffer ends in mid-word, the byte placed in ICW is the right half of the last buffer word, and not the last incoming character. Software must check for buffer full before it checks for a special character. 2. An incoming character which is overrun, or which has caused a parity or framing error. 3. An incoming character which is a special character. 4. The character in the receive-data register of the UART when carrier detect went high.

2.4.3.2 Input Termination Status

When an input operation is completed, the appropriate status is written into the ICW by the controller.

When the status is written, the input-operation-complete (IOC) bit is set to one (1). The incoming character which completed the operation is written in the right half of the right byte of word 0 (bits 8-15). The controller also sets the interrupt-pending bit and the interrupt-request flag.

When the interrupt occurs, the software has approximately one character time to determine which port's input operation is complete, and reinitialize the IOCB after interpreting the status word by creating a new input memory buffer address and resetting the IOC bit. Otherwise, a MUX overrun may occur.

2.4.3.3 Input Byte Pointer

Word 4 of the IOCB is the input byte pointer (IBP). IBP and LIB (last input byte) are only used in automatic-input mode. IBP must be set by the program to one (1) less than the first byte address of the automatic input buffer. Each time the MUX stores an incoming byte, it will increment IBP. The MUX uses the least significant bit (bit 15) as the byte indicator (BIN). If BIN = 0, the byte will be stored in the left half of the word addressed. If BIN = 1, it will be stored in the right half of the word addressed. IBP always points to the last input byte stored. If IBP ≥ LIB, an interrupt will be generated by the first incoming character, which will be stored at IBP.

2.4.3.4 Last Input Byte

Word 6 of the IOCB is the last input byte (LIB). It is set up by the program to the last byte address of the auto input buffer. The MUX will generate an input-done interrupt when a byte is stored at this address. IBP will then be equal to LIB.

2.4.4 OUTPUT OPERATIONS

The three odd-numbered words of each IOCB (1,5,7) control output. Word 1 is the output control word (OCW); word 5 is the output byte pointer (OBP); word 7 is the last output byte pointer (LOB).

2.4.4.1 Output Control Word

The structure of word 1 is shown in Figure 2-7.

Table 2-7 defines the output control word, indicating each bit, the appropriate function, and the resultant operation.

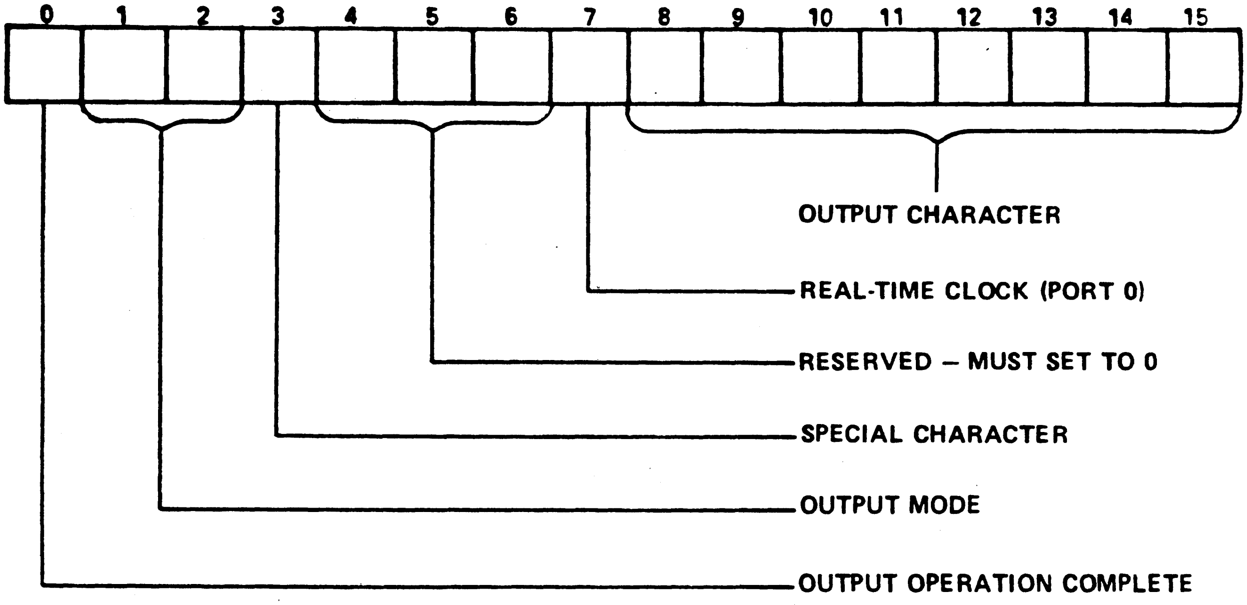


Figure 2-7. Output Control Word - Word 1

TABLE 2-7. OUTPUT CONTROL WORD DEFINITIONS (WORD 1 OF EACH CONTROL BLOCK)

Bit	Instruction	Operation
0	OOC (Output Operation Complete)	Set to one (1) if the operation has been terminated due to one of the following conditions: 1. If MUX is in single-character mode when the output byte is read for transmission to the port. The program must reload the output control word at an average of once per character time and a maximum of two character times for uninterrupted output to result. 2. If MUX is in automatic-buffer-output mode when the last byte of the automatic buffer is read for transmission. 3. If MUX is in automatic-buffer-output mode with special-character interrupt enabled, and if the transmitted character is a special control character, automatic output is terminated after transmission of the special character. The character which completed the operation is stored in the OCW. The controller sets the interrupt-pending bit and the interrupt-request flag. The OOC bit must be cleared by the MUX interrupt service program. If the IOCB is re-initialized at an average of once per character time, and within a maximum of two character times, uninterrupted transmission will result. When the last byte of the output buffer is transferred, OOC is the only bit set.

TABLE 2-7. OUTPUT CONTROL WORD DEFINITIONS (Cont) (WORD 1 OF EACH CONTROL BLOCK)

Bit	Instruction	Operation
1 & 2	Output Mode	Written by the program and read by MUX to determine the type of output.
Value		
0 0		Idle (no output)
0 1		Single Character Output Mode - the data byte in bits 8-15 of the OCW is transmitted, and the OOC bit is set.
1 0		Not Used (Illegal)
1 1		Automatic Buffer Output - characters are automatically transmitted from the output buffer, as defined by the output byte pointer (OBP) and the last output byte pointer (LOB).
Bit		
3	Special Character	Used only in the automatic-output mode. If this bit is set to one (1), MUX tests each outgoing character to determine if it is an ASCII special character (<40 octal or 174 through 177 octal). If it is, MUX transmits the character, sets the output-operation-complete bit in the output termination word to one (1), and stores the special character in bits 8-15 of the OCW. It then sets the interrupt-pending bit and generates an interrupt request. Automatic output is terminated.
4-6		Must be set to zero.

(Table continues on next page.)

TABLE 2-7. OUTPUT CONTROL WORD DEFINITIONS (Cont) (WORD 1 OF EACH CONTROL BLOCK)

Bit	Instruction	Operation
7	Real-Time Clock	Used only on Port 0. Set to one (1) if the software has enabled the real-time clock and the controller has been interrupted by the clock. (The real-time clock interrupts once each 10 milliseconds.) The interrupt-pending bit is set, and an interrupt request is generated. Does not set the OOC bit, or terminate automatic operation.
8-15	Output Character	In single-character mode, this byte is sent to the port for transmission. In automatic-output mode, each outgoing character which causes the OOC bit to be set is stored in this byte. These characters meet the following conditions: 1. An outgoing character is a special character, and the special-character-enable bit is set. 2. The outgoing character is the last character in the output buffer.

2.4.4.2 Output Termination Status

When an output operation is completed, the appropriate status is written into the OCW by the controller. When the status is written, the output-operation-complete (OOC) bit is set to one (1). The outgoing character which completed the operation is written in the right half of word 1 (bits 8-15). The controller also sets the interrupt-pending bit and the interrupt-request flag.

When the interrupt occurs, the software has approximately one character time to determine which port's output operation is complete, and to perform one of the following services:

1. If there is no more output to be done, disable the transmitter section of the corresponding port through programmed I/O.

CAUTION

> Exercise caution when using this method because the transmitter double-buffers characters. Software may disable the transmitter interrupt, but must not issue a master reset until the buffered characters have been transmitted or a character may be lost.

2. Reinitialize the IOCB after interpreting the status word by creating a new input memory buffer address and resetting the IOC bit. This must be effected within two character times for uninterrupted output to result.

If neither function is completed, MUX will examine the OCW word after each executed instruction, which will create a significant system slow-down.

2.4.4.3 Output Byte Pointer

Word 5 of the IOCB is the output byte pointer (OBP). OBP and LOB (last output byte) are used only in the automatic-output mode. OBP must be set up by the program to one (1) less than the first byte address of the automatic output buffer. Each time the MUX is ready for an output byte, it will increment OBP. The MUX fetches the byte for transmission from the appropriate half of the word address given by the most significant 15 bits of the OBP (word address) in conjunction with the byte indicator (BIN). If BIN = 0 the left byte will be output; if BIN = 1 the right byte will be output. OBP always points to the last byte transmitted.

2.4.4.4 Last Output Byte

Word 7 of the IOCB is the last output byte (LOB). It is set by the program to the last byte address of the automatic output buffer. When the byte at that address is picked up for transmission, the MUX will generate an output-done interrupt (OBP will then be equal to LOB). An automatic buffer may contain as little as one byte. In this case, initially LOB = OBP + 1. If initially LOB < OBP + 1, the MUX will transmit one byte from OBP + 1, set OOC and produce an interrupt.

2.5 INITIALIZATION PROCEDURES

This section covers setup procedures for the MUX controller, and includes the following steps: setting the IOCB starting address pointer; initialization guidelines; enabling and disabling ports and defining port parameters; and enabling and disabling transmitters and receivers.

2.5.1 SETTING IOCB STARTING ADDRESS

All port IOCBs must be contiguous in main memory. Software only sets up IOCBs for ports that are to be enabled, but must reserve space for all possible port IOCBs in the contiguous memory block. For example:

Memory Location	Port
1000	Port 0
1040	Port 1
1100	Port 2
1140	Port 3

When MUX is in automatic-transfer mode, the controller must be informed of the IOCB starting address by issuing a DOBS instruction with the contents of the accumulator containing the IOCB address. For example:

```
    DOBS ac,77

    Contents of ac = 1000
```

When the controller receives a DOBS, the memory address is stored in a register. No other activity occurs as a result of the DOBS (unlike the tape and disc controllers). Software has control over enabling and disabling each individual port through programmed I/O. See Section 1.2.2 for detailed description of multiplexer port initialization under programmed I/O control.

2.5.2 INITIALIZATION GUIDELINES

Four basic enabling and disabling considerations are listed below:

1. If an input or output section of an IOCB has not been previously set up by software, the corresponding receiver or transmitter of the port should not be enabled.

2. If a port is to operate in automatic-echo mode for incoming characters, the transmitter section of that port must be disabled.

3. Software may disable the receiver section of a port after the input operation is complete. There is a risk of having an overrun condition on any further incoming characters if the receiver section is disabled for an extended period.

4. When an output operation is complete, there is approximately one character time for software to either update the IOCB for a new transmission, or disable the transmitter with a DOA instruction.

2.5.3 ENABLING/DISABLING PORTS AND PORT PARAMETERS

After software has set the IOCBs and issued a DOBS ac,77 instruction, individual ports must be enabled. A list of instructions used for enabling and disabling the individual ports follows:

DOA ac,10	Port 0
DOA ac,12	Port 1
DOA ac,14	Port 2
DOA ac,16	Port 3

The accumulator contains the necessary information for port programming in the form of command register input as described in Section 2.3.3.

When the system powers up, the following parameters are standard:

- Receiver interrupts disabled
- Transmitter interrupts disabled
- Word length = 7 bits
- Even parity
- One stop bit

Software can change any of these parameters using a DOA instruction with the new port parameters in the specified accumulator.

2.5.3.1 Master Reset

Software may reinitialize the port at any time by issuing a master reset DOA. It should be used only to initialize the MUX when receive, transmit and status data are lost. The master reset should never be used once the port has been initialized and continuous operations are to be performed. An example of the master reset DOA instruction is:

```
DOA ac,10
```

in which the accumulator contains the value 3 as shown in Figure 2-8.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0 ⟵													⟶ 0	1	1

Figure 2-8. Master Reset Instruction Format

2.5.3.2 Port Initialization

Once a master reset DOA has been issued, software re-programs the word length, parity, and stop-bit parameters, using another DOA instruction. An example of the port-characteristic instruction for Port 0 is:

```
    DOA ac,10
```

where the contents of the accumulator have the format as shown in Figure 2-9.

Bits 11, 12, and 13 are defined during initialization (see Table 2-6).

This procedure should be followed for each port when first initializing the MUX. The procedure should only be used for an initialization routine.

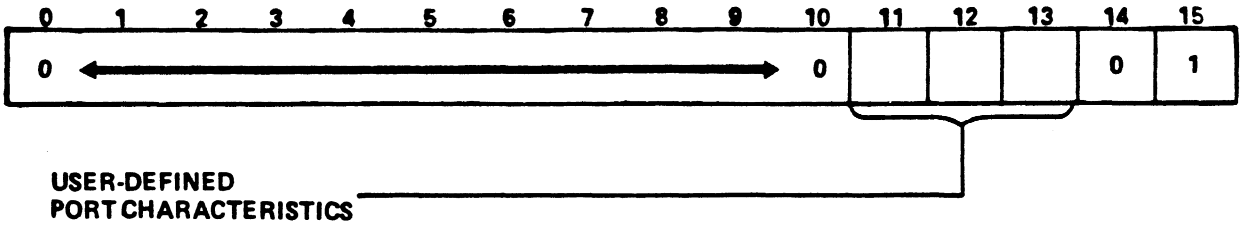


Figure 2-9. Port Initialization Instruction Format

2.5.3.3 Enabling/Disabling Transmitters And Receivers

The receiver and transmitter sections for each port operate independently, and may be enabled or disabled independently. An example of the enable/disable instruction for Port 0 is:

```
    DOA ac,10
```

where the contents of the accumulator have the format as shown in Figure 2-10.

Each DOA instruction updates the port-command register. If previous values have been defined (such as port parameters) for the specified port, they should be repeated on all DOA instructions used to define operation modes. Table 2-8 defines the bit settings used to enable/disable operation.

The MUX controller uses a hardware "attention flag" for servicing ports in the automatic mode. Bits 8 and 10 of the accumulator, when set to enable, allow a port's receiver or transmitter to generate the attention flag.

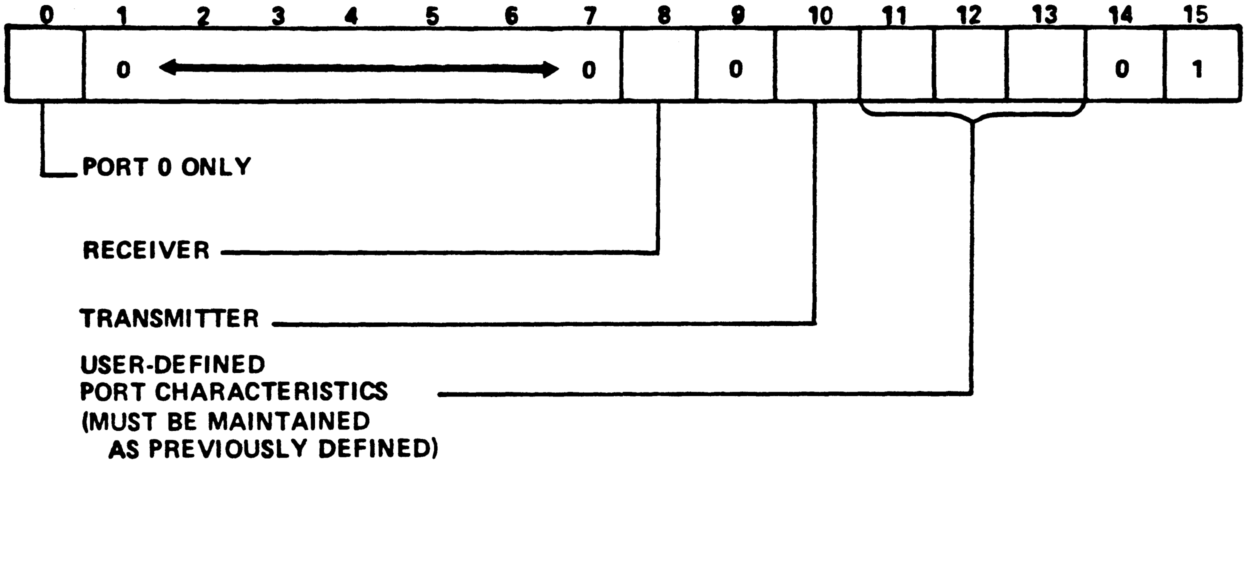


Figure 2-10. Transmitter/Receiver Instruction Format

TABLE 2-8. ENABLING/DISABLING OPERATION

Bit	Name	Operation
0	Real-Time Clock Enable (Port 0 Only)	0 = Disable Real-Time Clock 1 = Enable Real-Time Clock
8	Receiver	0 = Disable Receiver Section 1 = Enable Receiver Section
10	Transmitter	0 = Disable Transmitter Section 1 = Enable Transmitter Section
11-13	Port Parameters	Set to Values Defined During Initialization

2.6 POLLING MUX INTERRUPT

The following instruction is used to poll the MUX interrupt-pending bit:

```
    DIBS ac,77
```

This instruction causes the controller to transfer the contents of the MUX pointer (interrupt-pending bit + IOCB memory address) into the specified accumulator. Software may then test the most significant bit (interrupt-pending bit); MSB = 1 indicates a request for service.

The interrupt-pending bit is reset by resetting the MSB in the specified accumulator and issuing the following instruction:

```
    DOBS ac,77
```

The accumulator value will be loaded into the MUX IOCB pointer, thus resetting the interrupt-pending bit.

2.7 DEACTIVATING MUX

The instruction format for temporarily deactivating and then reactivating the MUX is as follows:

```
    DOB ac,77
```

When the instruction is first issued, with the accumulator equal to 0, it causes the MUX to ignore all port requests for service.

When the instruction is reissued, with the accumulator containing the IOCB pointer (see Section 2.5), the MUX is reactivated.

Section 3
DISC DRIVE INTERFACE

3.1 INTRODUCTION

The POINT 4 MARK 3 Disc Controller is designed to handle up to two SMD/CMD/LMD-type drives. The controller provides high-speed, direct-memory access between the disc drives and the POINT 4 MARK 3 CPU. The following are features of the controller:

- Handles SMD/CMD/LMD/FMD/MMD drives
- Interfaces up to two drives
- Drives of different type and manufacture may be connected to each port
- DMA transfer rates up to 1.25 megabytes per second
- Control block-oriented instructions
- Read-verify data operations
- Complete software control of sector addressing
- Status and error reporting on completion of operation

Disc operation control is handled through use of an input/output control block (IOCB). Actual data transfer is performed in DMA mode.

3.1.1 PERFORMANCE CHARACTERISTICS

Drives per Controller:
Two

Drive Type:
SMD/CMD/LMD/FMD/MMD drives supporting the sector mark interface signal

DMA Transfer Rate:
1.25 megabytes per second

Sector Size:
Header - 4 words   Header CRC - 1 word
Data - 256 words   Data CRC - 1 word

Controller Device Code:
50-55

Drive Port Assignments:
Port A-J8
Port B-J7

I/O Instructions:
Input - DIA
Output - DOA

DMA (Device Code 77):
DOBC/DIBC = DISC

3.1.2 MULTI-DRIVE CONNECTION

The disc controller supports two SMD, CMD or LMD drives, which must have a daisy-chain connection, as illustrated in Figure 3-1. The two drives need not be of the same type, nor produced by the same manufacturer.

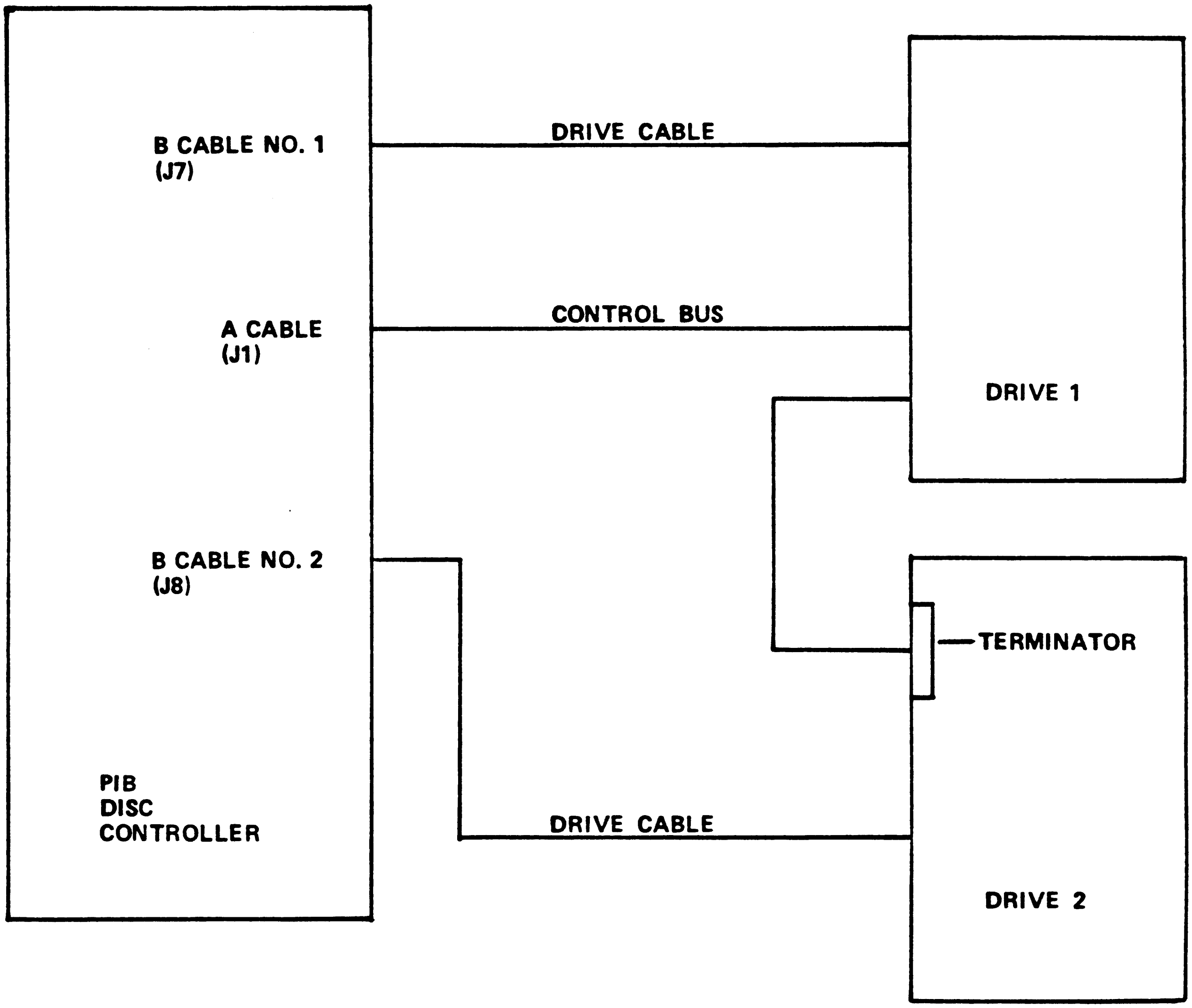


Figure 3-1. Daisy-Chain Drive Connection

3.1.3 DISC DRIVE CABLING

Two types of cables connect the Peripheral Interface Board to the disc drives:

1. Drive control bus (A Cable)

2. Drive cables (B Cables)

These cables are described in the following subsections.

3.1.3.1 Drive Control Bus (A Cable)

Cable A is the drive control bus. This bus leaves the PIB at connector J1 and is daisy chained between drives (see Figure 3-1). Figure 3-2 illustrates the signals carried on the control bus and Table 3-1 defines these signals.

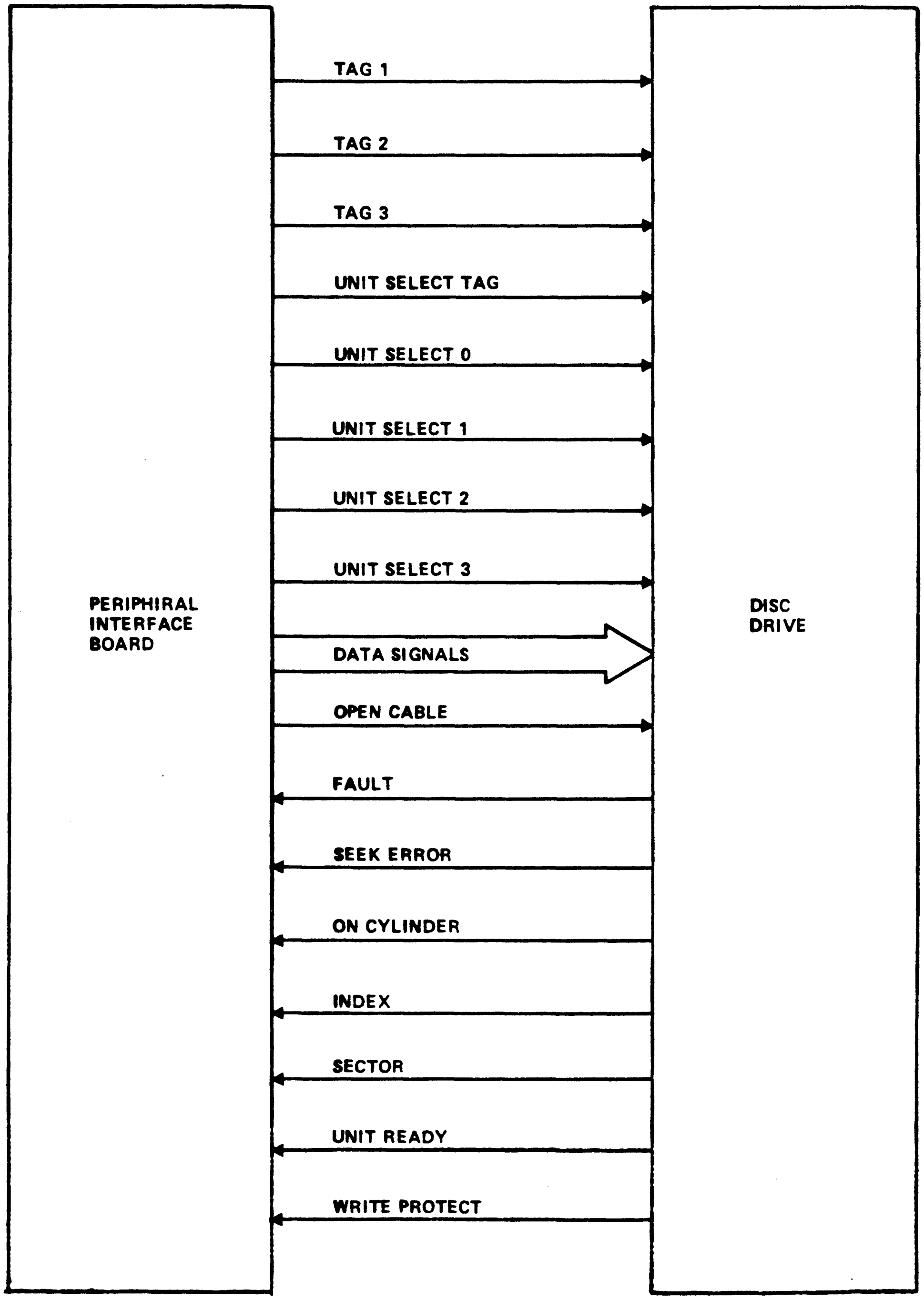


Figure 3-2. PIB/Disc Drive Differential Bus Signals

TABLE 3-1. DISC DRIVE DIFFERENTIAL BUS SIGNALS (A CABLE PIN ASSIGNMENT)

Pin	Signal	Direction	Description
1,31	Tag 1	Out	Used to strobe the cylinder address to disc drive. For CMD drives, a tag 2 must precede a tag 1 for a volume change.
2,32	Tag 2	Out	Used to strobe the head address to disc drive. For CMD drives, the volume is also sent to the drive.
3,33	Tag 3	Out	Used to strobe control bits. **NOTE** Bits 0 through 9 are defined according to tag. If tag 1, all bits are used to define the cylinder address. If tag 2, all bits are used to define the head address. For tag 3, each bit is defined as described below.
4,34	Bit 0	Out	Enables the write drivers. Must never be on simultaneously with read drivers.
5,35	Bit 1	Out	Enables the read drivers. Must never be on simultaneously with write drivers.
6,36	Bit 2	Out	Offsets the servo from the nominal on-cylinder toward the spindle.
7,37	Bit 3	Out	Offsets the servo from the nominal on-cylinder away from the spindle.
8,38	Bit 4	Out	Clears a fault only if the condition no longer exists.
9,39	Bit 5	Out	When issued with bit 0 or 1, enables address-mark writing or reading.

TABLE 3-1. DISC DRIVE DIFFERENTIAL BUS SIGNALS (Cont)
(A CABLE PIN ASSIGNMENT)

Pin	Signal	Direction	Description
10,40	Bit 6	Out	Causes seek to track 0, resetting of head register. For CMD drives, selects head 0, volume 0.
11,41	Bit 7	Out	Causes strobing of data earlier than normal.
12,42	Bit 8	Out	Causes strobing of data later than normal.
13,43	Bit 9	Out	Selects the unit, if enabled, and if priority does not exist for opposite channel.
14,44	Open Cable	Out	Disables interface if the control bus cable is disconnected.
15,45	Fault	In	Indicates that a fault exists on the drive.
16,46	Seek Error	In	Indicates that a seek error has occurred.
17,47	On Cylinder	In	Indicates that the heads are positioned over a track.
18,48	Index	In	A signal occurring once per revolution in coordination with the leading edge of sector 0.
19,49	Unit Ready	In	Indicates the drive is up to speed.
20,50	-	-	Reserved
21,51	-	-	Reserved
22,52	Unit Sel Tag	Out	Strobes the binary address formed by unit select 0, 1, 2, and 3.
23,53	Unit Sel 0	Out	Unit select 0.
24,54	Unit Sel 1	Out	Unit select 1.

TABLE 3-1. DISC DRIVE DIFFERENTIAL BUS SIGNALS (Cont)
(A CABLE PIN ASSIGNMENT)

Pin	Signal	Direction	Description
25,55	Sector	In	Indicates that a sector mark has been detected.
26,56	Unit Sel 2	Out	Unit select 2.
27,57	Unit Sel 3	Out	Unit select 3.
28,58	Write Prot	In	Disables write gate.
29,59	-	-	Reserved
30,60	-	-	Reserved

3.1.3.2 Drive Cables (B Cables)

The PIB requires one B cable per drive for use in synchronization and data transfers. These cables are connected to the PIB at connectors J7 and J8. Figure 3-3 illustrates the signals carried on the drive cables and Table 3-2 defines these signals.

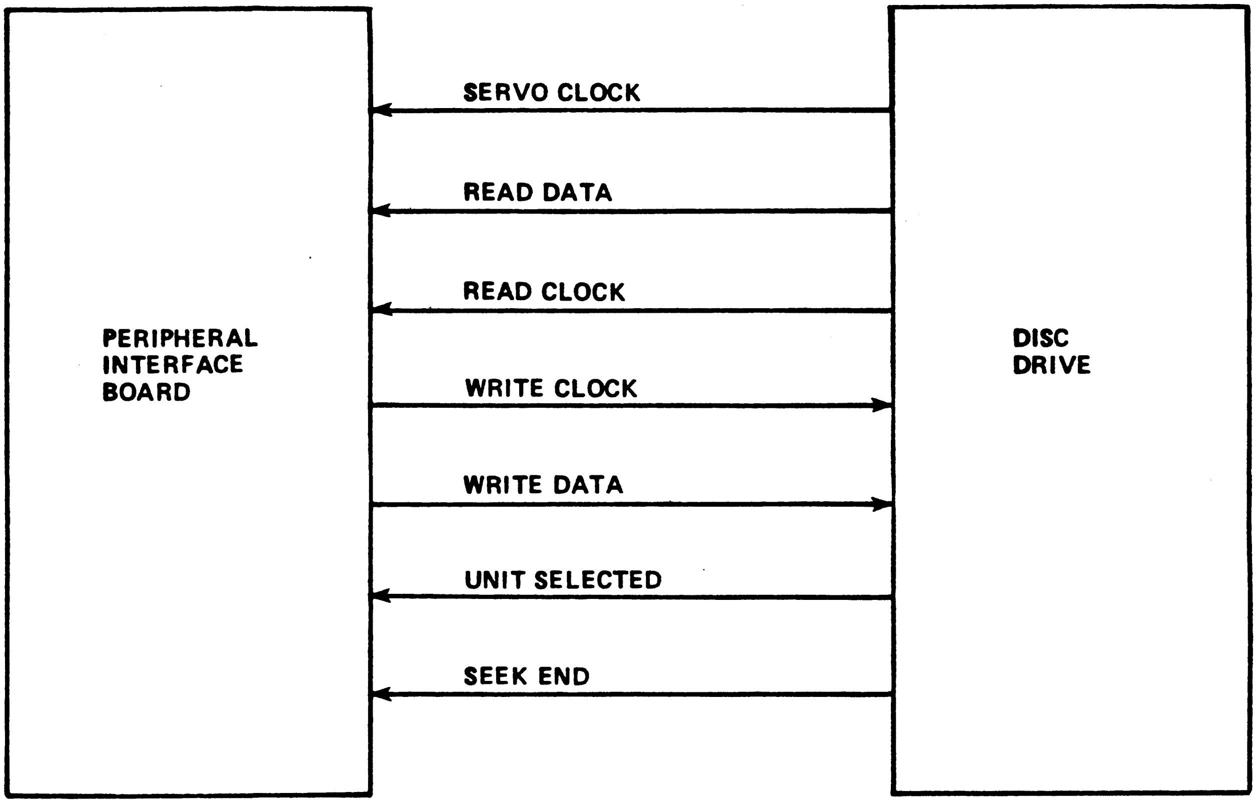


Figure 3-3. PIB/Disc Drive Differential Synchronization Cable Signals

TABLE 3-2. DRIVE SYNCHRONIZATION CABLE SIGNALS

Pin	Signal	Direction	Description
1	GND	-	Ground
2,14	SERVO CLK	In	Used to synchronize write data.
3,16	READ DATA	In	Data from drive to controller used on a read operation.
4	GND	-	Ground
5,17	READ CLK	In	Signals the beginning of a data cell.
6,19	WRITE CLK	Out	Transmits the write clock signal which is used to synchronize the write data.
7	GND	-	Ground
8,20	WRITE DATA	Out	Transmits data from the controller to the drive.
9,22	UNIT SELD	In	Disc drive unit has been selected.
10,23	SEEK END	In	The logical OR of on cylinder and seek error indicating that a seek operation has ended.
11	GND	-	Ground
12,24	-	-	Reserved
13,26	-	-	Reserved
15	GND	-	Ground
18	GND	-	Ground
21	GND	-	Ground
25	GND	-	Ground

3.2 PROGRAMMED INPUT/OUTPUT

The following subsections provide information on disc programmed I/O, detailing instruction formats and accumulator bit functions.

3.2.1 PROGRAMMED I/O INPUT

The DIA instruction is used to read the status of the disc drive. Software can use the status to detect the following:

- Unit Ready
- Drive Fault
- Write Protect
- On Cylinder
- Drive Select
- Seek Error

The instruction format is:

```
    DIA ac,50
```

where ac is the processor accumulator into which the status will be read and 50 is the device code. The format of the status word is shown in Figure 3-4.

Table 3-3 defines the function of each bit in the accumulator.

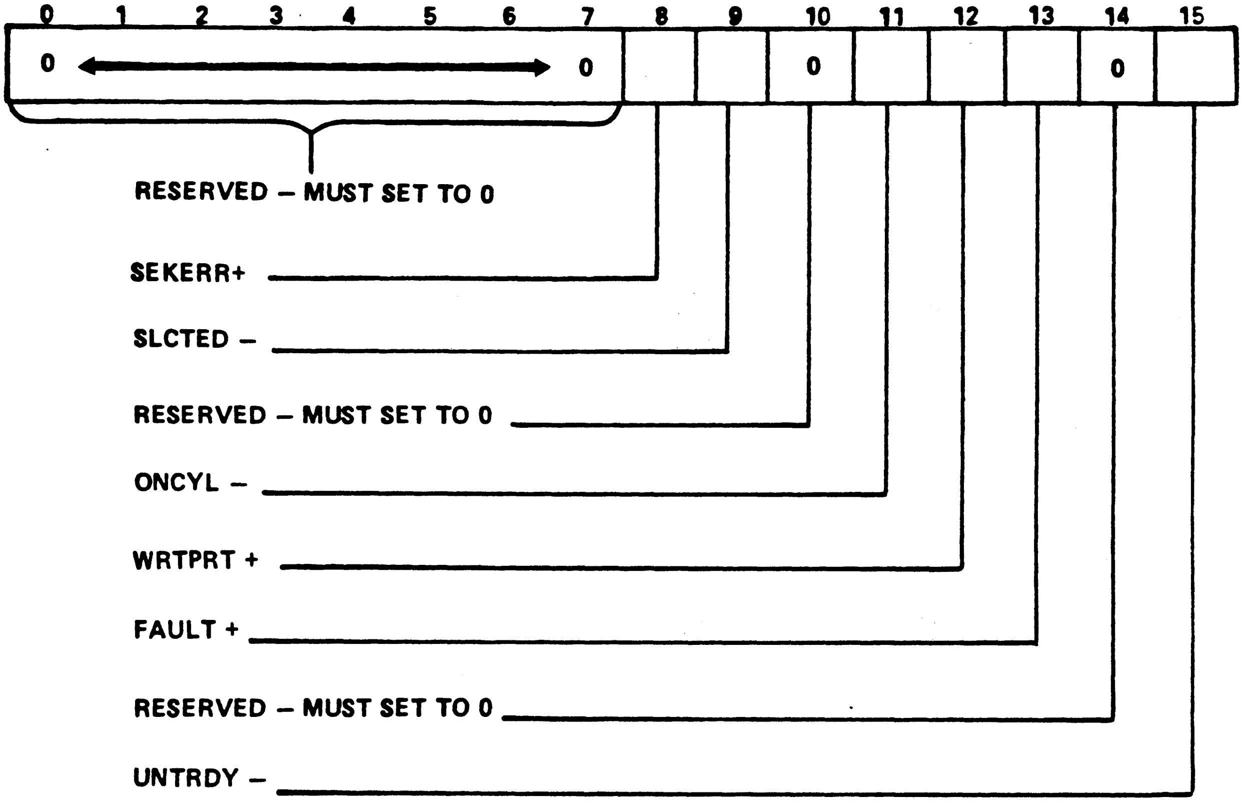


Figure 3-4. Drive Status

TABLE 3-3. DRIVE STATUS DEFINITIONS

Bits	Name	Function
0-7		Reserved, set to 0
8	SEKERR+	Seek Error - When set to 1, indicates a seek error has occurred. A seek error can result from the following: ● A seek could not be completed within 500 milliseconds ● An illegal track address was received by the drive ● The carriage has moved outside the recording field A seek error can only be cleared by issuing a return-to-zero command. See Section 3.4.3.
9	SLCTED-	Selected - When set to 0, indicates the drive has been selected. When set to 1, indicates the drive has not been selected.
10		Reserved, set to 0
11	ONCYL-	On Cylinder - When set to 0, indicates the heads are on cylinder. When set to 1, indicates the heads are not on cylinder.
12	WRTPRT+	Write Protect - When set to 1, indicates the selected drive may only be read (i.e., the drive is write protected). When set to 0, indicates the drive may be written to and read from.
13	FAULT+	Fault - When set to 1, indicates a fault in the selected drive has occurred. A fault may be caused by one or more of the following: ● Power fail ● Illegal head select ● Write fault ● Writing or reading while off cylinder ● Write and read gate on simultaneously

TABLE 3-3. DRIVE STATUS DEFINITIONS (Cont)

Bits	Name	Function
14		Reserved, set to 0
15	UNTRDY-	Unit Ready - When set to 0, the unit selected is ready. When set to 1, the unit selected is not installed, is not up to speed, has heads which are not engaged, or a drive fault exists.

3.2.2 PROGRAMMED I/O OUTPUT

DOA instructions utilizing device codes 50 and 52 are used to specify the disc drive unit output pulses.

3.2.2.1 DOA With Device Code 50 (Drive Unit Number)

The DOA instruction with a device code of 50 is used to output the unit number of the drive to be selected. The contents of the accumulator should contain the unit number found in word 1 of the IOCB (see Section 3.3.2). The format of the instruction is:

```
    DOA ac,50
```

where ac is the processor accumulator containing the unit number and 50 is the device code.

3.2.2.2 DOA With Device Code 52 (Tag Pulses)

The DOA instruction with a device code of 52 is used to output the different tag pulses. Tag pulses are sent to the drive to indicate that there is valid data in either the bus register, unit register, or both. The format of the instruction is:

```
    DOA ac,52
```

The contents of the accumulator should be arranged as indicated in Figure 3-5.

NOTE

> Tag 1, 2 or 3 may be used one at a time. It is illegal to have more than one tag signal active at the same time.

Table 3-4 gives a definition of the function of each bit in the accumulator which represent the different tag pulses.

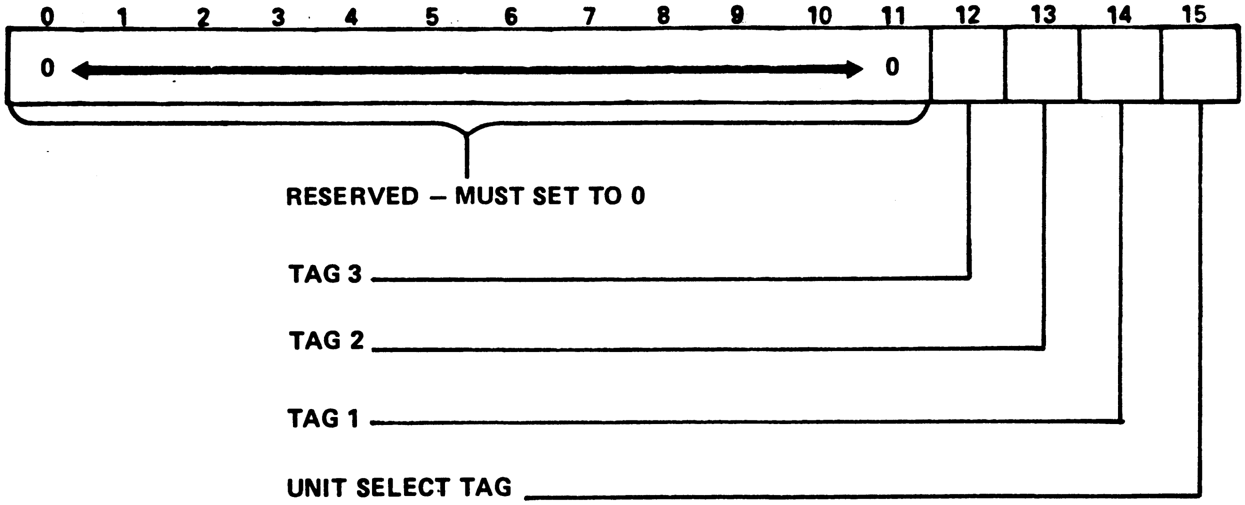


Figure 3-5. Contents of DOA Instruction Accumulator

TABLE 3-4. DOA INSTRUCTION ACCUMULATOR BITS

Bit	Function
0-11	Reserved, set to 0
12	Tag 3 - When set to 1, indicates to the drive that the bus register contains valid drive control information.
13	Tag 2 - When set to 1, indicates to the drive that the bus register contains a valid volume/head address.
14	Tag 1 - When set to 1, indicates to the drive that the bus register contains a valid cylinder address.
15	Unit Select Tag - When set to 1, indicates to the drive that there is valid data in the unit-select register. The unit-select tag must be on at The same time as a tag 1, 2, or 3. The tag signal must be held high through the entire operation.

3.2.2.3 DOA With Device Code 51 (Bus Register Loading)

The DOA instruction with a device code of 51 is used to load the bus register with the cylinder number. The format of the instruction is:

```
DOA ac,51
```

The meaning of the data bus register is determined by the selection of a tag 1, tag 2 or tag 3. If tag 1 is selected, the bus register contains cylinder information. If tag 2 is selected, the register contains head and volume information. If tag 3 is selected, the register contains control information to the disc controller, as defined in Table 3-1.

The instruction formats are defined and illustrated in the following subsections.

3.2.2.3.1 TAG 1

The instruction format for the cylinder address is shown in Figure 3-6.

Table 3-5 defines data bus bit functions with tag 1.

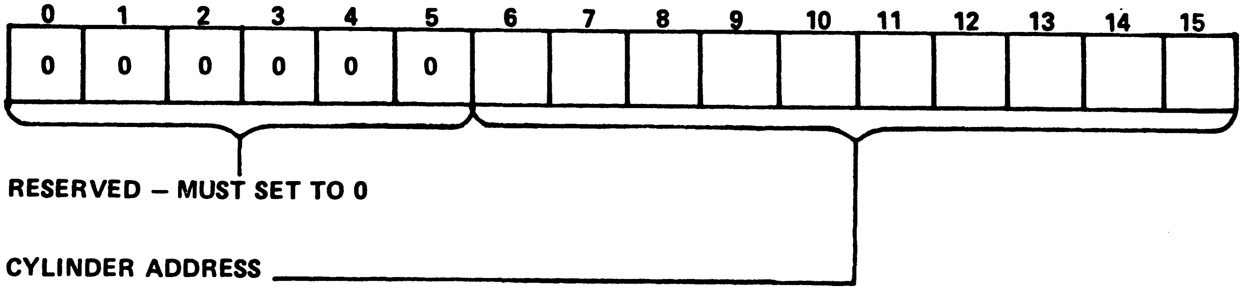


Figure 3-6. Tag 1 Instruction Format

TABLE 3-5. DATA BUS BIT FUNCTIONS WITH TAG 1

Bit	Function
0-5	Reserved, must be set to zero
6-15	Cylinder address (10 bits)

3.2.2.3.2 TAG 2

The instruction format for the cylinder address is shown in Figure 3-7. Note that the volume bit (bit 11) is for CMD drives only. For SMD drives, bit 11 is the most significant head bit.

Table 3-6 defines data bus bit functions with tag 2.

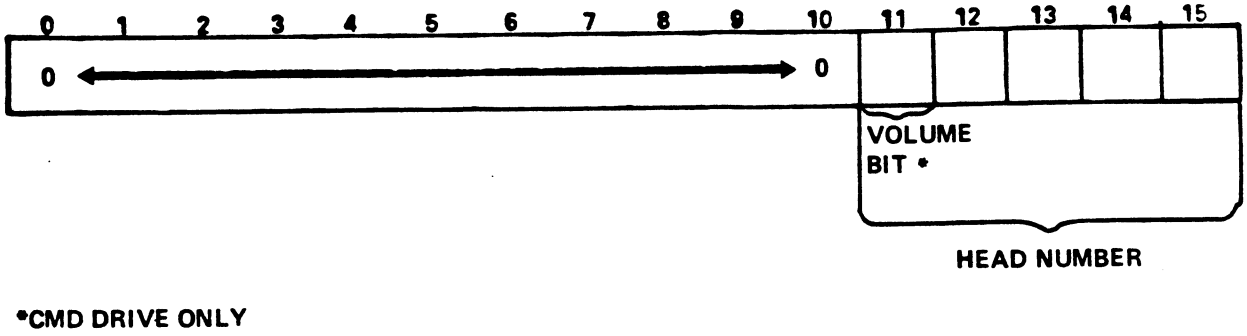


Figure 3-7. Tag 2 Instruction Format

TABLE 3-6. DATA BUS BIT FUNCTIONS WITH TAG 2

Bit	Function
0-10	Not defined; must be set to zero
11	Volume (CMD drives only); most significant head bit (SMD drives).
12-15	Head number

3.2.2.3.3 TAG 3

The instruction format which allows the drive to perform functions other than directly transferring data is shown in Figure 3-8.

Table 3-7 defines data bus bit functions with tag 3.

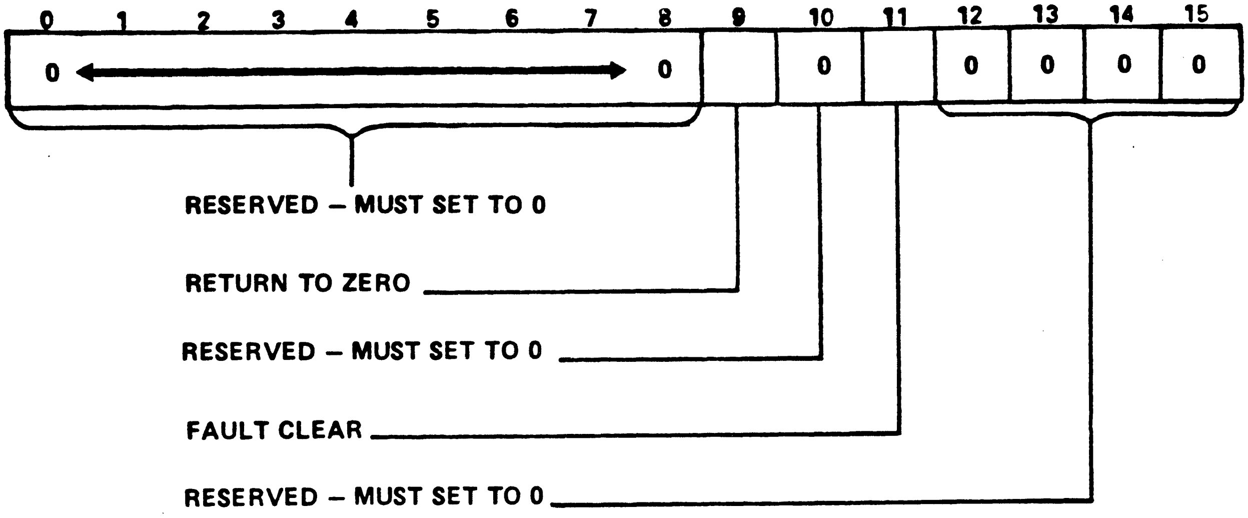


Figure 3-8. Tag 3 Instruction Format

TABLE 3-7. DATA BUS BIT FUNCTIONS WITH TAG 3

Bit	Function
0-8	Reserved; must be set to zero
9	Return to Zero - When this bit is set to 1, the disc will seek to track 0, reset the head register, select the cartridge volume (CMD only) and reset the seek-error bit in the termination-status word of the IOCB. (See Section 3.3.8).
10	Reserved; must be set to zero
11	Fault Clear - If the cause of a drive fault no longer exists, the fault is cleared and the drive is ready for operation. If the fault still exists, no action is taken.
12-15	Reserved; must be set to zero

3.3 INPUT/OUTPUT CONTROL BLOCK (IOCB)

The input/output control block (IOCB) for the disc controller consists of eight words located in main memory and reserved for holding information about a disc transfer. The eight words of the IOCB are loaded with information by the software before the operation begins.

Upon completion of an operation (successful or not), the controller writes status information about the operation into the termination status word. Figure 3-9 shows the format of the disc IOCB.

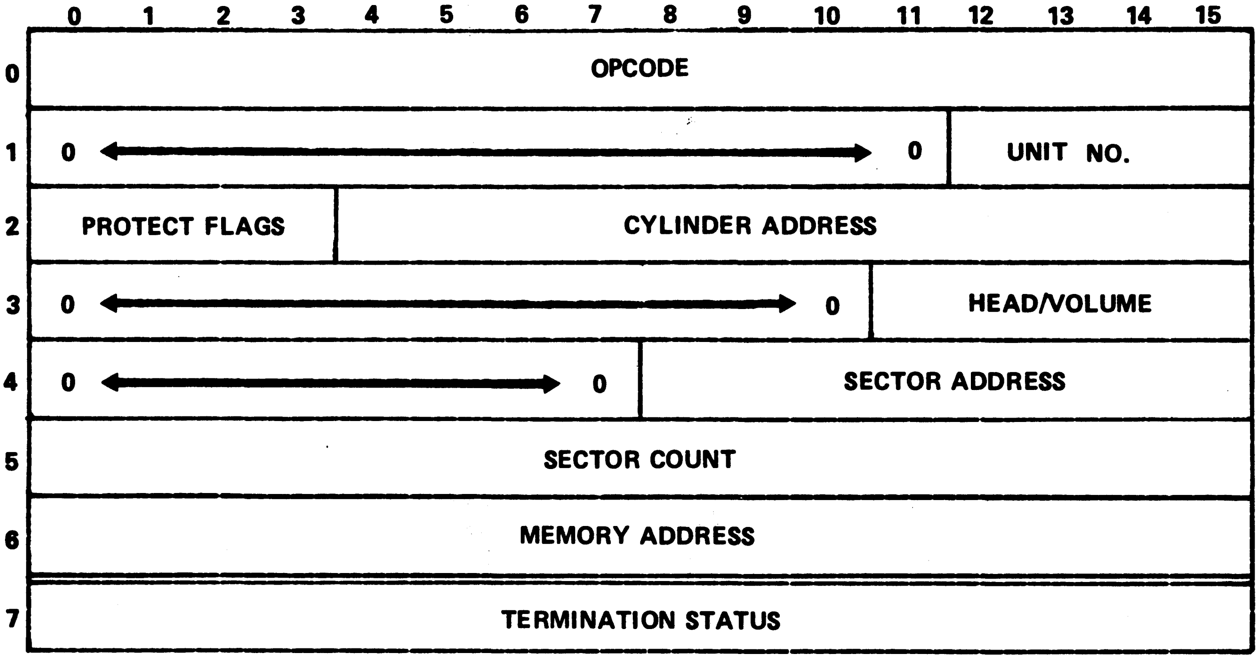


Figure 3-9. Disc IOCB Format

3.3.1 OPCODE (WORD 0)

The opcode word (word 0 of the disc IOCB) is used to define five operations which the disc may perform. The format of word 0 of the disc IOCB is shown in Figure 3-10.

Table 3-8 defines the bit usage in word 0.

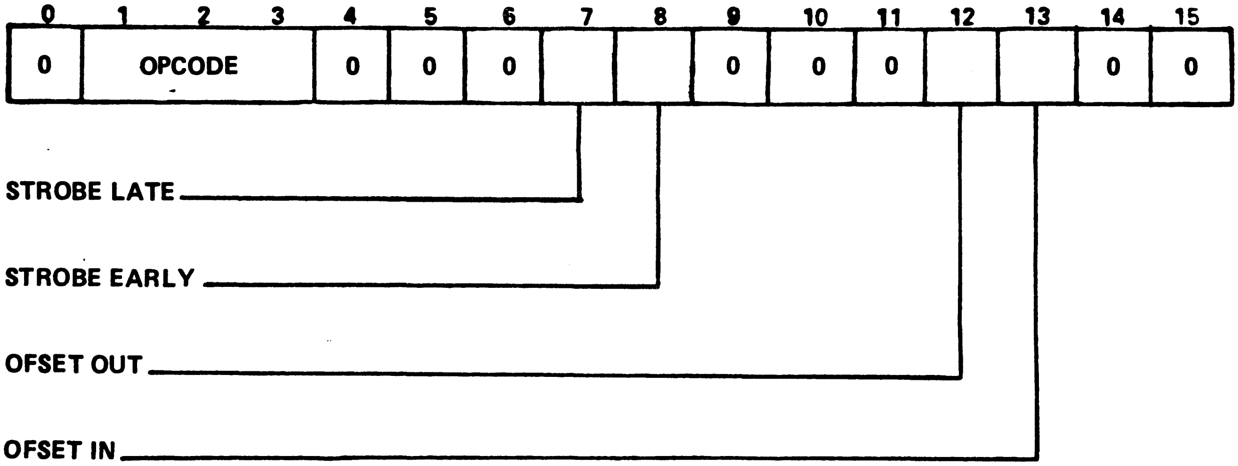


Figure 3-10. Opcode Word Format

TABLE 3-8. DISC IOCB WORD 0

Bits	Definition
0	Reserved - must be set to 0
1-3	Opcode - defines the following operations depending on their octal setting: 6 = Format 5 = Read Regardless 2 = Write Data 1 = Read Data 0 = Read Verify
4-6	Reserved - must be set to 0
7	Set to 1 for strobe late
8	Set to 1 for strobe early
9-11	Reserved - must be set to 0
12	Set to 1 for OFSET out
13	Set to 1 for OFSET in
14-15	Reserved - must be set to 0 **NOTE** Unused opcodes are reserved and should not be used.

3.3.2 UNIT NUMBER (WORD 1)

The unit-number word (word 1 of the disc IOCB) is used to specify the drive number on which to perform the operation. The format of word 1 of the disc IOCB is shown in Figure 3-11.

Table 3-9 defines the bit usage in word 1.

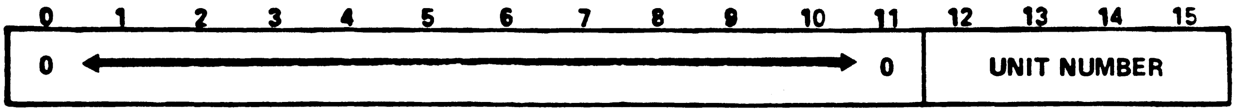


Figure 3-11. Unit Number Word Format

TABLE 3-9. DISC IOCB WORD 1

Bits	Definition
0-11	Reserved - must be set to zero
12-15	Represents the binary equivalent of the drive number

3.3.3 CYLINDER ADDRESS AND PROTECT FLAGS (WORD 2)

The cylinder-address word (word 2 of the disc IOCB) is used to specify the cylinder address for each operation. The controller uses this word to compare with the current cylinder address in the sector address header for address verification. The format of word 2 of the disc IOCB is shown in Figure 3-12.

Table 3-10 defines the bit usage in word 2.

If the file protect flags were set for this cylinder by the software at format time, they must also be set in the IOCB cylinder select word (see Sector Block Format Commands, Section 3.5).

All sixteen bits of the cylinder-select word are compared with the current cylinder address found in the sector header. If there is no match, a no-ID-compare error will occur and the operation will not be performed.

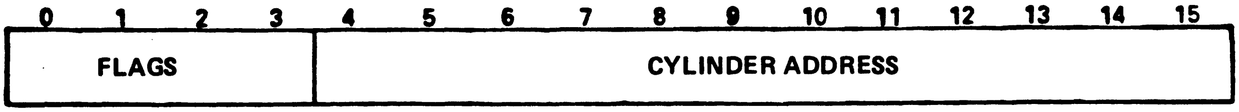


Figure 3-12. Cylinder Address Word Format

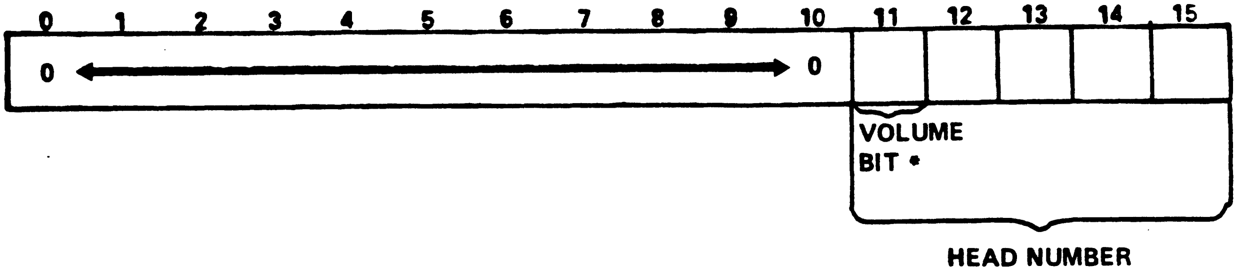
TABLE 3-10. DISC IOCB WORD 2

Bits	Definition
0-3	File protect flags
4-15	Cylinder address (12 bits)

3.3.4 HEAD/VOLUME (WORD 3)

The head/volume word (word 3 of the disc IOCB) is used by the controller during sector verification. The format of word 3 of the disc IOCB is shown in Figure 3-13.

Table 3-11 defines the bit usage in word 3. It further defines head/volume number bit usage for CMD and SMD drives.



*CMD DRIVE ONLY

Figure 3-13. Head/Volume Word Format

TABLE 3-11. DISC IOCB WORD 3

Bits	Definition
0-10	Reserved - must be set to 0
11-15	Used for head/volume number as follows: Bits — Drive 11*12 13 14 15 — CMD Drives 0 0 0 0 0 — Head number 0; Removable 1 0 0 0 0 — Head number 0; Fixed 1 0 0 0 1 — Head number 1; Fixed 11 12 13 14 15 — SMD Drives 0 0 0 0 0 — Head number 0 0 0 0 0 1 — Head number 1 0 0 0 1 0 — Head number 2
*Bit 11 = volume bit - used by CMD drives only.	

3.3.5 SECTOR ADDRESS (WORD 4)

The sector-address word (word 4 of the disc IOCB) is used to specify the sector address to be used by this operation. The controller compares the sector address word with the current sector address in the sector header. If the sector address does not match, the sector will not be accessed by the controller. The format of word 4 of the disc IOCB is shown in Figure 3-14.

Table 3-12 defines the bit usage in word 4.

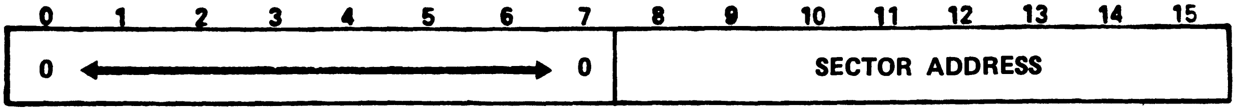


Figure 3-14. Sector Address Word Format

TABLE 3-12. DISC IOCB WORD 4

Bits	Definition
0-7	Reserved - must be set to 0
8-15	Specifies the 8-bit sector address

3.3.6 SECTOR COUNT (WORD 5)

The sector-count word (word 5 of the disc IOCB) is used only for read-regardless and format (see Section 3.5.3) operations. It allows a specific number of sectors to be included in an operation. The format for word 5 of the disc IOCB is shown in Figure 3-15.

Table 3-13 defines the bit usage in word 5.

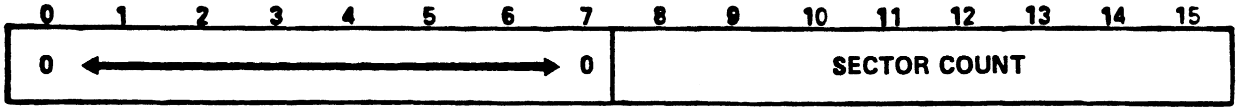


Figure 3-15. Sector Count Word Format

TABLE 3-13. DISC IOCB WORD 5

Bits	Definition
0-7	Reserved - must be set to 0
8-15	Specifies the binary equivalent of the number of sectors to be transferred in the operation. The sector count must never be greater than the sector capacity of one track. If it is, an error will occur.

3.3.7 MEMORY ADDRESS (WORD 6)

The 16-bit memory address word (word 6 of the disc IOCB) is used to specify the starting address for the data buffer. The format for word 6 of the disc IOCB is shown in Figure 3-16.

Table 3-14 defines the bit usage in word 6.

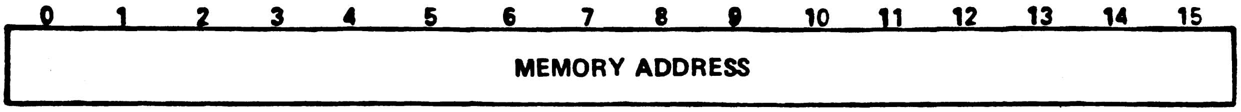


Figure 3-16. Memory Address Word Format

TABLE 3-14. DISC IOCB WORD 6

Bits	Definition
0-15	Starting address for the memory buffer.

3.3.8 TERMINATION STATUS (WORD 7)

Upon completion of an operation, the controller writes a termination status into the last word of the IOCB. The termination status provides information on the result of the operation, controller status, and drive status to the processor. Disc-related status information refers only to the drive selected by the unit select word in the IOCB involved. The format of word 7 of the disc IOCB is shown in Figure 3-17.

Table 3-15 defines the bit usage in word 7.

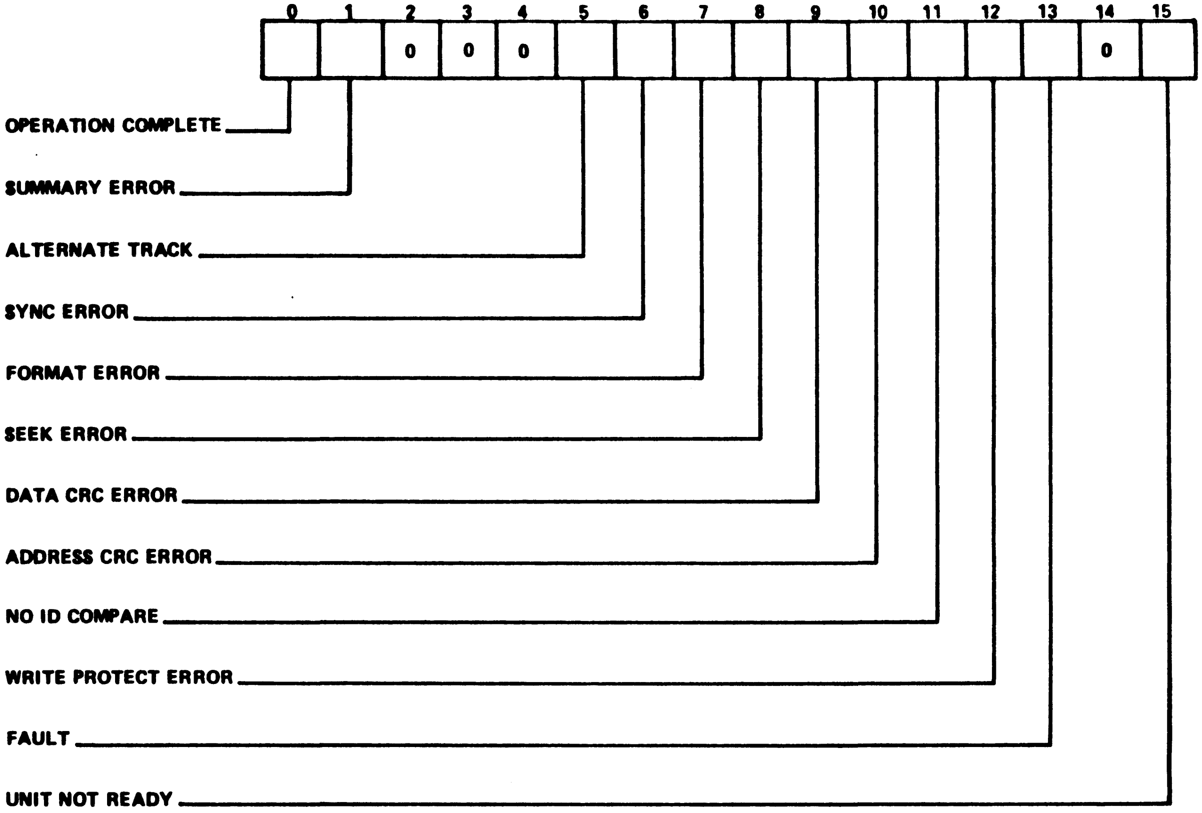


Figure 3-17. Termination Status Word Format

TABLE 3-15. DISC TERMINATION STATUS WORD 7

Bit	Definition
0	Operation Complete - Set to 1 upon completion of an operation, successful or not. It provides a means for software to determine when a valid termination status word has been written. The operation-complete bit must be cleared by software. For an error-free operation, this will be the only bit set in the termination status word.
1	Summary Error - Indicates that one or more of the controller error bits is set.
2-4	Reserved - set to 0 by PIB.
5	Alternate Track - Set to 1 when the controller detects a nonzero value in the next cylinder address of the sector address header, during a sector verification. The controller overwrites the cylinder and head number words in the IOCB with the value found in the sector header next-cylinder and next-head fields. The controller also sets the alternate-track bit to 1, the operation-complete bit to 1, and the summary-error bit to 1. The pending-interrupt bit in the disc IOCB pointer is set to 1 and an interrupt generated, if interrupts are enabled. When software detects the alternate-track bit set, a new seek should be issued based on the new information in the IOCB.
6	Synchronization Error - Indicates that the controller failed to synchronize with data (could not find any sectors on the track).
7	Format Error - Indicates that during a format, a track overrun occurred.
8	Seek Error - Indicates that a seek error occurred on the selected drive.

TABLE 3-15. DISC TERMINATION STATUS WORD 7 (Cont)

Bit	Definition
9	Data CRC Error - Indicates that the CRC code did not match with that obtained while reading the data field.
10	Address CRC Error - Indicates that the CRC code did not match with that obtained while reading the sector address header which the controller was trying to access.
11	No ID Compare - Indicates that the controller could not verify the sector address it was attempting to access. In effect, this means that there was no sector address match with any of the sectors on that track.
12	Write Protect Error - Indicates that a write or format operation was attempted on a drive that was write protected.
13	Fault - Indicates that a fault has occurred in the selected drive. A drive fault means that one or more of the following has occurred: ● Power-fail ● Illegal head select ● Write fault ● Writing or reading while off cylinder ● Write and read gate on simultaneously
14	Reserved - must be set to 0
15	Unit Not Ready - Indicates that the unit selected in the IOCB word 1 is not installed, is not up to speed, the heads are not engaged, or a drive fault exists.

3.3.9 DISC CONTROLLER INPUT CONTROL

The processor has a dedicated register that is reserved as a pointer to the IOCB for the disc. This register may be written into or read from by using the DOBC or DIBC instructions, respectively. The C control code is used to specify command instructions directed to the disc controller (rather than the multiplexer (S) or tape controller (P)). The format of the disc controller input control instruction is

```
    DIBC ac,77
```

where ac is the accumulator into which the value in the disc IOCB pointer is loaded.

The DIBC instruction may be used for two purposes:

1. Sensing the idle state - The idle state is indicated by a zero value in the accumulator following execution of a DIBC instruction to read the disc IOCB pointer.

2. Polling the interrupt-pending bit - When the disc controller has completed an operation, successful or not, it sets the interrupt-pending bit in the disc IOCB pointer. The interrupt-pending bit is the most significant bit in the disc IOCB pointer.

3.3.10 DISC CONTROLLER OUTPUT CONTROL

The DOBC instruction is used for activating and deactivating the disc controller. The format of the disc controller output control instruction is

```
DOBC ac,77
```

where ac is a general purpose accumulator. This instruction is used for two purposes:

1. Activating the disc controller - The disc controller is activated by a DOBC instruction in which the accumulator contains the 15-bit memory address of the first IOCB word. The most significant bit must be set to 0.

   NOTE

   > Location 0 is an illegal location for an IOCB starting address; 0 is defined as the idle state.

   When activated, the controller goes busy and starts the operation specified in the IOCB words.

2. Deactivating the disc controller - After the disc controller completes an operation and generates an interrupt, it must be deactivated by the software. A DOBC instruction with the accumulator set to zero deactivates the controller. This resets the interrupt-pending bit and places the disc controller in an idle state.

3.4 SEEK OPERATION

Software is responsible for initiating a seek, and ensuring its proper completion. Drive selection, seek initiation, and seek error recovery are described in the following subsections which include operational flowcharts. In these operations, tag pulse widths are of critical importance; a timing diagram is provided in Figure 3-18.

3.4.1 DRIVE SELECTION

To select a drive, issue a unit number (the DOA with the drive unit number) to device code 50. Then send a DOA to device code 52 with the accumulator containing 1 which outputs a unit select tag. Issue a DIA to verify the unit is selected and ready. If it is not selected, reissue the DIA until it is selected. This completes drive selection.

The drive select and seek initiation flowchart (Figure 3-19) illustrates this procedure.

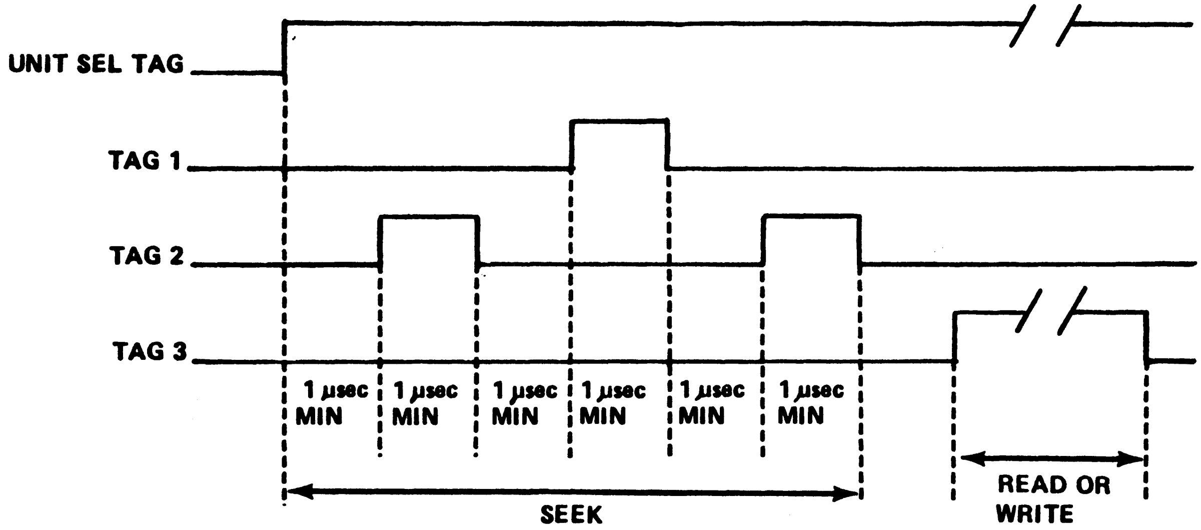


Figure 3-18. Seek Operation Timing

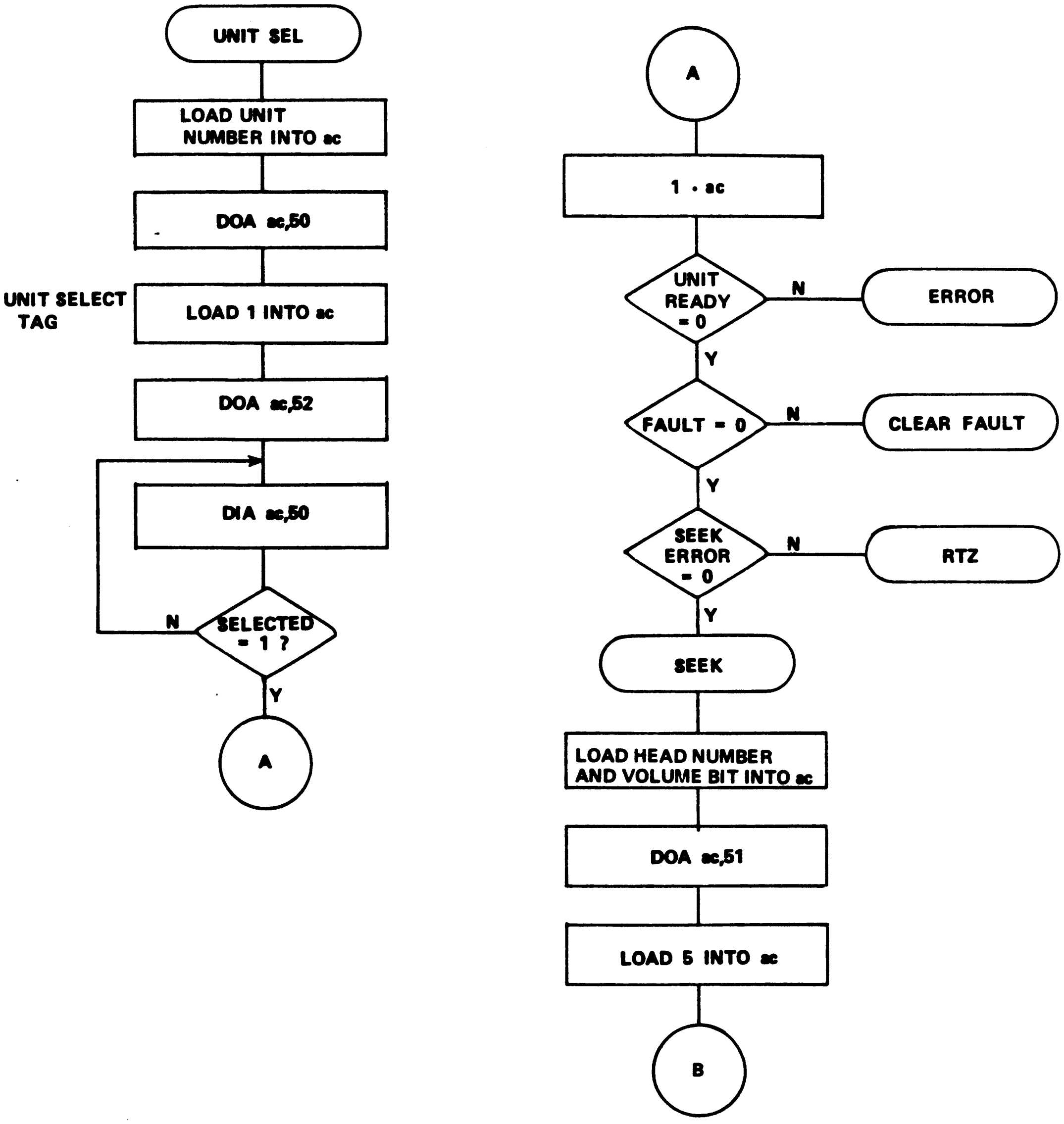


Figure 3-19. Drive Select and Seek Initiation (1 of 3)

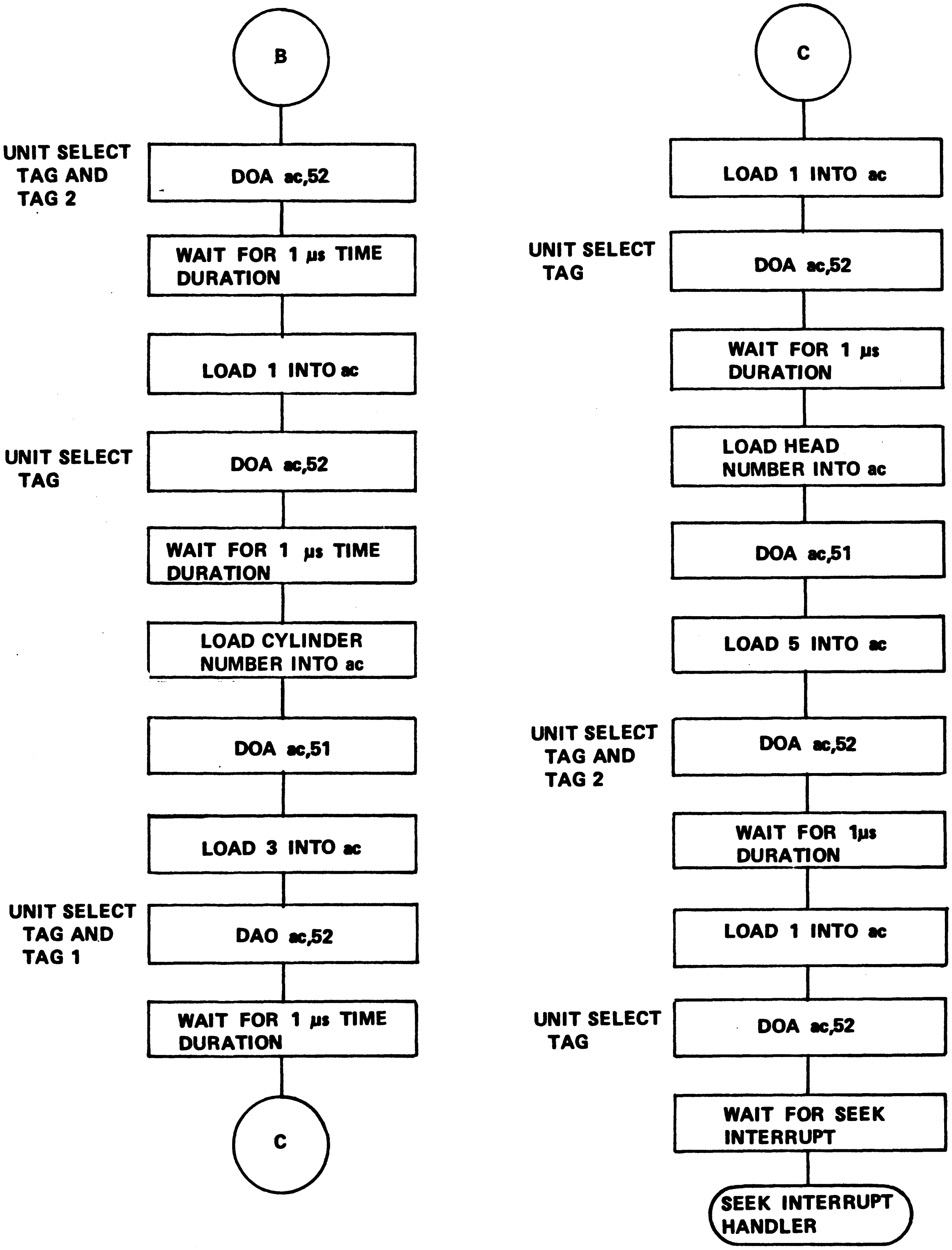


Figure 3-19. Drive Select and Seek Initiation (2 of 3)

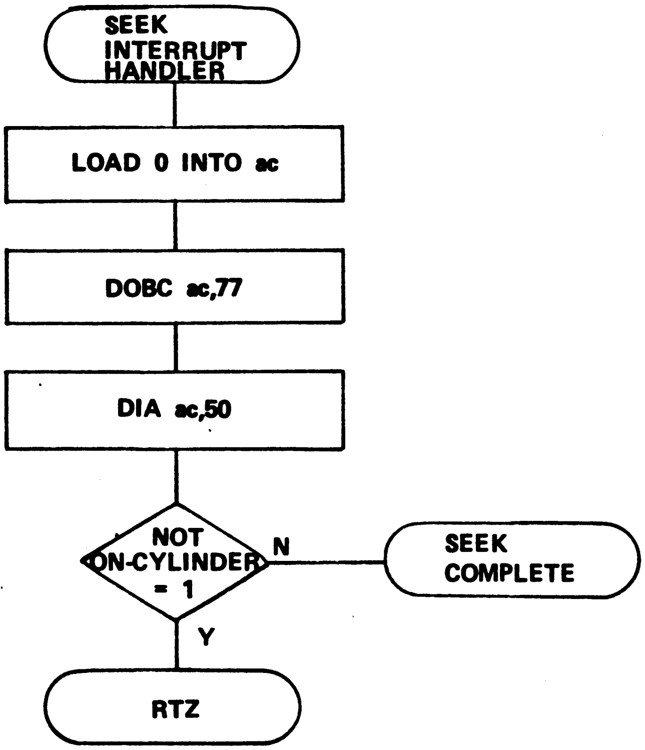


Figure 3-19. Drive Select and Seek Initiation (3 of 3)

3.4.2 SEEK INITIATION

The proper tag sequence to initiate a seek operation is tag 2, tag 1, tag 2. Note that when issuing a tag 1, 2 or 3, the unit select tag must be on.

The first tag 2 contains the head bits and/or volume bit. The volume bit must be issued in this tag 2 if utilizing a CMD drive.

The tag 1 issues the cylinder number. If using a CMD drive, the volume change takes effect, and initiates head movement.

The second tag 2 loads the head and is required for CMD drives. Setting the volume bit is not mandatory.

The seek is initiated.

On the first status read of the disc drive with the DIA instruction (Section 3.2), bit 11 (ONCYL- bit) is normally 1. The heads are in movement and not on cylinder. If the seek procedure has been successful, the ONCYL- goes to 0 and the seek error (bit 8) is also at 0.

A drive select and seek initiation flowchart is shown in Figure 3-19.

3.4.3 SEEK ERROR RECOVERY

A seek error occurs if one of the following conditions exist:

- The drive was unable to complete the seek within 500 milliseconds

- The carriage on the drive has moved outside the recording field

- The carriage has received an illegal track address

If on-cylinder (bit 11 = 0) and a seek error (bit 8 = 1) are indicated, the error must be cleared. To clear the seek error, issue a DOA to device code 51 with bit 9 (return-to-zero) set. Then issue a tag 3 to device code 52. Tag 3 must be kept low during the entire operation. Issue DIAs until the heads are on-cylinder. When they are on-cylinder, tag 3 can be released, and brought inactive. The seek is then reissued. If there is a drive fault (bit 13), issue a fault clear, tag 3. Note that tag 3 is kept low during the whole operation.

A seek error recovery flowchart is shown in Figure 3-20.

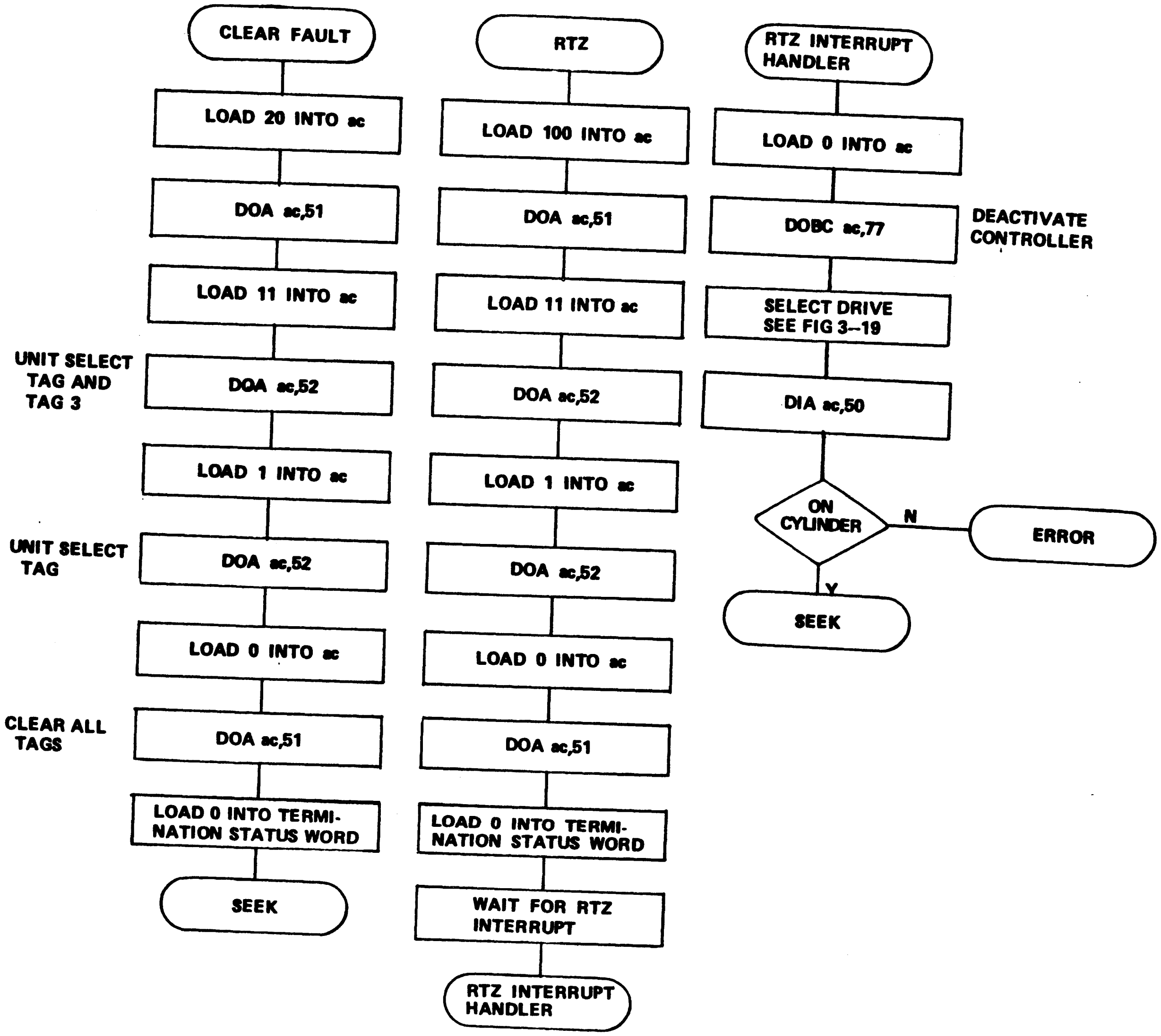


Figure 3-20. Seek Error Recovery Flowchart

3.5 SECTOR BLOCK FORMAT COMMANDS

This section describes disc sector block formatting, command functions, and I/O operations.

3.5.1 FORMATTING OPERATION

The disc is formatted under software control when a DOBC is issued with a format opcode (opcode 60000) specified in word 0 of the IOCB. The controller searches for the first index mark and begins reading address headers from processor memory. Each sector is written to the disc, allowing partitioning in the manner best suited for the particular application.

Before initiating a format operation, the software must build a sector table in memory. This table is used as a source of sector header information when formatting begins. The starting address for this table is located in word 6 of the disc IOCB. The sector table consists of a series of sector blocks, one for each sector to be formatted. All sector blocks must be stored in a contiguous area in processor memory, in the same sequential order that they are to be written to disc. The sector addresses, however, do not have to be in sequential order. Figure 3-21 shows the sector table as built in memory.

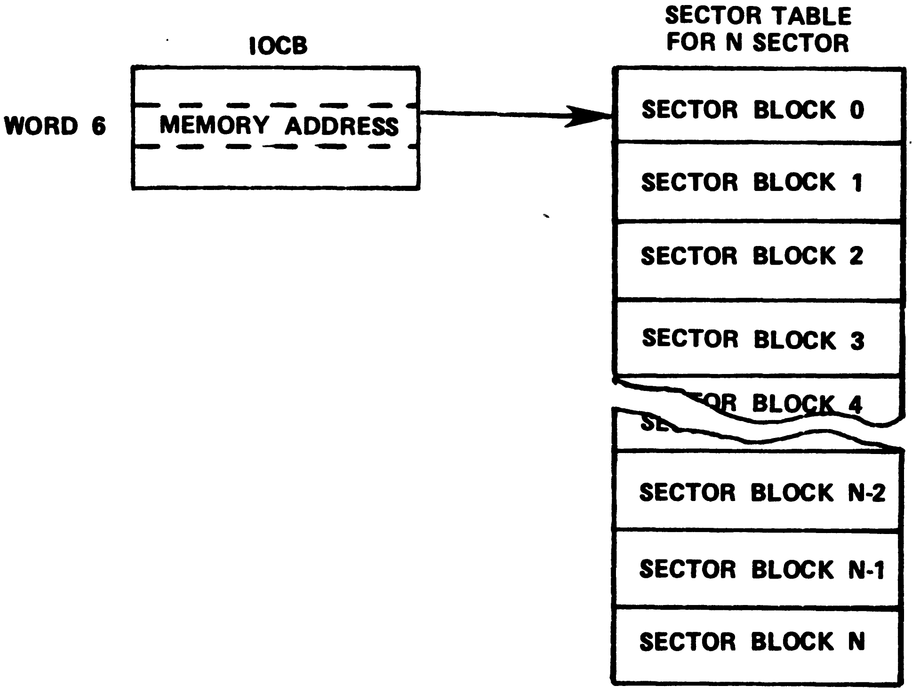


Figure 3-21. Sector Table in Memory

3.5.2 SECTOR BLOCK DESCRIPTION

A sector block is comprised of four words which contain the information to be written into the header for each sector.

3.5.2.1 Sector Block Format - Word 0 Functions

Word 0 is used to define the current cylinder address of the sector. Figure 3-22 shows the contents of word 0.

Table 3-16 defines the fields in word 0.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FLAGS				CURRENT CYLINDER											

Figure 3-22. Sector Block Format - Word 0 Contents

TABLE 3-16. SECTOR BLOCK FORMAT - WORD 0 FUNCTIONS

Bits	Function
0-3	Defines file-protect flags (see Section 3.3.3).
4-15	Defines the current cylinder address of the sector.

3.5.2.2 Sector Block Format - Word 1 Functions

Word 1 is used to define the current sector and head number of the sector. Figure 3-23 shows the contents of word 1.

Table 3-17 defines the fields in word 1.

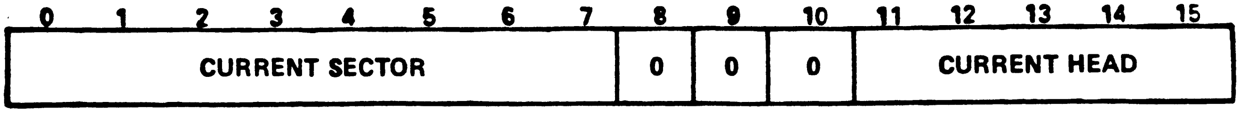


Figure 3-23. Sector Block Format - Word 1 Contents

TABLE 3-17. SECTOR BLOCK FORMAT - WORD 1 FUNCTIONS

Bits	Function
0-7	Defines the current sector address of the sector.
8-10	Reserved - must be set to 0.
11-15	Defines the head number of the sector.

3.5.2.3 Sector Block Format - Word 2 Functions

Word 2 is used to define the alternate-cylinder address of the sector. Figure 3-24 shows the contents of word 2.

Table 3-18 defines the fields in word 2.

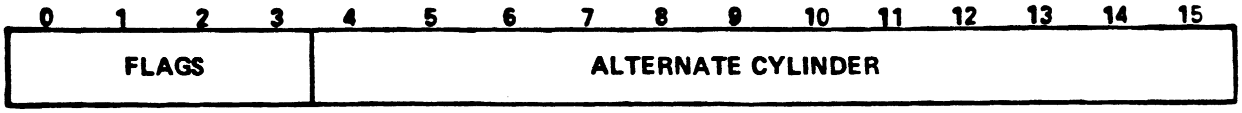


Figure 3-24. Sector Block Format - Word 2 Contents

TABLE 3-18. SECTOR BLOCK FORMAT - WORD 2 FUNCTIONS

Bits	Function
0-3	Defines file-protect flags (if alternate track is needed). Should be 0, unless there is a bad track.
4-15	Defines the alternate-cylinder address of the sector. When initially formatting a track, this word must be set to zero. If the track being formatted was bad, the software may update this word with an alternate-cylinder address by issuing another format command. A nonzero value in this word notifies the controller that this track is bad, sets the alternate-track bit to 1 in the termination status word, and terminates the operation. The alternate-cylinder word of each sector block in the sector table (every sector on the track) should contain the same value, since the entire track will be flagged as bad.

3.5.2.4 Sector Block Format - Word 3 Functions

Word 3 is reserved, and must be set to zero (0). Figure 3-25 shows the contents of word 3.

Table 3-19 defines the fields in word 3.

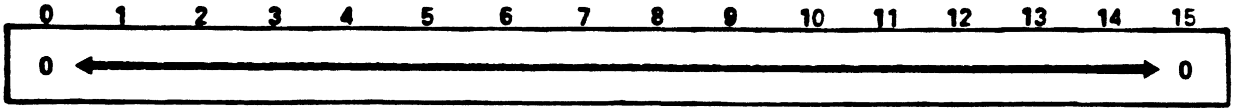


Figure 3-25. Sector Block Format - Word 3 Contents

TABLE 3-19. SECTOR BLOCK FORMAT - WORD 3 FUNCTIONS

Bits	Function
0-15	Reserved - must be set to zero (0).

3.5.3 FORMATTING PROCEDURE

The format command formats one track per operation. The software must ensure that the sector table contains all sectors to be written on a track. If fewer sectors are specified than are available on the track, the remaining sectors will be zeroed out. If more sectors are specified than are available, a format error will occur.

The software must also build the IOCB for the format operation. Figure 3-26 is an example of a format IOCB; Figure 3-27 shows the first sector block of the sector table. The IOCB cylinder and head select words must match the current cylinder and head number of each sector block. The IOCB memory address (word 6) must contain the address of the first word in the sector table. The IOCB sector count (word 5) must be set to the total number of sectors which will be formatted on the track.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	FORMAT OPCODE
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	UNIT 3
2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	CYLINDER 5
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	HEAD 0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SECTOR 0
5	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	SECTOR COUNT = 8
6	FIRST SECTOR BLOCK ADDRESS																
7	CONTROLLER STATUS																

Figure 3-26. Input/Output Control Block

	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	CURRENT CYLINDER = 5
CURRENT SECTOR = 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CURRENT HEAD = 0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ALTERNATE CYLINDER = 0
NEXT SECTOR = 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NEXT HEAD = 0

Figure 3-27. First Sector Block

When activated by a DOBC instruction and an index mark is recognized, the controller writes the sector information with the header contained in the first four words (first sector block) of the sector table. The controller writes the first sector address and the address header cyclic redundancy check (CRC), an algorithm used to detect errors in data transfer. Then the controller writes the data, followed by the data CRC. A worst-case pattern is written into the data field (66666). After the initial sector block, each sector mark reads the next entry and writes that onto the disc. The same procedure is followed for each sector.

The controller continues to write sectors until one of two conditions occurs:

1. The specified number of sectors has been formatted, i.e., the sector count is 0. In this case, the controller fills the remainder of the track with zeroes, sets the operation-complete bit in the termination-status word, sets the interrupt-pending bit and generates an interrupt, if interrupts are enabled.

2. All available sectors on the current track have been written, and the sector count indicates that there are still sectors to be written. In this case, the format operation aborts and a format error is reported in the termination-status word.

When the format operation is complete, the software should issue a read-verify for each sector. This will ensure that there are no CRC errors in the address fields or data fields.

When a header-CRC error is reported, an alternate cylinder must be selected. The alternate cylinder is written into the alternate-cylinder word in every location of the sector table, and the format command reissued. The alternate cylinder number is then written into all the current cylinder numbers and the alternate cylinder numbers in the sector table are zeroed out. A seek to the alternate cylinder is then issued. When the heads are on-cylinder, a format is issued.

When this procedure is completed, software returns the heads to the bad cylinder plus one (1) to continue the formatting operation.

It is also advisable for software to perform a read-regardless operation (see Section 3.5.4.4). This reads the header field, header CRC, the data field and data CRC into main memory. The sector count in the read-regardless operation should equal the number of sectors written to the track. After the read-regardless operation is performed, software may compare the address header information with the information in the sector table. This ensures that the address information was written to disc properly.

3.5.4 SECTOR BLOCK I/O OPERATIONS

I/O operations involve the various read and write operations as described in the following subsections.

3.5.4.1 Write Data Operation

A write-data operation causes the controller to transfer data from the processor memory buffer to the drive. When a DOBC instruction is issued with the write operation (opcode 20000) specified in word 0 of the IOCB, the controller responds by reading the sector address headers on the disc in search of a match to the address in words 2, 3 and 4 of the IOCB. When the controller finds a sector address header that compares with the IOCB address fields, it starts reading the data from the memory data buffer and writing it to the disc until 256 data words are written to the sector.

If there is no compare of the sector address and the address specified in the IOCB, the controller issues a no-ID-compare, a summary error and an operation-complete, and then generates an interrupt.

3.5.4.2 Read Data Operation

The read-data operation is used to read data from the disc and transfer it into the specified processor memory buffer. When a DOBC instruction is issued with the read operation (opcode 10000) specified in word 0 of the IOCB, the controller responds by searching for the sector address specified in words 2, 3 and 4 of the IOCB. When the controller finds a sector address header that compares with the IOCB address fields, it starts reading data from the disc and writing it to the memory buffer. After the sector is completely read (256 words), the CRC is checked. If the CRC is not correct, the data-CRC error bit will be set to 1 in the termination status word.

If there is no compare of the sector address and the address specified in the IOCB, the controller issues a no-ID-compare, a summary error and an operation-complete, and then generates an interrupt.

3.5.4.3 Read-Verify Operation

The read-verify operation is used to verify that data written on the disc during a write data or format operation was written without error. When a DOBC instruction is issued with the read-verify operation (opcode 00000) specified in word 0 of the IOCB, the controller responds by searching for the sector address specified in the IOCB. When the controller finds a sector address header that compares with the IOCB address fields, it reads the data in the sector without transferring it to memory. The data CRC is computed and compared with the CRC on disc and any error is reported in the termination status word.

If there is no compare of the sector address and the address specified in the IOCB, the controller issues a no-ID-compare, a summary error and an operation-complete, and then generates an interrupt.

3.5.4.4 Read-Regardless Operation

The read-regardless operation is used by the software for error recovery and the identification of bad tracks. When the disc controller is activated by a DOBC instruction with the read-regardless operation (opcode 50000) specified in word 0 of the IOCB, it starts transfer of data from the disc to the memory buffer upon detection of an index mark.

The number of sectors transferred is specified in the IOCB sector count. The maximum sector count should not be greater than the sector capacity of one track. If the count is less than the number specified, the sectors are transferred, beginning from the index mark. If the count is greater than the capacity of the track, the whole track will be transferred until index is seen again, and the operation terminates.

The largest read cycle that a read-regardless operation can perform is one track. No sector verification is involved and the controller transfers all 262 words of the sector including the address header (four words), the address CRC (one word), the data field (256 words) and the data CRC (one word).

Sector transfer is illustrated in Figure 3-28.

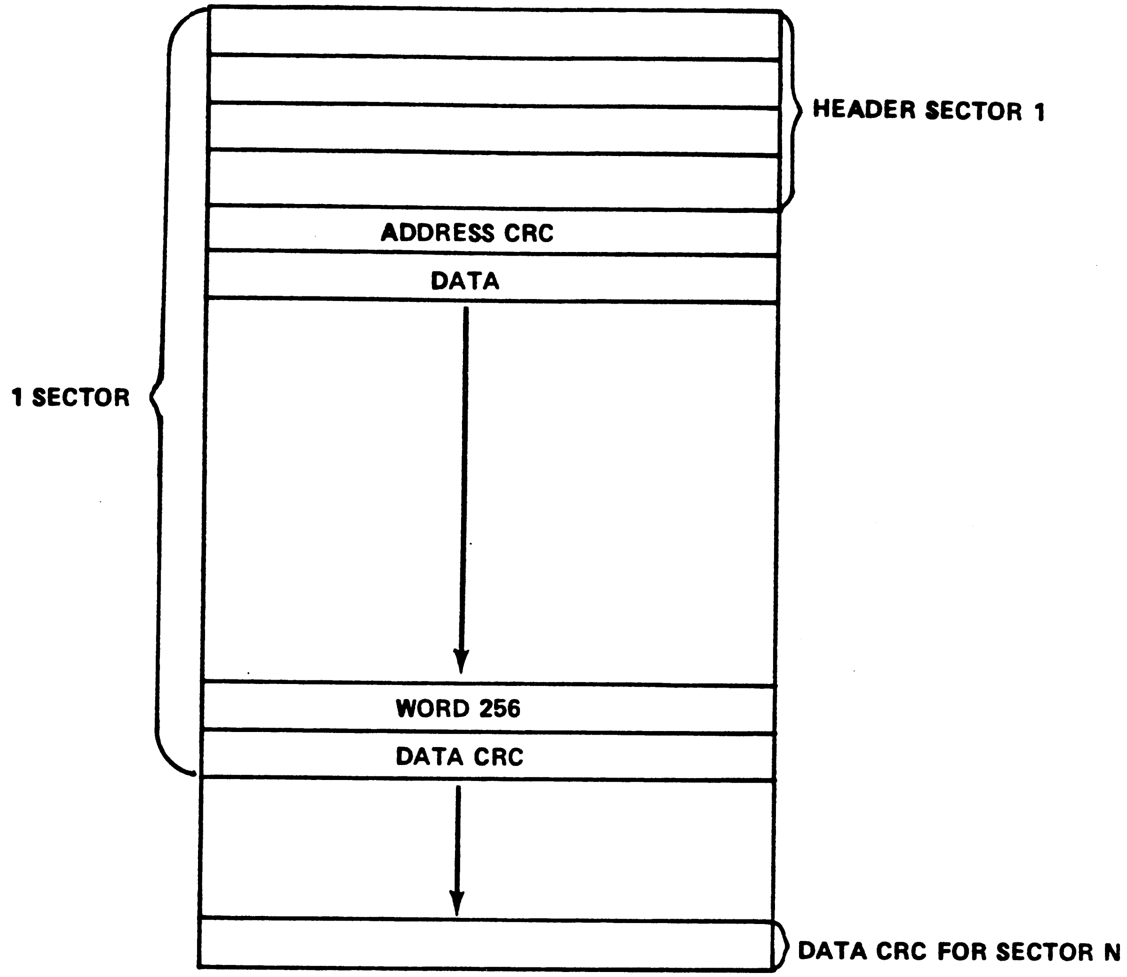


Figure 3-28. Read-Regardless Operation

3.6 INTERRUPT OPERATION

The disc controller generates an interrupt after completion of an operation whether successful or not. For a description of interrupt servicing, see Section 1.2.1.2 on Programming Polling and Interrupts.

Section 4
TAPE DRIVE INTERFACE

4.1 INTRODUCTION

The POINT 4 MARK 3 Streaming Tape Controller is designed to handle one Archive-compatible Intelligent Interface. One Intelligent Interface can handle four physical drives. The Tape Controller provides high speed, direct memory access between the tape drives and the POINT 4 MARK 3 CPU. A streaming tape system provides maximum tape utilization and high throughput for those applications which do not require individual records. One major application is backup for disc systems. To achieve maximum tape utilization and data throughput, a streaming-tape system incorporates very short inter-record gaps and constant high-speed tape motion.

4.1.1 FEATURES

The POINT 4 MARK 3 Tape Controller has the following features:

- Handles one Intelligent Interface which accommodates four Archive-compatible streaming tape drives
- DMA transfers at 90 kilobytes per second
- Accommodates tape speeds of 90 inches per second

4.1.2 PERFORMANCE CHARACTERISTICS

Tape Controller

Drives:
Four

Drive type:
Archive-compatible 1/4-inch streamers

DMA Transfer Rate:
90 kilobytes per second

Controller Device Codes:
60-62

Tape Port Assignments:
Port - J1

I/O Instructions:
Input - DIA
Output - DOA

DMA (Device Code 77):
DIBP
DOBP

Tape Drive

Track positioning:
500 milliseconds/track

Tape start:
300 milliseconds maximum for 90 inches per second
100 milliseconds maximum for 30 inches per second

Tape stop:
300 milliseconds maximum for 90 inches per second
100 milliseconds maximum for 30 inches per second

Unformatted capacity:
21.6 megabytes

Formatted capacity:
20 megabytes

Recording tracks:
4

4.2 TAPE CONTROLLER INTERFACE

Interface between the Tape Controller and the Intelligent Interface consists of:

- Two programmed I/O control lines
- Two programmed I/O status lines
- Two DMA control lines
- 8-bit bidirectional bus
- Bus-direction line
- Interface reset line

4.2.1 TAPE CONTROLLER INTERFACE SIGNALS

Tape controller interface signals are carried on 50-pin cable connector J1. Figure 4-1 illustrates these signals; their functions are defined in Table 4-1.

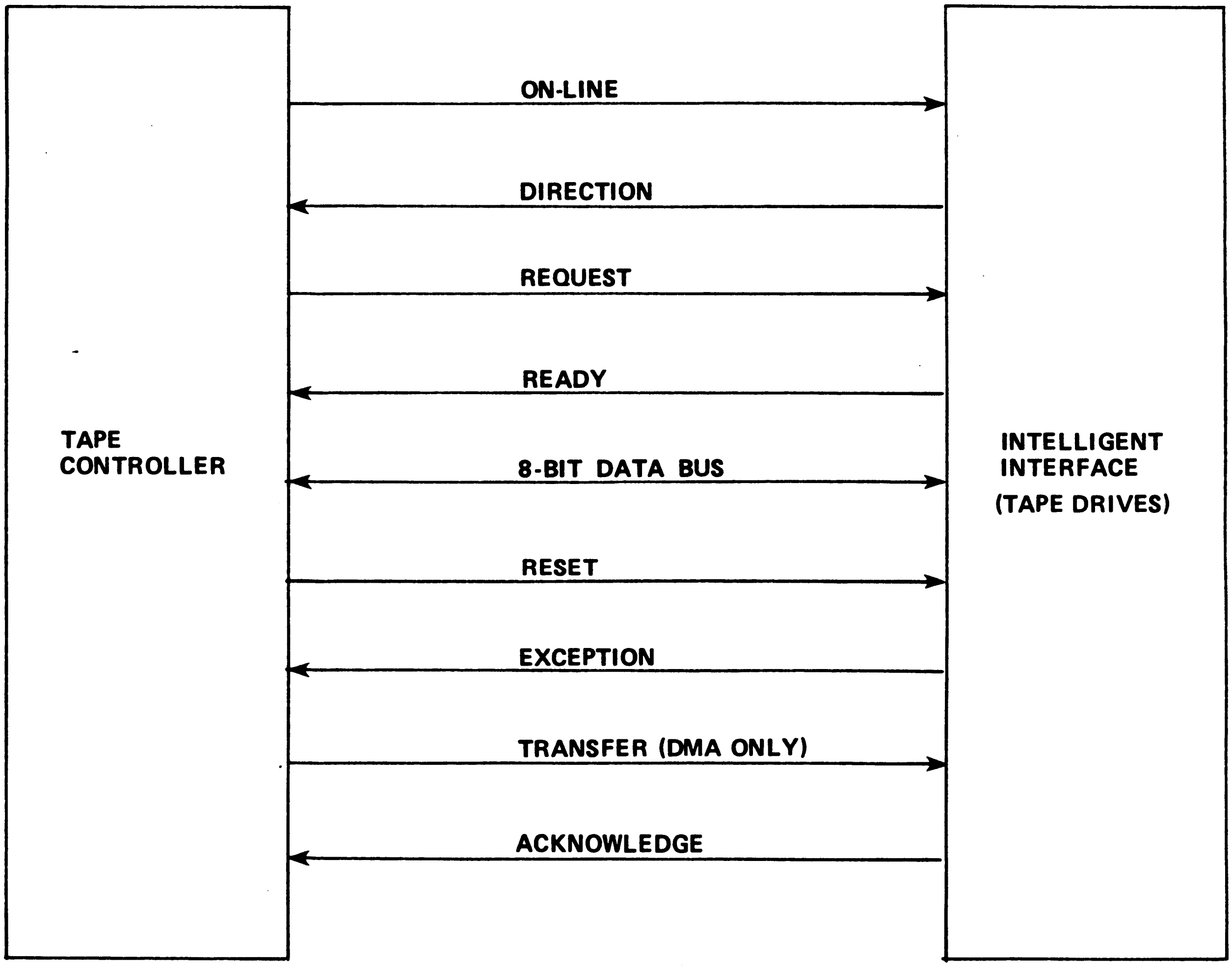


Figure 4-1. Tape Controller Interface Signals

TABLE 4-1. TAPE CONTROLLER INTERFACE CABLE J1 SIGNAL FUNCTIONS

Pin	Signal	Direc-tion	Description
1	-	-	Reserved
2	GND	-	Ground
3	ONL-	OUT	Tape Controller is on-line to Intelligent Interface
4	GND	-	Ground
5	REQ-	OUT	Request - Used by the Tape Controller to initiate command sequences
6	GND	-	Ground
7	-	-	Reserved
8	GND	-	Ground
9	REST-	OUT	Resets Intelligent Interface, aborting all operations
10	GND	-	Ground
11	EXC-	IN	Exception condition exists in Intelligent Interface/drive operation
12	GND	-	Ground
13	-	-	Reserved
14	-	-	Reserved
15	-	-	Reserved
16	GND	-	Ground
17	DIRC-	IN	Indicates the direction of data flow
18	GND	-	Ground
19	-	-	Reserved
20	GND	-	Ground
21	-	-	Reserved

TABLE 4-1. TAPE CONTROLLER INTERFACE CABLE J1 SIGNAL FUNCTIONS (Cont)

Pin	Signal	Direc-tion	Description
22	GND	-	Ground
23	RDY-	IN	Intelligent Interface ready for new operation
24	GND	-	Ground
25	-	-	Reserved
26	GND	-	Ground
27	DA0-	Bidirec	Data Bit 0
28	GND	-	Ground
29	DA1-	Bidirec	Data Bit 1
30	GND	-	Ground
31	DA2-	Bidirec	Data Bit 2
32	GND	-	Ground
33	DA3-	Bidirec	Data Bit 3
34	GND	-	Ground
35	DA4-	Bidirec	Data Bit 4
36	GND	-	Ground
37	DA5-	Bidirec	Data Bit 5
38	GND	-	Ground
39	DA6-	Bidirec	Data Bit 6
40	GND	-	Ground
41	DA7-	Bidirec	Data Bit 7
42	GND	-	Ground
43	-	-	Reserved
44	-	-	Reserved

TABLE 4-1. TAPE CONTROLLER INTERFACE CABLE J1 SIGNAL FUNCTIONS (Cont)

Pin	Signal	Direction	Description
45	XFER-	OUT	Used in control of data transfer
46	GND	-	Ground
47	ACK-	IN	Intelligent Interface acknowledges data transfer
48	GND	-	Ground
49	-	-	Reserved
50	-	-	Reserved

4.3 INPUT/OUTPUT CONTROL OPERATIONS

The following subsections provide information on tape programmed I/O, detailing instruction formats and accumulator bit functions.

4.3.1 PROGRAMMED I/O INPUT

The DIA instruction is used by software to read status information from the Tape Controller. Software can detect the following status information using the DIA instruction:

- An exception to normal operation
- Direction of data transfer
- Ready for command transfer

The instruction format is:

```
DIA ac,60
```

where ac is the processor accumulator to receive status information and 60 is the device code. The format of the accumulator is shown in Figure 4-2; bit functions are defined in Table 4-2.

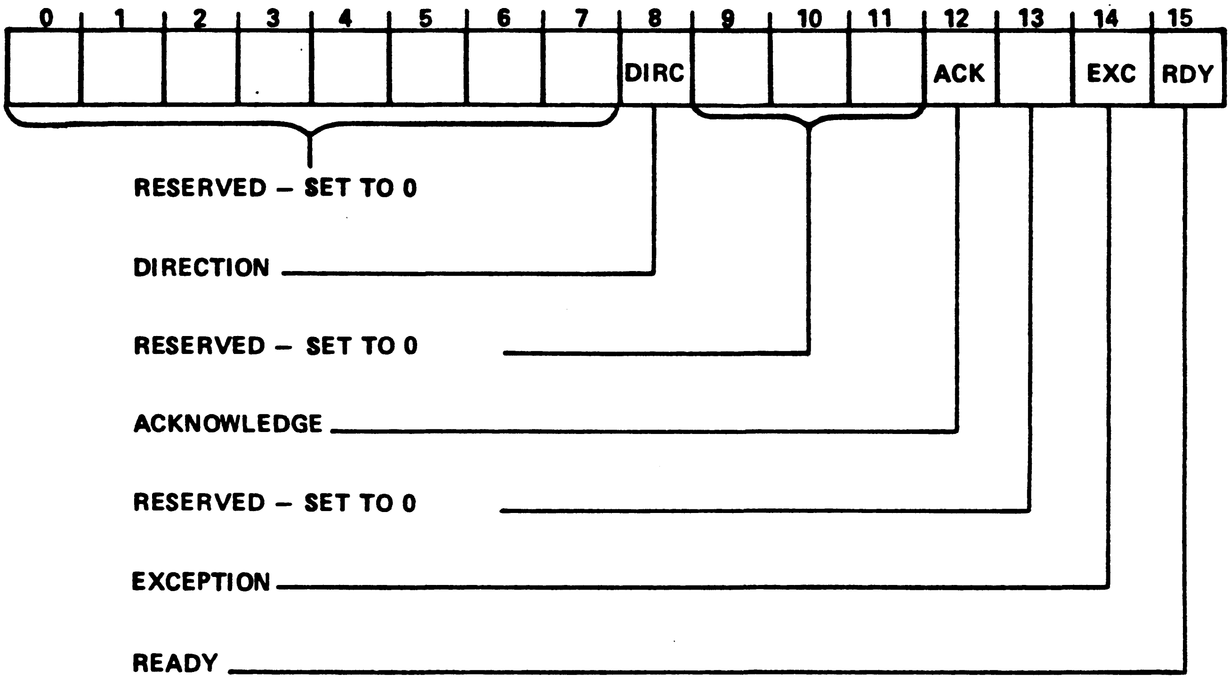


Figure 4-2. Format of DIA Instruction Accumulator

TABLE 4-2. DIA INSTRUCTION ACCUMULATOR BITS

Bits	Signal	Function
0-7	-	Reserved - set to 0
8	DIRC	Indicates the direction of data bus. Set to 1 for input; set to 0 for output.
9-11	-	Reserved - set to 0
12	ACK	The status signal supplied by the Intelligent Interface to indicate that it has read (for write operations) the byte from the data bus. For read operations, the Intelligent Interface indicates that a byte has been placed on the data bus to be read by the Tape Controller. **NOTE** This bit is used by the Tape Controller and need not concern the programmer.
13	-	Reserved - set to 0
14	EXC	The Intelligent Interface reports that an error condition exists, and that the previous command has been aborted. The Tape Controller issues a read-status command to determine the exact cause of the exception condition.

(Table continues on next page)

TABLE 4-2. DIA INSTRUCTION ACCUMULATOR BITS (Cont)

<table>
<tr><th>Bits</th><th>Signal</th><th colspan="2">Function</th></tr>
<tr><td>15</td><td>RDY</td><td colspan="2">The ready-status bit has different meanings depending on the operational mode. The following outlines the conditions indicated when this bit is set to one.</td></tr>
<tr><td></td><td></td><td><u>Mode</u></td><td><u>RDY Bit Function</u></td></tr>
<tr><td></td><td></td><td>Command Transfer</td><td>Command has been taken from the data bus by the Intelligent Interface</td></tr>
<tr><td></td><td></td><td>Status Input</td><td>Status byte has been placed on the data bus by the Intelligent Interface</td></tr>
<tr><td></td><td></td><td>Positioning</td><td>Position command has been completed</td></tr>
<tr><td></td><td></td><td>Writing</td><td>An Intelligent Interface buffer is ready to be filled by the Tape Controller or a write-file-mark can be issued</td></tr>
<tr><td></td><td></td><td>Write File Mark</td><td>Write-file-mark command is completed</td></tr>
<tr><td></td><td></td><td>Reading</td><td>A buffer is ready to be emptied by the Tape Controller</td></tr>
<tr><td></td><td></td><td>Idle</td><td>Tape Controller is ready to receive a new command</td></tr>
</table>

4.3.1.1 Status Byte Transfer

The DIA instruction with a device code of 61 is used to transfer status bytes from the Tape Controller to the Intelligent Interface. The format of the instruction is

```
DIA ac,61
```

where ac is the processor accumulator to receive the data byte and 61 is the device code.

4.3.2 PROGRAMMED I/O OUTPUT

The DOA instruction with a device code of 61 is used to transfer control information from the Tape Controller to the Intelligent Interface. The instruction format is:

```
    DOA ac,61
```

where ac is the processor accumulator from which the control information will be transferred and 61 is the device code.

The format of the processor accumulator is shown in Figure 4-3; bit functions are described in Table 4-3.

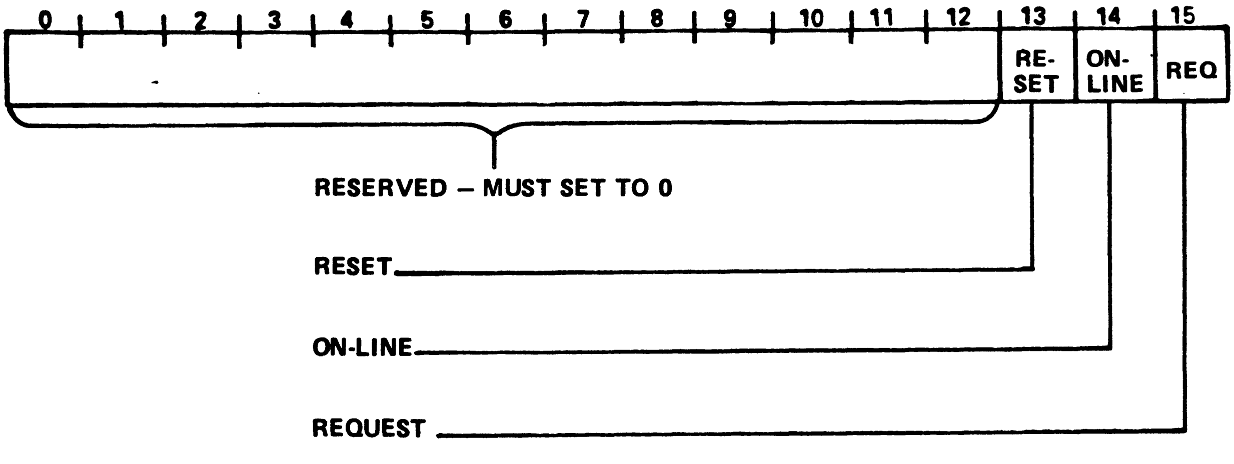


Figure 4-3. Contents of DOA Instruction Accumulator

TABLE 4-3. DOA INSTRUCTION ACCUMULATOR BITS

Bits	Signal	Function
0-12	-	Reserved - set to 0
13	RESET	Provides the capability to reset the Intelligent Interface. This bit causes a reset equivalent to a power-on reset. All operations abort and the Intelligent Interface waits for a new command.
14	ONLINE	This bit must be set for the following command modes: ● Writing ● Reading ● Write File Mark ● Read File Mark The on-line bit performs the following functions depending upon the operational mode. Mode On-line Bit Function Write The Intelligent Interface must be on-line to the Tape Controller to execute the write command. The Intelligent Interface continues to request data blocks as long as on-line is true. To terminate data transfer, the on-line bit must be set to 0. The interface writes the remaining buffers, writes a file mark and goes to BOT (beginning of tape). Read The Intelligent Interface must be on-line to the Tape Controller to execute the read command. The Tape Controller receives read data as long as on-line is set to 1. If on-line is set to 0, the Intelligent Interface will not attempt to transfer the next block even though read data may remain in the buffer memories. The tape is positioned to BOT.

TABLE 4-3. DOA INSTRUCTION ACCUMULATOR BITS (Cont)

Bits	Signal	Function
		Mode — On-line Bit Function Write File Mark — The Intelligent Interface must be on-line to the Tape Controller to execute the write-file-mark command. Read File Mark — The Intelligent Interface must be on-line to the Tape Controller to execute the read-file-mark command.
15	REQ	Used to initiate command sequences to the Intelligent Interface. Request (REQ) can only be set in response to the interface setting either ready (RDY) or exception (EXC). Request is used in conjunction with ready to transfer command and status bytes between the Tape Controller and the Intelligent Interface.

4.4 TAPE CONTROLLER IOCB

For DMA operations the input/output control block (IOCB) contains the command to be performed. The IOCB for the Tape Controller consists of three words located in main memory and used to indicate operational mode. The three words of the IOCB are set up by the software before the operation begins.

An additional word immediately following the IOCB must be reserved for a termination status word. Upon completion of an operation (successful or not), the controller writes operation complete into the termination status word. Figure 4-4 shows the format of the Tape Controller IOCB.

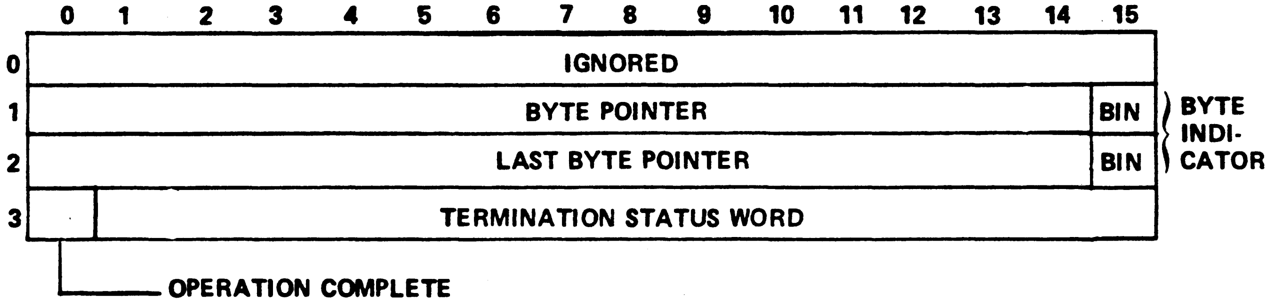


Figure 4-4. Tape Controller IOCB Format

Word 0 - Ignored.

Word 1 - Byte Pointer - points to one less than the first byte to be transferred. Bit 15 of word 1 is a byte indicator. If BIN=0 the left byte will be output. If BIN=1 the right byte will be output.

Word 2 - Last Byte Pointer - equal to word 1 + 1000 octal.

DMA transfer is accomplished one block at a time (400 words = 1000 bytes octal). Initiation of DMA transfer is effected with a DOBP command (see Section 4.4.1).

Word 3 - Termination Status Word - contains an operation-complete bit (bit 0). The operation-complete bit will be set upon completion of an operation, successful or not. As each byte is transferred, the byte pointer is incremented until it equals the last byte, which signals that the transfer is completed.

4.4.1 TAPE TRANSFER CONTROL

The processor has a dedicated register that is reserved as a pointer to the IOCB for the Tape Controller. This register may be written into or read from by using DOBP or DIBP instructions, respectively. The P control code is used to specify instructions directed to the Tape Controller (rather than the multiplexer (S) or disc controller (C)).

Tape transfer control is effected using DOBP and DIBP instructions with device code 77. The DOBP instruction activates and deactivates the Tape Controller; the DIBP instruction senses the idle state and polls the interrupt-pending bit in the tape IOCB pointer.

4.4.1.1 DOBP Instruction

The DOBP instruction is used for activating and deactivating the Tape Controller. The format of the control output instruction is

```
DOBP ac,77
```

where ac is a processor accumulator into which the address of the IOCB pointer will be loaded. This instruction is used for two purposes:

1. Activating the Tape Controller - The Tape Controller is activated when a DOBP instruction is issued with the contents of the accumulator containing the memory address of the first tape IOCB word and the most significant bit set to 0. When activated, the controller will go to a busy state and start the operation specified, e.g.

    a. Transfer direction (read or write), determined by the last DOA ac,61

    b. Memory address, determined by the Tape IOCB

2. Deactivating the Tape Controller - When the Tape Controller completes an operation and generates an interrupt, it must be deactivated by the software. The DOBP instruction with the accumulator set to the value zero is used to deactivate the controller. This resets the interrupt-pending bit and places the Tape Controller in an idle state.

4.4.1.2 DIBP Instruction

The format of the tape-input-control instruction is

```
    DIBP ac,77
```

where ac is the accumulator into which the value of the tape IOCB pointer is loaded.

The DIBP instruction may be used for two purposes:

1. Sensing the idle state - The idle state is defined as a zero value in the accumulator after executing a DIBP instruction to read the tape IOCB pointer.

2. Polling the interrupt-pending bit - When the Tape Controller has completed an operation, successful or not, it sets the interrupt-pending bit in the tape IOCB pointer. The interrupt-pending bit is the most significant bit in the tape IOCB pointer.

   The DIBP instruction can be used to read the IOCB pointer to check for the interrupt-pending bit set to 1. This allows for polling to check for devices requiring service.

4.5 TAPE CONTROLLER COMMANDS

Tape Controller commands for interface to the drives are single-byte commands. Bits 0 through 7 must be zero. The most significant bit is 0; the least significant bit is 15. The instruction format is:

```
DOA ac,62
```

The command format is shown in Figure 4-5.

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td colspan="8">0 ←→ 0</td><td colspan="3">COMMAND TYPE</td><td colspan="5">COMMAND DATA</td></tr>
</table>

Figure 4-5. Tape Controller Command Format

The command-type field selects one of seven possible commands. The command types and bit positions for bits 8, 9 and 10 are shown below. Commands are described in detail in the subsections that follow.

	Bit Position	
Type	8 9 10	Command
0	0 0 0	Select
1	0 0 1	Position
2	0 1 0	Write
3	0 1 1	Write File Mark
4	1 0 0	Read
5	1 0 1	Read File Mark
6	1 1 0	Read Status
7	1 1 1	Reserved for later use

The Intelligent Interface will accept any command when the ready bit has been set to one, and the interface is not currently executing a command. Only a read-status command will be accepted when the exception bit is set.

The on-line bit must be set for command types 2, 3, 4 and 5 to be executed; otherwise the interface will cause an exception status to be reported to the processor. The on-line control bit is ignored for command types 0, 1 and 6.

Direction is established in the write mode by issuing to the interface either command type 2 (write) or type 3 (write file mark). After write mode has been established, command types 2 or 3 will be accepted between data block transfers only. Direction is established in the read mode by issuing command type 4 (read) or type 6 (read status). After read mode has been established, no commands will be accepted between data blocks.

4.5.1 SELECT COMMAND (TYPE 0)

The select command is used to select one of four drives attached to the interface. For systems equipped with a single tape drive, use of this command is optional. The interface will default to tape drive 0 if no select command is issued. For multi-drive systems, this command should be issued before any of the other six commands. Once a drive has been selected by this command, it will remain selected until another select command is issued, or until power is removed from the tape unit.

Bit descriptions for the select command are shown below.

Bit	Description
15	Select drive 0
14	Select drive 1
13	Select drive 2
12	Select drive 3
11	Not used

NOTE

Execution of the select command is virtually instantaneous. If two or more drives are selected simultaneously, the interface will set the exception bit.

4.5.2 POSITION COMMAND (TYPE 1)

The position command permits mechanical manipulation of the tape for rewinding, erasing or retensioning.

Bit descriptions for the position command are shown below.

Bit	Description
15	BOT-rewind to beginning of tape
14	Erase tape
13	Retension tape
12	Set to 0
11	Set to 0

Upon receiving the position command, the Intelligent Interface will check the drive status for cartridge-in. If a cartridge is not installed in the drive, the controller will abort the position command and report an exception (illegal command) condition to the processor. At the completion of the position command, and if no abnormal conditions exist, the controller will report a ready status to the processor and wait for a new command.

Setting of bit 15 in the position command data field will reposition the tape at the BOT position. A BOT-position command may be issued prior to a read or write command. If a position command is not issued, the interface will default to BOT before executing a read or write command.

Setting of bit 14 in the command-data field erases the entire tape. Tape erasure occurs while writing is in progress, up to and including the end of that write routine. However, previously written data may still exist on this and other tracks when the file just written is shorter than the previously existing files. Erasure of a tape before writing new files on the tape prevents confusion in such an activity as reading file marks to determine the number of files on the tape. The erase-position command causes the tape to be moved from BOT to EOT with the erase channel activated and then returned to BOT.

Setting of bit 13 in the position command-data field causes a retensioning pass to be performed on the cartridge tape, as recommended by the tape manufacturer. When excessive read errors are encountered, a retensioning pass should be executed prior to writing. A retensioning pass should be executed any time an unreasonable number of tape errors is experienced.

4.5.3 WRITE COMMAND (TYPE 2)

The write command is used to initiate recording of data blocks onto the tape. Bit descriptions for the write command are shown below.

Bit	Description
15	Set to 0
14	Set to 0
13	Set to 0
12	Set to 0
11	Set to 0

The on-line control bit must be set to 1 before issuing a write command or an illegal command status will occur.

Upon receiving the write command, the interface will check the tape drive status for cartridge-in and write-protect status. If either condition exists, the write command is inhibited and an exception status will be reported to the processor.

If the on-line bit is reset to 0, the remaining buffered data will be written to tape, a file mark written, and the tape positioned at BOT. Otherwise, writing will continue until a write-file-mark command is issued between data blocks. A write-file-mark will cause the controller to finish writing the remaining buffer memory and then write the file mark. Another write command resumes recording of data. Another write-file-mark command or a position command can be issued to terminate writing.

4.5.4 END OF TAPE HANDLING

The interface may terminate writing by setting the exception bit to 1 due to an irrecoverable-data error or end-of-tape being sensed. When the early warning hole (EWH) of the last track is detected, the controller will discontinue accepting new data blocks, will write the remaining buffer memories, and will set the exception bit to warn the processor of a change in status.

When a read-status command is issued subsequent to EWH detection, the tape drive will stop, pending further instructions. Three possible courses of action may follow:

1. Issue a new write command

2. Issue a write-file-mark command

3. Set the on-line bit to 0

If a new write command is issued, two blocks of data will be accepted and the end-of-tape procedure repeated. These blocks are used to indicate a completed file or a file continued on another cartridge. A write-file-mark command followed by resetting the on-line bit causes a file mark and a return to BOT. A failure to issue either a write or write-file-mark command before resetting the on-line bit will reposition the tape to BOT without writing a file mark, and there will be no indication that the final record is complete. The blocks written from this write command or a write command with no data, followed by resetting of on-line to 0, will cause a file mark to be written and the tape returned to BOT.

4.5.5 WRITE FILE MARK COMMAND (TYPE 3)

The write-file-mark command provides for the separation of the data stored on the tape cartridge into more logical segments. This enables the use of better-defined data areas. Bit descriptions for the write-file-mark command are shown below.

Bit	Description
15	Set to 0
14	Set to 0
13	Set to 0
12	Set to 0
11	Set to 0

The on-line bit must be set to 1 when the write-file-mark command is issued or an illegal command status will occur. During execution of a write command, the write-file-mark command can only be issued between data blocks since the data bus is occupied by data during the block transfers. A write-file-mark command terminates the write command after the data block in progress has been recorded and then writes a file mark.

4.5.6 READ COMMAND (TYPE 4)

The read command is used to retrieve data blocks from a pre-recorded cartridge. Bit descriptions for the read command are shown below.

Bit	Description
15	Set to 0
14	Set to 0
13	Set to 0
12	Set to 0
11	Set to 0

The on-line bit must be set to 1 when the read command is issued or an illegal command status will occur. Before beginning a read operation, the interface will check the drive status for cartridge in place. If no cartridge is installed, the controller will report an error status by setting the exception bit.

The interface will set the exception bit to terminate the read operation after reading a tape file mark. The exception bit will also be set after transferring the block in error (BIE) when an irrecoverable read error occurs.

In order to continue data transfers, the on-line bit must be set to 1 and another read command issued. If the read command was terminated because a file mark was encountered, the new read command will cause the next file to be read from tape. If the read command was terminated because of an irrecoverable read error in the data field, issuing another read command causes the controller to continue reading the current file beginning at the next block. If the block in error was a file mark, the data transferred by the new read command will be from the next file.

No commands are accepted by the interface during execution of read data transfers. However, a read operation may be terminated between data blocks by issuing a DOA instruction with the on-line bit set to 0. When this instruction is received, the controller will position the tape to BOT. The tape may be returned to BOT after the completion of an entire read command by issuing a position-to-BOT command while the on-line bit is still set to 1.

4.5.7 READ FILE MARK COMMAND (TYPE 5)

The read-file-mark command is used to detect file marks. Bit descriptions for the read-file-mark command are shown below.

Bit	Description
15	Set to 0
14	Set to 0
13	Set to 0
12	Set to 0
11	Set to 0

The on-line bit must be set to 1 when the read-file-mark command is issued or an illegal command status will occur. Before beginning a read-file-mark operation, the interface will check the drive status for cartridge in place. If no cartridge is in place, the interface will report an error status to the processor by setting the exception bit to 1.

After reading the tape file mark, the interface will set the exception bit to 1. To continue reading file marks, the on-line bit must be kept set to 1 and another read-file-mark command issued. If the block in error is a file mark, the new read-file-mark command will begin reading at the next file mark following the block in error.

For a read-file-mark command, no actual data is transferred to the processor. To terminate a read-file-mark operation, reset the on-line bit to 0. When this instruction is received, the controller will position the tape to BOT. The tape may be returned to BOT after the completion of an entire read-file-mark command by issuing a position-to-BOT command while the on-line bit is still set to 1.

4.5.8 READ STATUS COMMAND

The read-status command is used to retrieve six bytes of information from the interface. The interface status can be read under two conditions:

- When the interface sets the exception bit to 1 due to a change in status
- At the normal completion of a data transfer command

The command data field of the read-status command is defined as follows:

Bit	Description
15	Set to 0
14	Set to 0
13	Set to 0
12	Set to 0
11	Set to 0

4.5.8.1 Exception Status Bytes

Six exception-status bytes define the conditions which can cause the interface to report an exception status. The six status bytes transferred are defined in Table 4-4.

TABLE 4-4. STATUS BYTE DEFINITIONS

Byte	Bit	Operation	Exception Status
0	15	-	File Mark Detected
	14	-	Block in Error Not Detected
	13	-	Irrecoverable Data Error
	12	-	End of Tape
	11	-	Write Protected
	10	-	Drive Not On-line
	9	-	Cartridge Not in Place
	8	-	Exception Byte 0
1	15	-	Reset Occurred
	14	-	Reserved - Set to 0
	13	-	Reserved - Set to 0
	12	-	Reserved - Set to 0
	11	-	Reserved - Set to 0
	10	-	No data detected
	9	-	Illegal command
	8	-	Exception Byte 1
2&3	-	Write	Data Error counter: Number of blocks rewritten
		Read	Data Error counter: Number of soft read errors
4&5		Write Read	Number of buffer underruns

4.5.8.1.1 EXCEPTION BYTE 0

The eight bits of exception byte 0 are shown in Figure 4-6 and defined in Table 4-5.

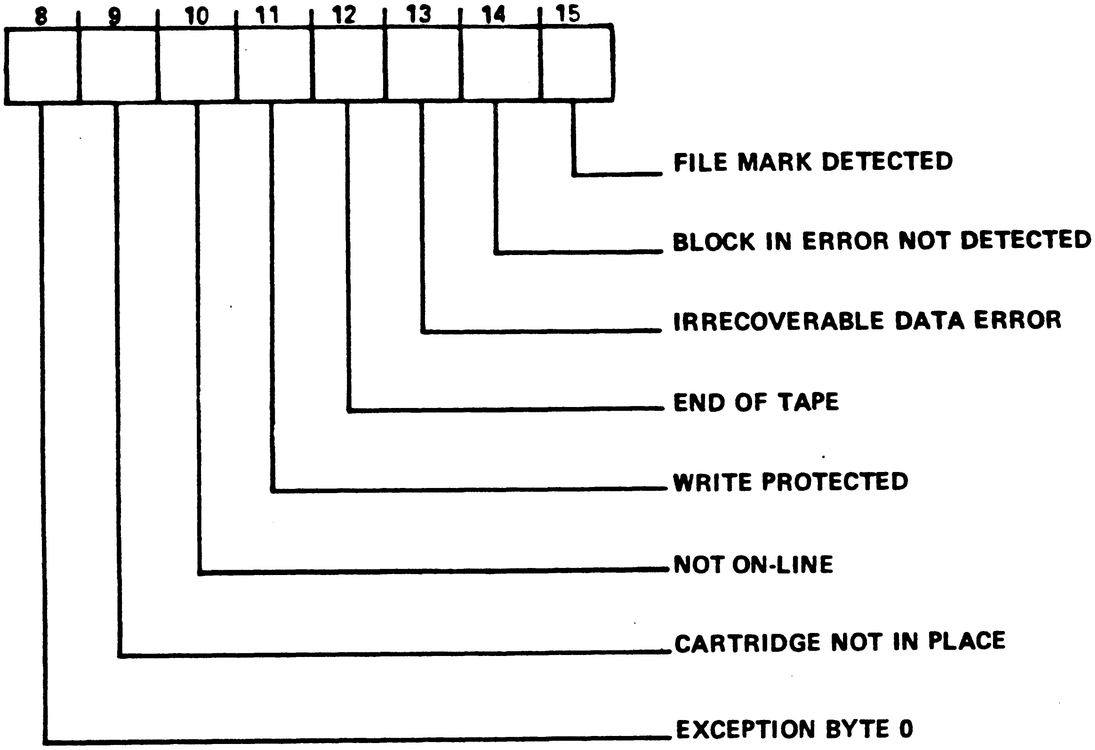


Figure 4-6. Exception Byte 0

TABLE 4-5. EXCEPTION BYTE 0 BIT DEFINITION

Bit	Name	Definition
15	File Mark Detected	Set to one each time the controller-read channel encounters a file mark during a read or position command. This bit and the exception bit will be reset to 0 after a read-status command is executed. This bit does not indicate an error condition and data transfers or tape positioning can continue if the processor issues another read or position command.
14	Block In Error Not Detected	Set to one along with the irrecoverable-error status bit (bit 13) if the last block transferred during a read command was not the block containing the irrecoverable error.
13	Irrecoverable Data Error	Set to one when the soft error retry sequence fails to recover a data block from the tape during a read or read-file-mark command. This bit will be set to one on a write or write-file-mark command if the same data block is rewritten more than 16 times. This bit and the exception bit will be reset to 0 after a read-status command has been executed. The tape will be repositioned to BOT after the read-status command if the irrecoverable error occurred during a write or write-file-mark command.
12	End of Tape	When set to one, indicates that an attempt was made to read past the EOT holes for the last track or may write past the end of the recording area on the last track. For write operations, two additional data blocks and a file mark can be written after encountering an end-of-tape status report. For read operations, if this status bit is set, it indicates that the controller missed the last file mark on the tape, or that the tape was positioned past the final file mark when another read or read-file-mark command was issued. This bit and the exception bit will be reset after execution of the read-status command.

TABLE 4-5. EXCEPTION BYTE 0 BIT DEFINITION (Cont)

Bit	Name	Definition
11	Write Protected	When set to one, operator intervention is required. Set if the write-protect plug is set in the file-position (bracket safe). Operator must determine error-status cause before this bit can be reset to 0.
10	Drive Not On-line	Set if the selected drive is not physically connected to the interface, or is not receiving power.
9	Cartridge Not in Place	When set to one, operator must check cartridge placement before this bit can be reset to 0.
8	Exception Byte 0	Set to one when any other bit in exception byte 0 is set to one. Reset to 0 after a read-status command.

4.5.8.1.2 EXCEPTION BYTE 1

The eight bits of exception byte 1 are shown in Figure 4-7, and defined in Table 4-6.

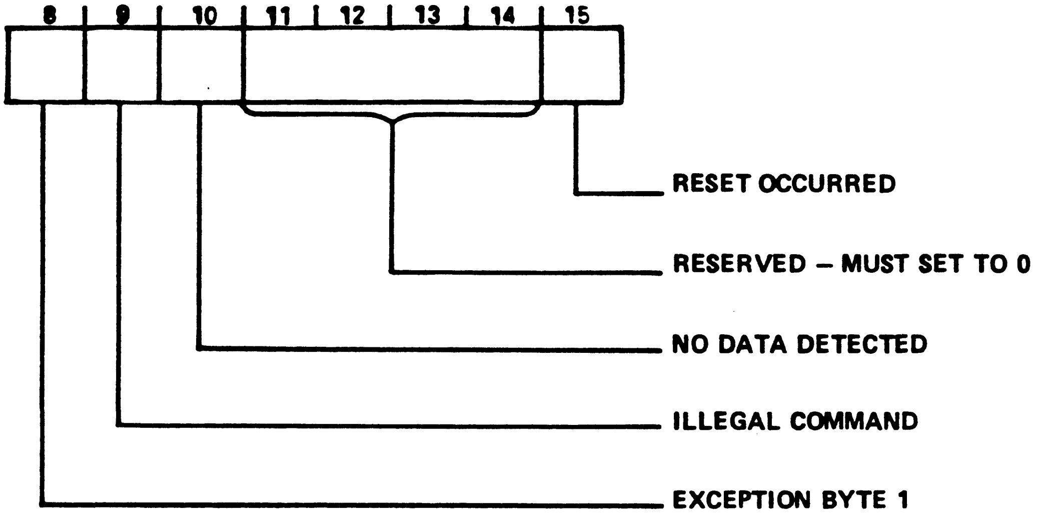


Figure 4-7. Exception Byte 1

TABLE 4-6. EXCEPTION BYTE 1 BIT DEFINITION

Bit	Name	Definition
15	Reset Occurred	When set to one, indicates a reset has occurred. Causes all operations to abort. The interface sets the exception bit.
14		Reserved - set to 0
13		Reserved - set to 0
12		Reserved - set to 0
11		Reserved - set to 0
10	No Data Detected	Set when an irrecoverable-data error occurs due to insufficient recorded data, caused by failure to detect a data block within an interface timeout period. It is reset after a read-status command.
9	Illegal Command	Detected by the controller and set to one when the following conditions exist: 1. A select command was issued with more than one drive specified or no drive specified. 2. A position command was issued with more than one or no qualifier specified. 3. The on-line bit is reset to 0 and a write, write-file-mark, read or read-file-mark command is issued. 4. A command other than write or write-file-mark was issued during execution of a write command.
8	Exception Byte 1	Set to one when any other bit in status byte 1 is set. It is reset to 0 after issuing a read-status command.

4.5.8.1.3 BYTES 2 AND 3

Status bytes 2 and 3 provide a 16-bit binary count of the tape data errors. Byte 2 contains the most significant byte and byte 3 contains the least significant byte. The counter is reset to 0 after issuing a DOA instruction with the on-line bit set to one.

During write operations, the counter will increment for each data block that is rewritten due to a read-after-write error. Since the rewrite operation rewrites two data blocks for each error, the counter will be incremented twice for each error. A four-track, 450-foot tape cartridge will store approximately 40,000 512-byte blocks.

During read operations, the data error counter will be incremented for each data block containing an unexpected cyclic-redundancy-code (CRC) error. Expected CRC errors are blocks that were rewritten during the write operation. The counter is incremented for each read retry until the block is successfully read or logged as an irrecoverable-data error.

4.5.8.1.4 BYTES 4 AND 5

These two bytes will contain a 16-bit binary count of buffer underruns. Byte 4 contains the MSB and byte 5 contains the LSB. For write operations, this count will be incremented each time the processor is unable to keep data flowing to the interface. If the interface is ready to write the next block, but the buffer is not full and ready to write, the interface will stop tape motion and wait for the processor.

During a read operation, this count will increment when the processor is unable to empty the interface buffers fast enough. If an empty buffer is not available for the next block of data to be read from the tape, tape motion will stop.

4.6 TAPE COMMAND INTERFACE

This section describes the sequence of events that occur when I/O commands interface with the tape drive. Flowcharts for the various operational modes are provided in Appendix B. The operational sequences described in Sections 4.6.1 through 4.6.6 are for error-free handling. Refer to Section 4.6.7 for error processing and recovery.

4.6.1 COMMAND TRANSFER SEQUENCE

The sequence for transferring a command between the processor and the interface is:

1. The program places the command byte on the command data bus.

2. The program sets the REQ (request) bit.

3. The interface takes the command byte from the command data bus and sets the RDY (ready) bit.

4. The processor receives RDY (ready) and resets REQ (request).

5. The interface resets RDY (ready).

See Appendix B-1 for command transfer sequence flowchart.

4.6.2 READ STATUS COMMAND SEQUENCE

The sequence for reading status information is:

1. The program issues a read-status command as described in Section 4.6.1.

2. The interface takes the command from the command data bus and sets the DIRC (direction) bit.

3. The interface places the status byte on the command data bus.

4. The interface sets the RDY (ready) bit.

5. The status byte is transferred from the interface to the processor.

6. The program sets the REQ (request) bit.

7. The interface resets RDY (ready).

8. The program drops REQ (request).

9. Steps 3 through 8 are repeated until all six status bytes have been transferred.

10. The interface resets the DIRC (direction) bit.

See Appendix B-2 for read-status sequence flowchart.

4.6.3 WRITE COMMAND OPERATIONS

The sequence of events for write command operations is:

1. The processor, in preparation for a write operation, checks to see if the interface is ready. If it is, the processor will place a write command on the bus and notify the interface.

2. The interface will read the command, verify that a cartridge is in place and not write-protected, and signal the processor that the command was accepted.

3. If the processor had not preceded the write command with select and position commands, the interface will automatically select unit 0 and track 0, BOT.

4. The interface will begin accepting data. When the first buffer is full, the interface will begin writing.

5. The interface will write blocks of data on the tape, automatically adding gap, sync, block address and CRC.

6. After the data is written, it passes over the read head, which read-checks the data for CRC errors.

7. After the data has been verified by the read-after-write check, the buffer which held that data will be released for the next transfer of data from the processor.

8. The interface will continue writing blocks of data to the tape until it detects the early warning hole. When the interface detects this hole, it will stop accepting new data from the processor on the next block boundary. The interface will write all existing buffers to tape and then write a constant gap signal to the end-of-tape (EOT) hole.

9. The interface will then deselect the write mode, disable the erase head and stop tape motion.

10. The interface will switch to track 1, reverse the capstan motor direction, and start tape motion. When the tape is up to speed and in the recording zone, the interface will resume writing.

11. The interface will continue writing for the full length of track 1. When the load-point hole is reached, the interface will again halt tape motion. This time the controller will physically reposition the read/write head and select track 2.

12. If the processor continues writing data until the early warning hole of track 3, the interface will stop accepting data and inform the processor that the tape is full. The processor may now write a block describing that the file is not complete and write a file mark.

See Appendix B-3 for write sequence flowchart.

4.6.4 READ COMMAND OPERATIONS

Read command operations follow the sequence of events detailed below.

1. The program issues the read command.
2. The interface positions the tape to BOT of track 0.
3. The interface brings the tape up to speed and begins searching for the first data block.
4. The interface reads the entire data block to the read buffer and checks the CRC and the block address number.
5. If the CRC and block address are good, the block is transferred to the processor.
6. Steps 4 and 5 are repeated for all of the blocks on track 0.
7. When the EOT (end-of-tape) holes are detected for track 0, the interface stops the drive, selects track 1, reverses the direction of the drive's capstan motor and brings the tape up to speed.
8. All of the records on track 1 are read, error checked and transferred to the processor.
9. Tracks 2 and 3 are handled in a similar manner to tracks 0 and 1.
10. The read command can be terminated by the program any time a block boundary is reached, or automatically by the interface if a file mark or EOT holes for the last track are detected.

See Appendix B-4 for read sequence flowchart.

4.6.5 WRITE FILE MARK OPERATIONS

The write-file-mark command will generate a standard length data block with unique codes in the data field, a block address and a CRC. The processor does not transfer any data for the file mark block. The sequence of events is:

1. The program issues the write-file-mark command at any data block boundary.
2. The interface writes the remaining blocks of data from the interface buffers to the tape.
3. The interface generates a file-mark block and writes this block to the tape.
4. The file-mark block is checked with a read-after-write operation and if an error occurs it is treated like any other block in error (BIE).

There is no restriction concerning the number of file marks which can be written on a tape. There are also no restrictions as to the number of user blocks which can be written between file marks.

A read command will terminate upon encountering a file mark and will set the file-mark bit in the status words.

If the program terminates a write command without issuing a write-file-mark command (i.e., by resetting the on-line bit), the controller automatically generates a file mark.

After issuing a write file mark, the program can immediately issue another write command (or write-file-mark command) and continue transferring files to tape.

4.6.6 READ FILE MARK OPERATIONS

Read-file-mark operations follow the sequence of events detailed below.

1. A read-file-mark command is performed the same as a read command, but no data is transferred to the processor.
2. A read-file-mark command is terminated by reading a file mark. Counting the total number of files (or file marks) on a tape would require reissuing the read-file-mark command for each file mark found.

4.6.7 ERROR PROCESSING AND RECOVERY

The Intelligent Interface provides error processing and recovery sequences which are transparent to the program/processor and reduces the load on the processor. Information passed to the processor consists only of statistical data on the number of errors automatically processed by the interface. These statistics can be used for evaluating media integrity and system performance.

The following subsections discuss simple error handling. The controller is capable of handling compound (multiple) errors. However, this discussion is intended only to illustrate the principle of error recovery procedures.

4.6.7.1 Write Buffer Overrun

Streaming-tape systems use constant tape motion with small gaps between the data blocks. This means the processor must maintain uninterrupted transfer of data blocks to the controller. Once writing has been initiated and a full buffer is not available for transfer to the drive, the interface will automatically extend the length of the inter-record gap up to 0.3 inches (a little over half a record). The automatic gap extension will be logged in the statistical counters. If a buffer becomes available before the end of the extended gap, the gap extension will be terminated and the buffer written immediately. Regardless of the length of the gap extension, the statistical counter will be updated for each gap extension. A complete buffer overrun will occur when at the end of the extended gap period, a buffer is still not available for the write channel. Gap extension is provided to allow for processor system contention without decreasing the system throughput by terminating tape motion. The number of gap extensions will directly subtract from the total number of data blocks available on the tape (approximately 10,000 per track). For this reason, write buffer overruns are highly undesirable.

4.6.7.2 Read After Write Errors

With high-density streaming tapes, information is recorded in 0.0001-inch increments. The recording area must be free of all contamination and imperfections approaching 0.0001 inches. For this reason, the interface is designed to accommodate occasional data errors. A read-check is performed on each block of data immediately following the write operation. If a CRC error is found during the read-check, that block will be rewritten.

The tape drive head has two gaps, one for writing and one for reading. These gaps are separated by a distance of 0.3 inches. For tape streaming the inter-record gap length is only 0.013 inches. Therefore the interface must begin writing the next record before the previous record has been verified.

If the block being verified has a CRC error, the interface will continue rewriting the blocks until no read-after-write errors are detected, or until 16 rewrites have been performed. If 16 rewrites are performed, the interface terminates the write command and reports an irrecoverable data error to the processor. A read-after-write counter is provided to inform the program of the number of blocks automatically rewritten by the interface. These rewritten blocks will subtract from the total capacity of the tape.

4.6.7.3 Read Buffer Underruns

The interface has an extra buffer reserved in case the processor system gets behind the transfer rate of the read channel. This provides a one-block buffer to allow for short-term processor system contentions before the read channel overruns the controller buffer.

If the processor fails to stay ahead of the read channel, a read-buffer underrun will occur. This condition arises when the read channel has located the next block of data and no interface buffer is available. To prevent the loss of this block, the interface must stop the tape, back up the tape and wait for a buffer to become available. The controller restarts the tape and begins reading until it locates the block that underran the buffer. From this point on, the normal read sequence is resumed.

Since the read buffer underrun recovery is invisible to the processor, a statistical counter is provided to keep track of the number of underruns.

4.6.7.4 Read Data Errors

The read-after-write check operation occasionally detects errors. To ensure data integrity, the error recovery routine rereads the block in error (BIE) up to 16 times. The process of rereading the BIE is a soft error retry.

If the data block has not been successfully recovered after 16 retries, the interface will transfer the BIE if it can be located, terminate the read operation, and report an irrecoverable read error to the processor. A block is always transferred. If it is not the block in error, the processor will be notified.

Since soft error retries are invisible to the processor, a statistical counter is available. This counter is updated for each soft error retry. Data blocks with CRC errors that were rewritten during the write process do not increment the soft error retry counter.

4.6.7.5 Read Sequence Errors

The interface appends a block-address byte to each data block written to the tape. During a write operation, a data block with a read-after-write check will be rewritten. Rewritten blocks will alter the normal sequence of blocks. The interface uses the block-address byte to maintain the proper sequence of data blocks sent to the processor.

During read operations, when a data block without a CRC error is read and the next block contains an unexpected address, a block-sequence error results. Block-sequence errors will automatically invoke a soft error retry. This retry sequence is the same as performed for read-data errors. The soft error retry counter will be incremented for each retry until the proper sequence is re-established, or until a limit of 16 retries is exceeded. The interface will transfer the BIE if it can be located, terminate the read command and report an irrecoverable data error to the processor. A block is always transferred. The processor will be informed if it is not the BIE.

APPENDICES

Appendix A
ASCII CODES

TABLE A-1. ASCII CODES IN OCTAL

000	NUL	<CTRL-@>	040	BLANK	100	@	140	`
001	SOH	<CTRL-A>	041	!	101	A	141	a
002	STX	<CTRL-B>	042	"	102	B	142	b
003	ETX	<CTRL-C>	043	#	103	C	143	c
004	EOT	<CTRL-D>	044	$	104	D	144	d
005	ENQ	<CTRL-E>	045	%	105	E	145	e
006	ACK	<CTRL-F>	046	&	106	F	146	f
007	BEL	<CTRL-G>	047	'	107	G	147	g
010	BKSP	<CTRL-H>	050	(	110	H	150	h
011	HTAB	<CTRL-I>	051	)	111	I	151	i
012	LF	<CTRL-J>	052	*	112	J	152	j
013	VTAB	<CTRL-K>	053	+	113	K	153	k
014	FF	<CTRL-L>	054	,	114	L	154	l
015	CR	<CTRL-M>	055	-	115	M	155	m
016	SO	<CTRL-N>	056	.	116	N	156	n
017	SI	<CTRL-O>	057	/	117	O	157	o
020	DLE	<CTRL-P>	060	0	120	P	160	p
021	XON	<CTRL-Q>	061	1	121	Q	161	q
022	AUXON	<CTRL-R>	062	2	122	R	162	r
023	XOFF	<CTRL-S>	063	3	123	S	163	s
024	AUXOFF	<CTRL-T>	064	4	124	T	164	t
025	NAK	<CTRL-U>	065	5	125	U	165	u
026	SYN	<CTRL-V>	066	6	126	V	166	v
027	ETB	<CTRL-W>	067	7	127	W	167	w
030	CAN	<CTRL-X>	070	8	130	X	170	x
031	ENDMD	<CTRL-Y>	071	9	131	Y	171	y
032	SUB	<CTRL-Z>	072	:	132	Z	172	z
033	ESC	<CTRL-[>	073	;	133	[	173	{
034	F SEP	<CTRL-\>	074	<	134	\	174	|
035	G SEP	<CTRL-]>	075	=	135	]	175	}
036	R SEP	<CTRL-^>	076	>	136	^	176	~
037	U SEP	<CTRL-_>	077	?	137	_	177	DEL

Appendix B
TAPE HANDLING FLOWCHARTS

This appendix contains programming flowcharts for typical tape handling operations. They are intended as a guide to and a pictorial representation of the tape handling command sequences discussed in Section 4.

Operations covered by these flowcharts are:

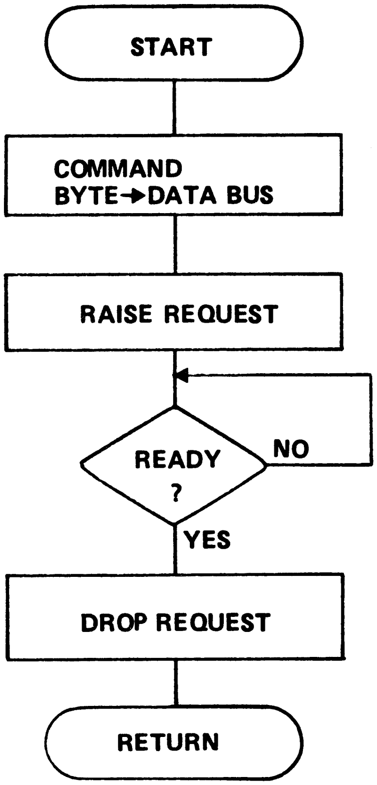


Figure B-1. Command Transfer Sequence Flowchart

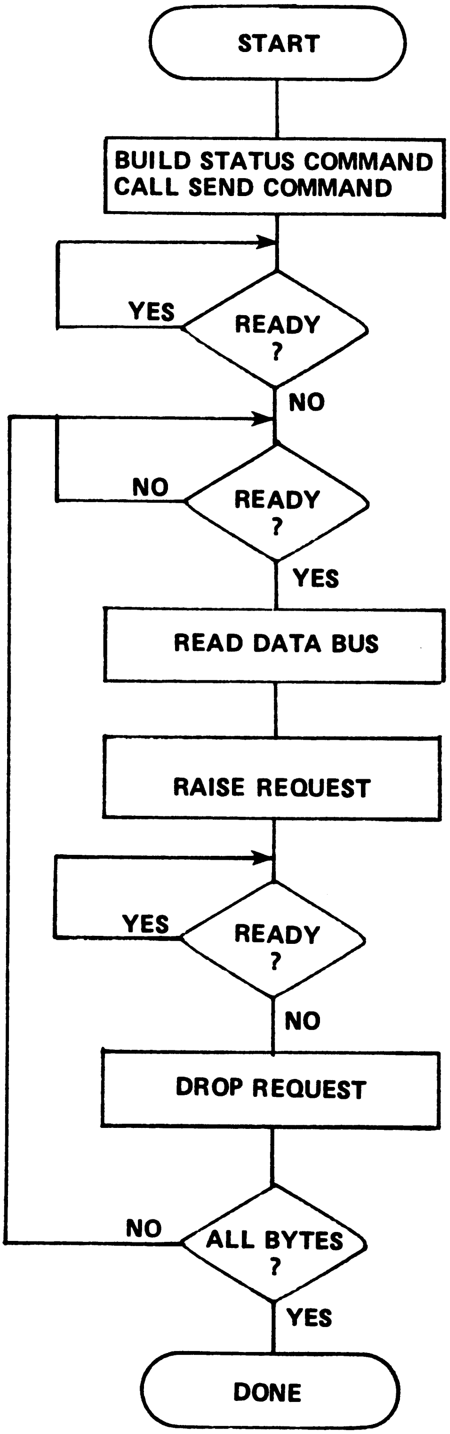


Figure B-2. Read Status Sequence Flowchart

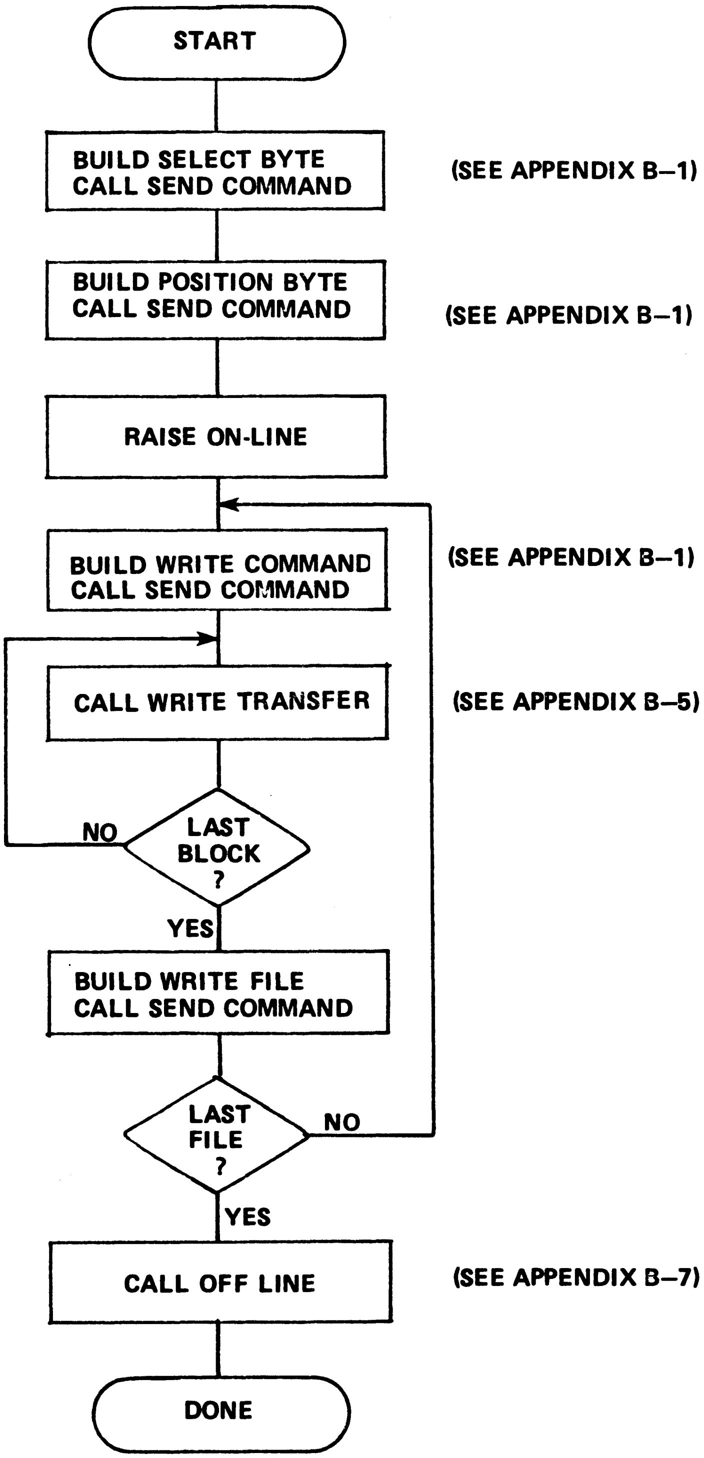


Figure B-3. Write Sequence Flowchart

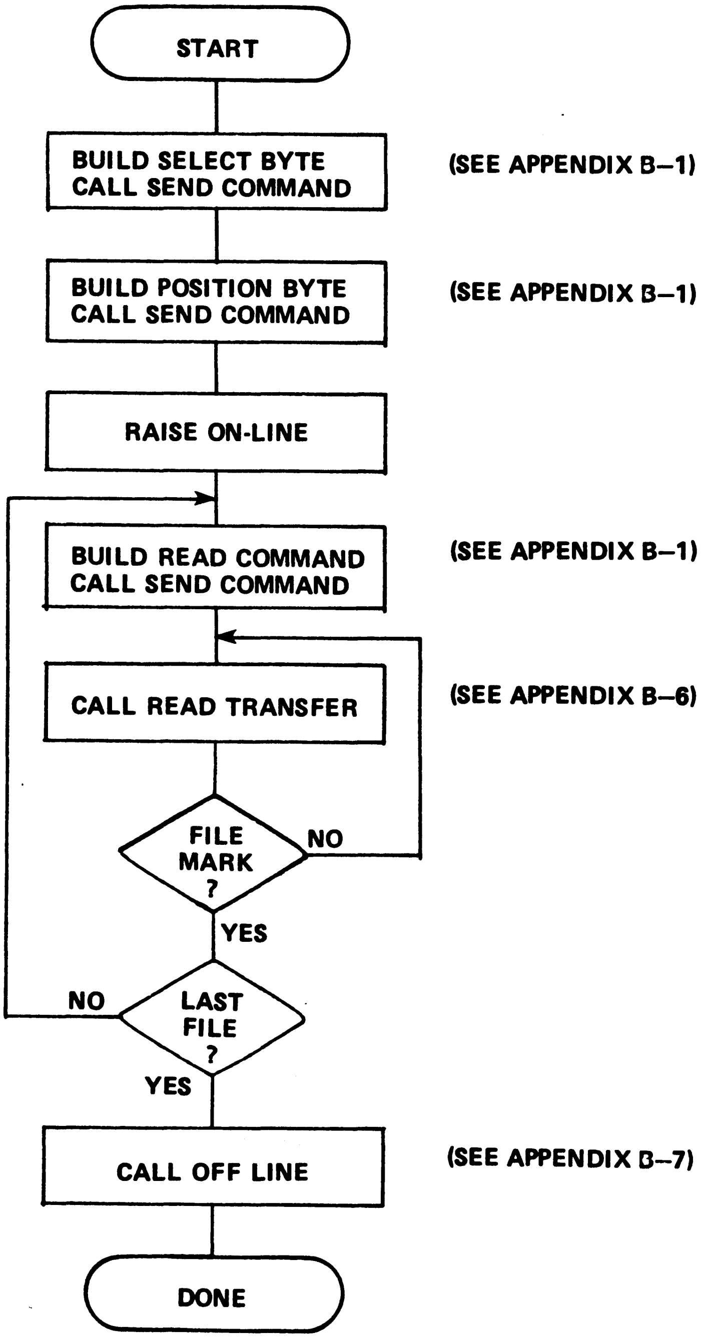


Figure B-4. Read Sequence Flowchart

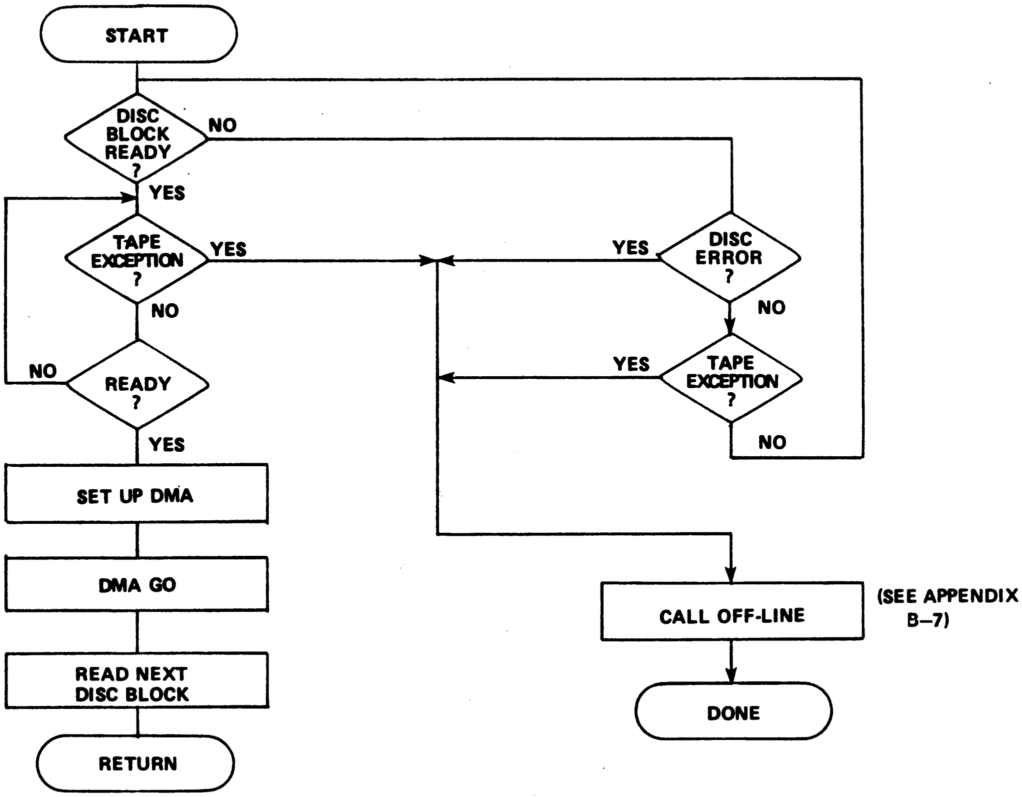


Figure B-5. Write Transfer Sequence Flowchart

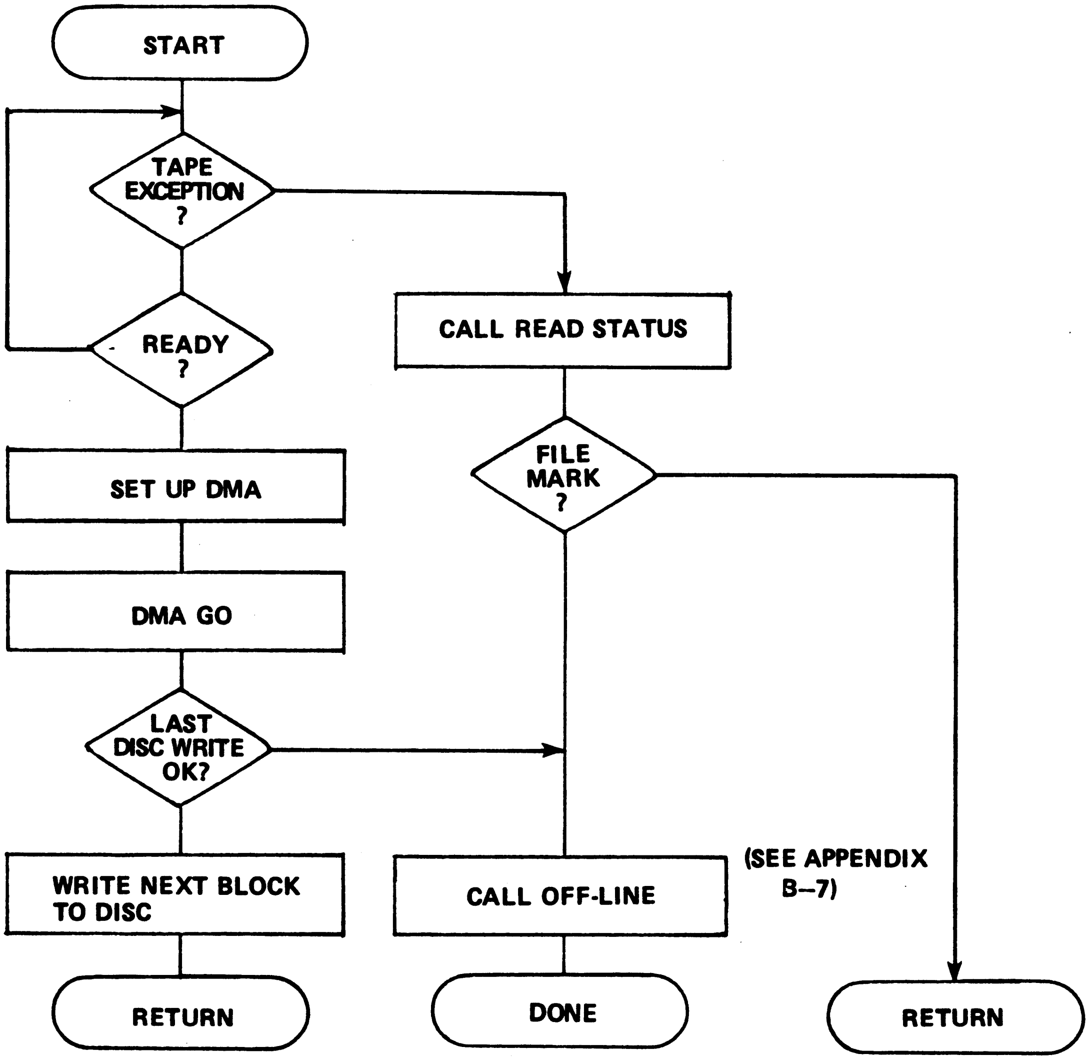


Figure B-6. Read Transfer Sequence Flowchart

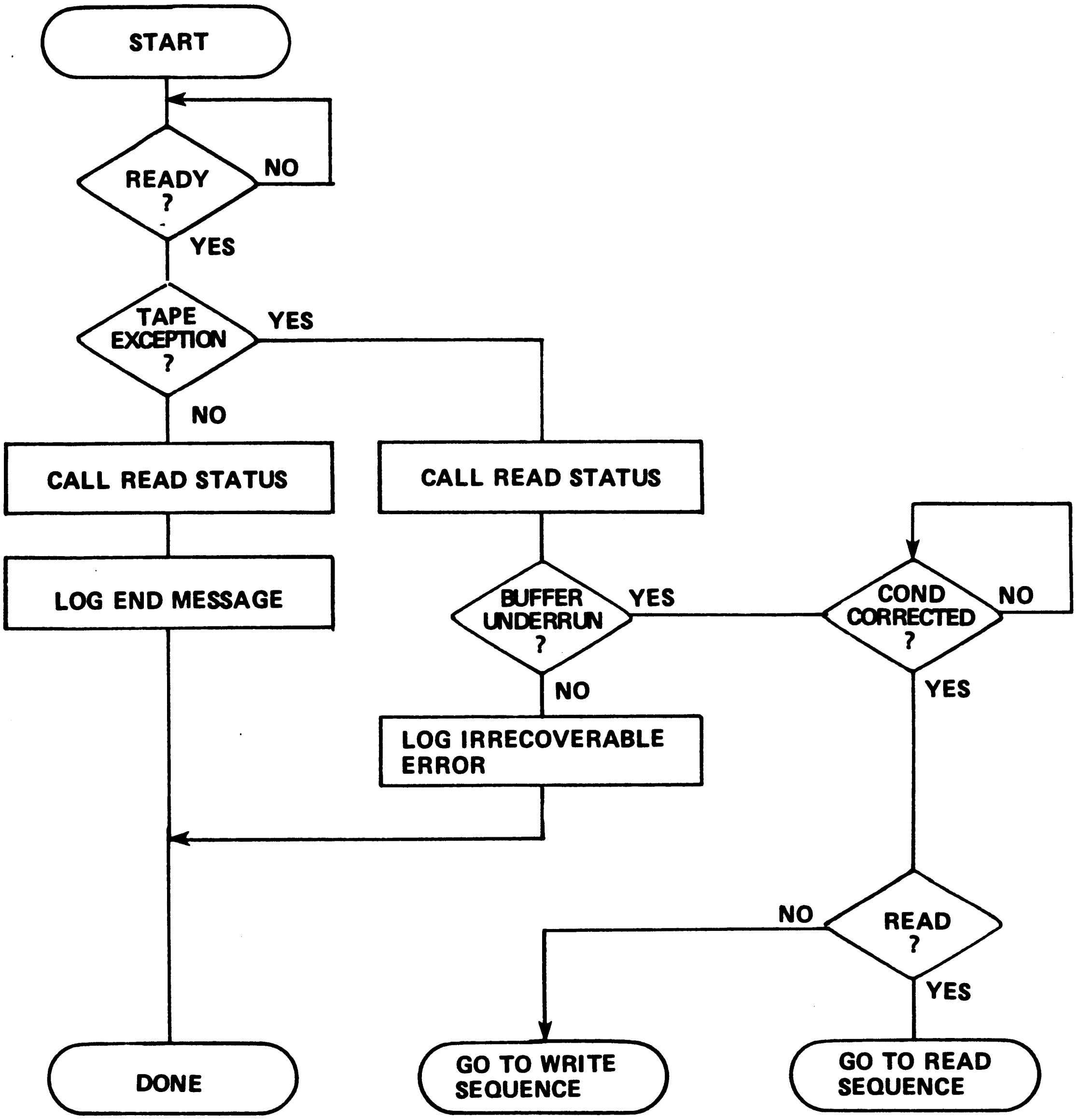


Figure B-7. Off-line Sequence Flowchart

COMMENT SHEET

MANUAL TITLE POINT 4 MARK 3 Peripherals Interface Manual

PUBLICATION NO. HM-081-0027 REVISION A

FROM: NAME/COMPANY: ______________________

BUSINESS ADDRESS: ______________________

CITY/STATE/ZIP: ______________________

COMMENTS: Your evaluation of this manual will be appreciated by POINT 4 Data Corporation. Notation of any errors, suggested additions or deletions, or general comments may be made below. Please include page number references where appropriate.

NO POSTAGE NECESSARY IF MAILED IN U.S.A.
Fold on Dotted Lines and Tape

NO POSTAGE
NECESSARY
IF MAILED IN
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 5755 SANTA ANA, CA

POSTAGE WILL BE PAID BY ADDRESSEE

POINT 4 Data Corporation
PUBLICATIONS DEPARTMENT
2569 McCabe Way
Irvine, CA 92714

CUT ON THIS LINE

POINT 4 DATA CORPORATION

2569 McCabe Way / Irvine, California 92714 / (714) 754-4114