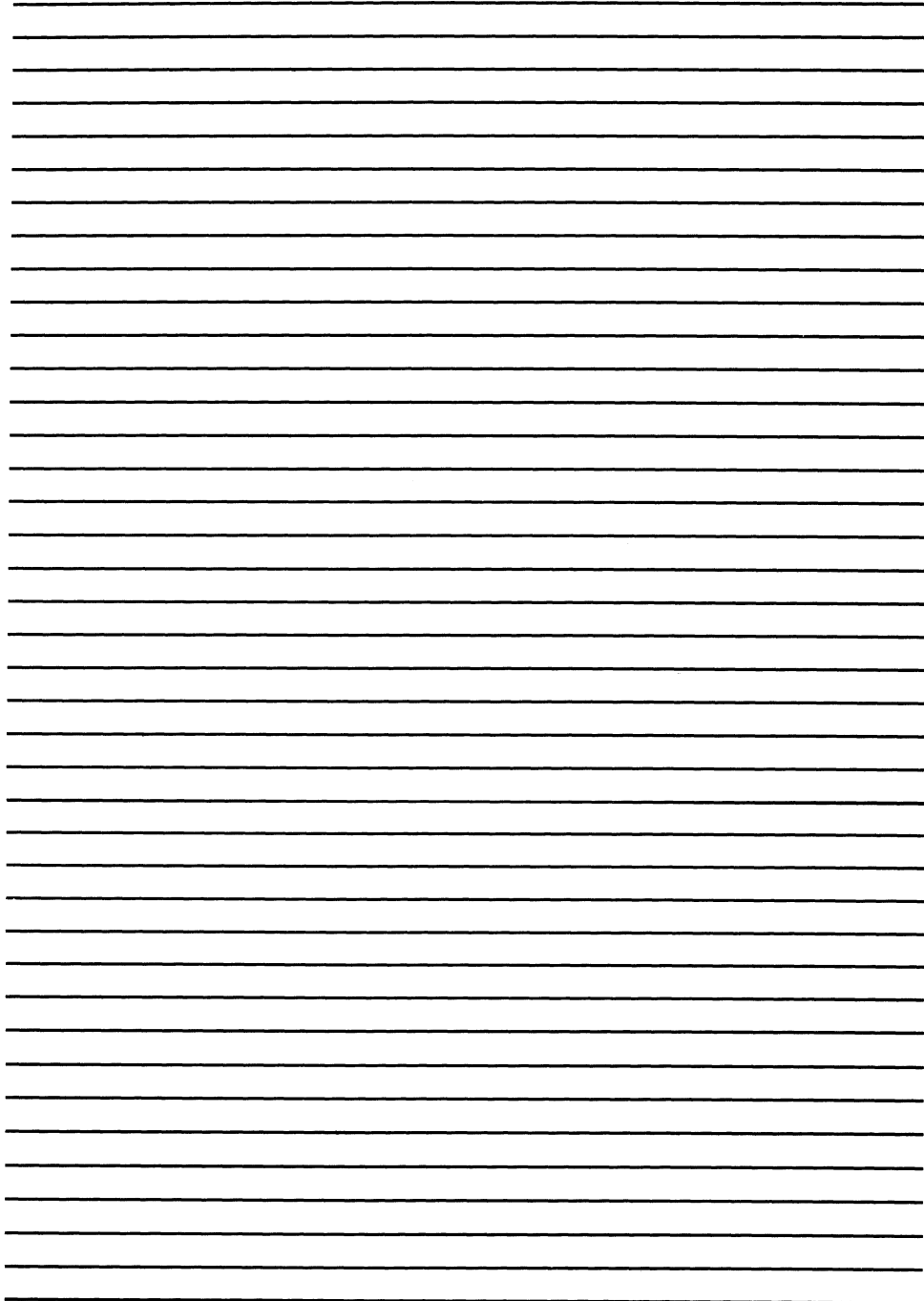


MARK 2/3

*Peripherals
Interface
Manual*







POINT 4TM
MARK 2/3
PERIPHERALS
INTERFACE
MANUAL

Revision D

NOTICE

Every attempt has been made to make this 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.

Copyright © 1982, 1984 by POINT 4 Data Corporation (formerly Educational Data Systems, Inc). Printed in the United States of America. All rights reserved. No part of this work covered by the copyrights hereon may be reproduced or copied in any form or by any means--graphic, electronic, or mechanical, including photocopying, recording, taping, or information and retrieval systems--without the prior written permission of:

POINT 4 Data Corporation
15442 Del Amo Avenue
Tustin, CA 92680
(714) 259-0777

REVISION RECORD

PUBLICATION NUMBER: HM-081-0027

<u>Revision</u>	<u>Description</u>	<u>Date</u>
A	Initial Release	08/06/82
B	Update package incorporating information on the MARK 3B PIB	10/30/82
C	Update package incorporating information on the MARK 2 PIB	02/20/84
D	Update package incorporating information relevant to the MARK 2 with floppy disk drive	12/01/84

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. The effective revision for each page is shown below.

<u>Page</u>	<u>Rev</u>	<u>Page</u>	<u>Rev</u>	<u>Page</u>	<u>Rev</u>
Cover	-	4-7 thru 4-37	A		
Title	D	5-1, 5-2	C		
ii	C	5-3	B		
iii, iv	D	5-4	C		
v/vi	C	5-5 thru 5-14	B		
vii thru xi	D	5-15, 5-16	C		
xii	C	5-17 thru 5-40	B		
xiii	D	5-41	C		
xiv	C	6-1 thru 6-28	D		
xv	D	Appendix Title	-		
1-1	D	A-1	A		
1-2 thru 1-4	C	B-1 thru B-8	A		
1-5	D	C-1 thru C-12	B		
1-6 thru 1-9	C	Comment Sheet	D		
1-10, 1-11	D	Mailer	-		
1-12 thru 1-30	C	Back Cover	-		
2-1	B				
2-2, 2-3	C				
2-4 thru 2-28	A				
2-29	B				
2-30, 2-31	A				
2-32	B				
3-1, 3-2	C				
3-3, 3-4	B				
3-5 thru 3-18	A				
3-19	D				
3-20 thru 3-32	A				
3-33 thru 3-35	C				
3-36, 3-37	A				
3-38, 3-39	C				
3-40	D				
3-41, 3-42	C				
3-43 thru 3-49	A				
4-1 thru 4-6	C				

PREFACE

This manual describes the peripheral interface functions supported by the POINT 4 MARK 2 or 3 Computer System. The peripheral functions are controlled by a Peripherals Interface Board (PIB). The PIB is available in three versions. All versions contain a built-in multiplexer and a disc drive interface. However, the disc drive interfaces are different on the MARK 2 and MARK 3 systems. In addition, the MARK 2 and the MARK 3 PIBs provide a tape drive interface. The MARK 3B PIB provides a floppy disc drive interface.

The introduction provides information regarding the characteristics, internal architecture, and input/output interface handling of each of the PIBs.

Separate sections are devoted to the multiplexer, the different disc drives, tape drive, and floppy disc drive interfaces. The appendices provide a table of ASCII codes, and both tape and floppy disc handling flowcharts.

Separate manuals describe the MARK 2/3 Computer System and provide step-by-step instructions for installation, upgrade and operating procedures.

Related manuals include:

<u>Title</u>	<u>Pub. Number</u>
POINT 4 MARK 3 Computer System Manual	HM-081-0019
POINT 4 MARK 3 Diagnostics Document	
POINT 4 MARK 2 Diagnostics Document	
POINT 4 MARK 3T Computer User Guide	HM-081-0029
POINT 4 MARK 2 Computer User Guide	HM-081-0031



CONTENTS

<u>Section</u>	<u>Title</u>	<u>Page</u>
1	INTRODUCTION	1-1
1.1	GENERAL DESCRIPTION	1-1
1.1.1	System Description	1-6
1.1.2	Equipment Characteristics	1-8
1.1.2.1	Communications Control	1-8
1.1.2.2	Disc Drive Interface on MARK 3 and MARK 3B PIBs	1-9
1.1.2.3	Disc Drive Interface on a MARK 2 PIB	1-9
1.1.2.4	Tape Drive Interface on MARK 3 and MARK 2 PIBs	1-9
1.1.2.5	Floppy Disc Drive Interface on a MARK 3B PIB	1-10
1.1.2.6	Floppy Disc Drive Interface on a MARK 2 PIB	1-10
1.1.2.7	Input/Output Instructions	1-11
1.1.3	PIB Internal Architecture	1-12
1.2	INPUT/OUTPUT INTERFACE HANDLING	1-16
1.2.1	Program Interrupt and Priority Scheme	1-16
1.2.1.1	Interrupt Sequence	1-17
1.2.1.2	Programming Polling and Interrupts	1-18
1.2.2	Programmed I/O Transfers	1-25
1.2.2.1	Master Terminal Interface	1-25
1.2.3	Direct Memory Access Transfers	1-27
1.2.3.1	Enabling/Disabling DMA Transfers	1-28
1.2.3.2	Initiating DMA Transfer	1-29
1.2.3.3	Turning off Automatic Block Transfer Mode for Disc, Tape or Floppy	1-30
1.2.3.4	Turning off Automatic Block Transfer Mode for Multiplexer	1-30
2	MULTIPLEXER	2-1
2.1	INTRODUCTION	2-1
2.2	BAUD-RATE SELECTION	2-2
2.3	COMMUNICATIONS BYTE MODE	2-4
2.3.1	Programmed I/O Instructions	2-4
2.3.2	UART Asynchronous Communications	2-5
2.3.3	Command Register	2-5
2.3.3.1	Standard Mode of Operation	2-7
2.3.3.2	Operational Mode Modifications	2-7
2.3.4	Status Register	2-8
2.3.5	Software Polling	2-11

2.4	COMMUNICATIONS AUTO MODE	2-12
2.4.1	MUX I/O Control Block	2-12
2.4.2	Control Word Definitions	2-13
2.4.3	Input Operations	2-14
2.4.3.1	Input Control Word	2-14
2.4.3.2	Input Termination Status	2-19
2.4.3.3	Input Byte Pointer	2-19
2.4.3.4	Last Input Byte	2-19
2.4.4	Output Operations	2-20
2.4.4.1	Output Control Word	2-20
2.4.4.2	Output Termination Status	2-24
2.4.4.3	Output Byte Pointer	2-25
2.4.4.4	Last Output Byte	2-25
2.5	INITIALIZATION PROCEDURES	2-26
2.5.1	Setting IOCB Starting Address	2-26
2.5.2	Initialization Guidelines	2-27
2.5.3	Enabling/Disabling Ports and Port Parameters	2-28
2.5.3.1	Master Reset	2-28
2.5.3.2	Port Initialization	2-29
2.5.3.3	Enabling/Disabling Transmitters and Receivers	2-30
2.6	POLLING MUX INTERRUPT	2-32
2.7	PAUSING MUX	2-32
3	DISC DRIVE INTERFACE	3-1
3.1	INTRODUCTION	3-1
3.1.1	Performance Characteristics	3-2
3.1.2	Multi-Drive Connection	3-3
3.1.3	Disc Drive Cabling	3-4
3.1.3.1	Drive Control Bus (A Cable)	3-4
3.1.3.2	Drive Cables (B Cables)	3-9
3.2	PROGRAMMED INPUT/OUTPUT	3-11
3.2.1	Programmed I/O Input	3-11
3.2.2	Programmed I/O Output	3-13
3.2.2.1	DOA with Device Code 50 (Drive Unit Number)	3-13
3.2.2.2	DOA with Device Code 52 (Tag Pulses)	3-14
3.2.2.3	DOA with Device Code 51 (Bus Register Loading)	3-15
3.2.2.3.1	Tag 1	3-16
3.2.2.3.2	Tag 2	3-17
3.2.2.3.3	Tag 3	3-18
3.3	INPUT/OUTPUT CONTROL BLOCK (IOCB)	3-19
3.3.1	Opcode (Word 0)	3-19
3.3.2	Unit Number (Word 1)	3-21
3.3.3	Cylinder Address and Protect Flags (Word 2)	3-22
3.3.4	Head/Volume (Word 3)	3-23
3.3.5	Sector Address (Word 4)	3-24
3.3.6	Sector Count (Word 5)	3-25
3.3.7	Memory Address (Word 6)	3-26
3.3.8	Termination Status (Word 7)	3-27
3.3.9	Disc Controller Input Control	3-30
3.3.10	Disc Controller Output Control	3-31

3.4	SEEK OPERATION	3-32
3.4.1	Drive Selection	3-32
3.4.2	Seek Initiation	3-36
3.4.3	Seek Error Recovery	3-37
3.5	SECTOR BLOCK FORMAT COMMANDS	3-39
3.5.1	Formatting Operation	3-39
3.5.2	Sector Block Description	3-40
3.5.2.1	Sector Block Format - Word 0 Functions	3-40
3.5.2.2	Sector Block Format - Word 1 Functions	3-41
3.5.2.3	Sector Block Format - Word 2 Functions	3-42
3.5.2.4	Sector Block Format - Word 3 Functions	3-43
3.5.3	Formatting Procedure	3-44
3.5.4	Sector Block I/O Operations	3-46
3.5.4.1	Write Data Operation	3-46
3.5.4.2	Read Data Operation	3-46
3.5.4.3	Read-Verify Operation	3-47
3.5.4.4	Read-Regardless Operation	3-47
3.6	INTERRUPT OPERATION	3-49
4	QIC-02 STREAMING TAPE DRIVE INTERFACE	4-1
4.1	INTRODUCTION	4-1
4.1.1	Features	4-1
4.1.2	Performance Characteristics	4-2
4.2	TAPE CONTROLLER INTERFACE	4-3
4.2.1	Tape Controller Interface Signals	4-3
4.3	INPUT/OUTPUT CONTROL OPERATIONS	4-7
4.3.1	Programmed I/O Input	4-7
4.3.1.1	Status Byte Transfer	4-9
4.3.2	Programmed I/O Output	4-10
4.4	TAPE CONTROLLER IOCB	4-13
4.4.1	Tape Transfer Control	4-14
4.4.1.1	DOBP Instruction	4-14
4.4.1.2	DIBP Instruction	4-15
4.5	TAPE CONTROLLER COMMANDS	4-16
4.5.1	Select Command (Type 0)	4-17
4.5.2	Position Command (Type 1)	4-17
4.5.3	Write Command (Type 2)	4-18
4.5.4	End of Tape Handling	4-19
4.5.5	Write File Mark Command (Type 3)	4-20
4.5.6	Read Command (Type 4)	4-20
4.5.7	Read File Mark Command (Type 5)	4-21
4.5.8	Read Status Command	4-22
4.5.8.1	Exception Status Bytes	4-23
4.5.8.1.1	Exception Byte 0	4-24
4.5.8.1.2	Exception Byte 1	4-27
4.5.8.1.3	Bytes 2 and 3	4-29
4.5.8.1.4	Bytes 4 and 5	4-29
4.6	TAPE COMMAND INTERFACE	4-30
4.6.1	Command Transfer Sequence	4-30
4.6.2	Read Status Command Sequence	4-31
4.6.3	Write Command Operations	4-32
4.6.4	Read Command Operations	4-33
4.6.5	Write File Mark Operations	4-34

4.6.6	Read File Mark Operations	4-34
4.6.7	Error Processing and Recovery	4-35
4.6.7.1	Write Buffer Overrun	4-35
4.6.7.2	Read After Write Errors	4-36
4.6.7.3	Read Buffer Underruns	4-36
4.6.7.4	Read Data Errors	4-37
4.6.7.5	Read Sequence Errors	4-37
5	FLOPPY DISC DRIVE INTERFACE	5-1
5.1	INTRODUCTION	5-1
5.1.1	Floppy Disc Controller Features	5-2
5.1.2	Typical Drive Performance Characteristics	5-3
5.1.3	Diskette Specifications	5-4
5.2	FLOPPY DISC DRIVE INTERFACE	5-5
5.3	PROGRAMMED INPUT/OUTPUT	5-12
5.3.1	Select Register	5-13
5.3.2	Command register	5-14
5.3.2.1	Restore Command Bit Configuration	5-15
5.3.2.2	Seek Command Bit Configuration	5-16
5.3.3	Status Register	5-17
5.3.4	Track Register	5-21
5.3.5	Sector Register	5-21
5.3.6	Data Register	5-22
5.4	INPUT/OUTPUT CONTROL BLOCK (IOCB)	5-23
5.4.1	Command Word (Word 0)	5-23
5.4.1.1	Read Sector Command	5-24
5.4.1.2	Write Sector Command	5-27
5.4.1.3	Read Address Command	5-29
5.4.1.4	Read Track Command	5-30
5.4.1.5	Write track Command	5-31
5.4.2	Memory Address (Word 1)	5-32
5.4.3	Termination Status Word (Word 2)	5-32
5.4.4	Floppy Disc Controller Input Control	5-33
5.4.5	Floppy Disc Controller Output Control	5-34
5.5	READ/WRITE HEAD POSITIONING	5-35
5.5.1	Seek Operation	5-35
5.5.1.1	Seek Command Initiation	5-36
5.5.1.2	Seek Error Recovery	5-38
5.5.2	Restore Operation	5-38
5.6	FORMATTING DISKETTES	5-39
5.6.1	IBM-Type Formats	5-40
5.6.2	Non-IBM Formats	5-41

6	MARK 2 DISC/FLOPPY DRIVE INTERFACE	6-1
6.1	INTRODUCTION	6-1
6.2	MARK 2 INTERFACE	6-3
6.3	PROGRAMMED INPUT/OUTPUT	6-10
6.3.1	Data Register	6-11
6.3.2	Write Precomp Register	6-11
6.3.3	Error Register	6-12
6.3.4	Sector Count Register	6-15
6.3.5	Sector Number Register	6-15
6.3.6	Cylinder High and Low Registers	6-16
6.3.7	Select/Drive/Head Register	6-17
6.3.8	Command Register	6-18
6.3.8.1	Test Command Bit Configuration	6-18
6.3.8.2	Restore Command Bit Configuration	6-19
6.3.8.3	Seek Command Bit Configuration	6-20
6.3.9	Status Register	6-21
6.4	INPUT/OUTPUT CONTROL BLOCK (IOCB)	6-23
6.4.1	Opcode/Write Precomp Word (Word 0)	6-23
6.4.1.1	Read Sector Command	6-24
6.4.1.2	Readlong Command	6-25
6.4.1.3	Write Sector Command	6-25
6.4.1.4	Writelong Command	6-26
6.4.1.5	Format Track Command	6-26
6.4.2	Sector Number and Count Word (Word 1)	6-27
6.4.3	Cylinder Number Word (Word 2)	6-27
6.4.4	Memory Address Word (Word 3)	6-28
6.4.5	Termination Status Word (Word 4)	6-28

APPENDICES

A	ASCII Codes	A-1
B	Tape Handling Flowcharts	B-1
C	Floppy Disc Handling Flowcharts	C-1

FIGURES

<u>Number</u>	<u>Title</u>	<u>Page</u>
1-1	POINT 4 MARK 3 Peripheral Interface Board	1-3
1-2	POINT 4 MARK 3B Peripheral Interface Board	1-4
1-3	POINT 4 MARK 2 Peripheral Interface Board	1-5
1-4	Peripheral Interface Board System Block Diagram	1-7
1-5	MARK 3 PIB Internal Architecture	1-13
1-6	MARK 3B PIB Internal Architecture	1-14
1-7	MARK 2 PIB Internal Architecture	1-15
1-8	IOCB Pointer Format	1-18
1-9	Software Polling/Interrupt Handling Flowchart	1-20
1-10	I/O Service Routine Multiplexer Handling Procedures	1-23
1-11	I/O Service Routine Disc/Tape/Floppy Handling Procedures	1-24
2-1	Baud Rate Header Locations	2-3
2-2	Command Register	2-5
2-3	Status Register	2-8
2-4	Software Polling Flowchart for Output Operations	2-11
2-5	MUX Input/Output Control Block	2-13
2-6	Input Control Word - Word 0	2-14
2-7	Output Control Word - Word 1	2-20
2-8	Master Reset Instruction Format	2-28
2-9	Port Initialization Instruction Format	2-29
2-10	Transmitter/Receiver Instruction Format	2-30
3-1	Daisy Chain Drive Connection	3-3
3-2	PIB/Disc Drive Differential Bus Signals	3-5
3-3	PIB/Disc Drive Differential Synchronization Cable Signals	3-9
3-4	Drive Status	3-11
3-5	Contents of DOA Instruction Accumulator	3-14
3-6	Tag 1 Instruction Format	3-16
3-7	Tag 2 Instruction Format	3-17
3-8	Tag 3 Instruction Format	3-18
3-9	Disc IOCB Format	3-19
3-10	Opcode Word Format	3-20
3-11	Unit Number Word Format	3-21
3-12	Cylinder Address Word Format	3-22
3-13	Head/Volume Word Format	3-23
3-14	Sector Address Word Format	3-24
3-15	Sector Count Word Format	3-25
3-16	Memory Address Word Format	3-26
3-17	Termination Status Word Format	3-27
3-18	Seek Operation Timing	3-32
3-19	Drive Select and Seek Initiation	3-33
3-20	Seek Error Recovery Flowchart	3-38
3-21	Sector Table in Memory	3-39

3-22	Sector Block Format - Word 0 Contents	3-40
3-23	Sector Block Format - Word 1 Contents	3-41
3-24	Sector Block Format - Word 2 Contents	3-42
3-25	Sector Block Format - Word 3 Contents	3-43
3-26	Input/Output Control Block	3-44
3-27	First Sector Block	3-44
3-28	Read-Regardless Operation	3-48
4-1	Tape Controller Interface Signals	4-3
4-2	Format of DIA Instruction Accumulator	4-7
4-3	Contents of DOA Instruction Accumulator	4-10
4-4	Tape Controller IOCB Format	4-13
4-5	Tape Controller Command Format	4-16
4-6	Exception Byte 0	4-24
4-7	Exception Byte 1	4-27
5-1	Daisy-Chain Drive Connection	5-5
5-2	PIB/Floppy Disc Drive Interface Signals	5-6
5-3	Select Register Word	5-13
5-4	Restore Command Bit Configuration	5-15
5-5	Seek Command Bit Configuration	5-16
5-6	Drive Status Register Word For Data Transfer Commands	5-17
5-7	Status Register For Head Movement Commands	5-19
5-8	Floppy Disc IOCB Format	5-23
5-9	Read Sector Command Word Bit Configuration	5-25
5-10	Write Sector Command Word Bit Configuration	5-27
5-11	ID Field	5-29
5-12	Read Address Command Word Bit Configuration	5-29
5-13	Read Track Command Word Bit Configuration	5-30
5-14	Write Track Command Word Bit Configuration	5-31
5-15	Memory Address Word Bit Configuration	5-32
5-16	Termination Status Word Format	5-32
5-17	Seek Operation Flowchart	5-37
5-18	512-Byte Sector Format	5-40
6-1	WD1002-5 Disc Controller Interface Signals	6-4
6-2	Error Register Bit Configuration	6-13
6-3	Low-High Cylinder Registers and Cylinder Number	6-16
6-4	SDH Register Bit Configuration	6-17
6-5	Test Command Bit Configuration	6-18
6-6	Restore Command Bit Configuration	6-19
6-7	Seek Command Bit Configuration	6-20
6-8	Status Register Bit Configuration	6-21
6-9	ST506 Disc IOCB Format	6-23
6-10	Read Sector/Readlong Command Bit Configuration	6-24
6-11	Write Sector/Writelong Command Bit Configuration	6-25
6-12	Format Track Command Bit Configuration	6-26
6-13	Sector Number and Count Bit Configuration (Word 1 of IOCB)	6-27
6-14	Cylinder Number Bit Configuration (Word 2 of IOCB)	6-27
6-15	Memory Address Bit Configuration (Word 3 of IOCB)	6-28
6-16	Termination Status Bit Configuration (Word 4 of IOCB)	6-28

	(Tape Flowcharts)	
B-1	Command Transfer Sequence Flowchart	B-2
B-2	Read Status Sequence Flowchart	B-3
B-3	Write Sequence Flowchart	B-4
B-4	Read Sequence Flowchart	B-5
B-5	Write Transfer Sequence Flowchart	B-6
B-6	Read Transfer Sequence Flowchart	B-7
B-7	Off-line Sequence Flowchart	B-8
	(Floppy Disc Flowcharts)	
C-1	Head Movement (Type I) Flowchart	C-3
C-2	Read/Write Sector (Type II) Flowchart	C-6
C-3	Write Track (Type III) Flowchart	C-9
C-4	Read Track/Read Address (Type III) Flowchart	C-11

TABLES

<u>Number</u>	<u>Title</u>	<u>Page</u>
2-1	Baud Rate Jumpering	2-2
2-2	Port Assignments for Baud Rate Headers	2-2
2-3	DOA, DIA Device Codes	2-4
2-4	Command Register Bit Functions	2-6
2-5	Status Register Bit Operations	2-9
2-6	Input Control Word Definitions (Word 0 of Each Control Block)	2-15
2-7	Output Control Word Definitions (Word 1 of Each Control Block)	2-21
2-8	Enabling/Disabling Operation	2-31
3-1	Disc Drive Differential Bus Signals (A Cable Pin Assignment)	3-6
3-2	Drive Synchronization Cable Signals	3-10
3-3	Drive Status Definitions	3-12
3-4	DOA Instruction Accumulator Bits	3-15
3-5	Data Bus Bit Functions with Tag 1	3-16
3-6	Data Bus Bit Functions with Tag 2	3-17
3-7	Data Bus Bit Functions with Tag 3	3-18
3-8	Disc IOCB Word 0	3-20
3-9	Disc IOCB Word 1	3-21
3-10	Disc IOCB Word 2	3-22
3-11	Disc IOCB Word 3	3-23
3-12	Disc IOCB Word 4	3-24
3-13	Disc IOCB Word 5	3-25
3-14	Disc IOCB Word 6	3-26
3-15	Disc Termination Status Word 7	3-28
3-16	Sector Block Format - Word 0 Functions	3-40

3-17	Sector Block Format - Word 1 Functions	3-41
3-18	Sector Block Format - Word 2 Functions	3-42
3-19	Sector Block Format - Word 3 Functions	3-43
4-1	Tape Controller Interface Cable J1 Signal Functions	4-4
4-2	DIA Instruction Accumulator Bits	4-8
4-3	DOA Instruction Accumulator Bits	4-11
4-4	Status Byte Definitions	4-23
4-5	Exception Byte 0 Bit Definition	4-25
4-6	Exception Byte 1 Bit Definition	4-28
5-1	8-Inch Floppy Disc Controller Interface (J1 Signal Functions)	5-7
5-2	5-1/4-Inch Floppy Disc Controller Interface (J17 Signal Functions)	5-10
5-3	Drive Status Definitions For Data Transfer Commands	5-18
5-4	Drive Status Definitions For Head Movement Commands	5-20
5-5	Sector Length Flag Setting	5-24
5-6	Read Sector Command Bit Configuration	5-26
5-7	Write Sector Command Bit Configuration	5-28
5-8	Gap Sizes	5-41
6-1	ST506 Drive Control (J7) Signals	6-5
6-2	ST506 Drive Data (J1 thru J3) Signals	6-7
6-3	Floppy Disc Drive Control (J8) Signals	6-8
6-4	Diagnostic Error Codes	6-12
6-5	Error Register Bit Definitions	6-14
6-6	Status Register Bit Functions	6-22
A-1	ASCII Codes in Octal	A-1



Section 1

INTRODUCTION

1.1 GENERAL DESCRIPTION

The Peripheral Interface Board (PIB) is a controller that interfaces the POINT 4 MARK 3 CPU to various groups of peripherals via the backplane. The PIB provides the following features:

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

The PIB is available in three versions: the MARK 3 PIB, the MARK 3B PIB, and the MARK 2 PIB. The peripherals interfaced by each PIB are as follows:

1. MARK 3 PIB interfaces:
 - Asynchronous communications lines
 - Up to two disc drive units
 - QIC-02 streaming tape drive unit
2. MARK 3B PIB interfaces:
 - Asynchronous communications lines
 - Up to two disc drive units
 - Up to four floppy disc drive units
3. MARK 2 PIB interfaces:
 - Asynchronous communications lines
 - One ST506 5-1/4-inch disc drive unit
 - QIC-02 streaming tape drive unit
 - 5-1/4-inch floppy disc drive unit

The boards are easily distinguished because the layout of connectors and crystals is different for each board.

The MARK 3 PIB has the following distinguishing features:

- MARK 3 PIB label at the lower right corner of the board
- One crystal at the upper right
- A 50-pin connector at location J1

The MARK 3 PIB layout is shown in Figure 1-1.

The MARK 3B PIB can be recognized by the following features:

- MARK 3B PIB label at the lower left corner of the board
- Two crystals along the right side of the board
- A 40-pin chip above location J17
- 50-pin connector at location J1 for 8-inch floppy disc drives
- 34-pin connector at location J17 for 5-1/4-inch floppy disc drives

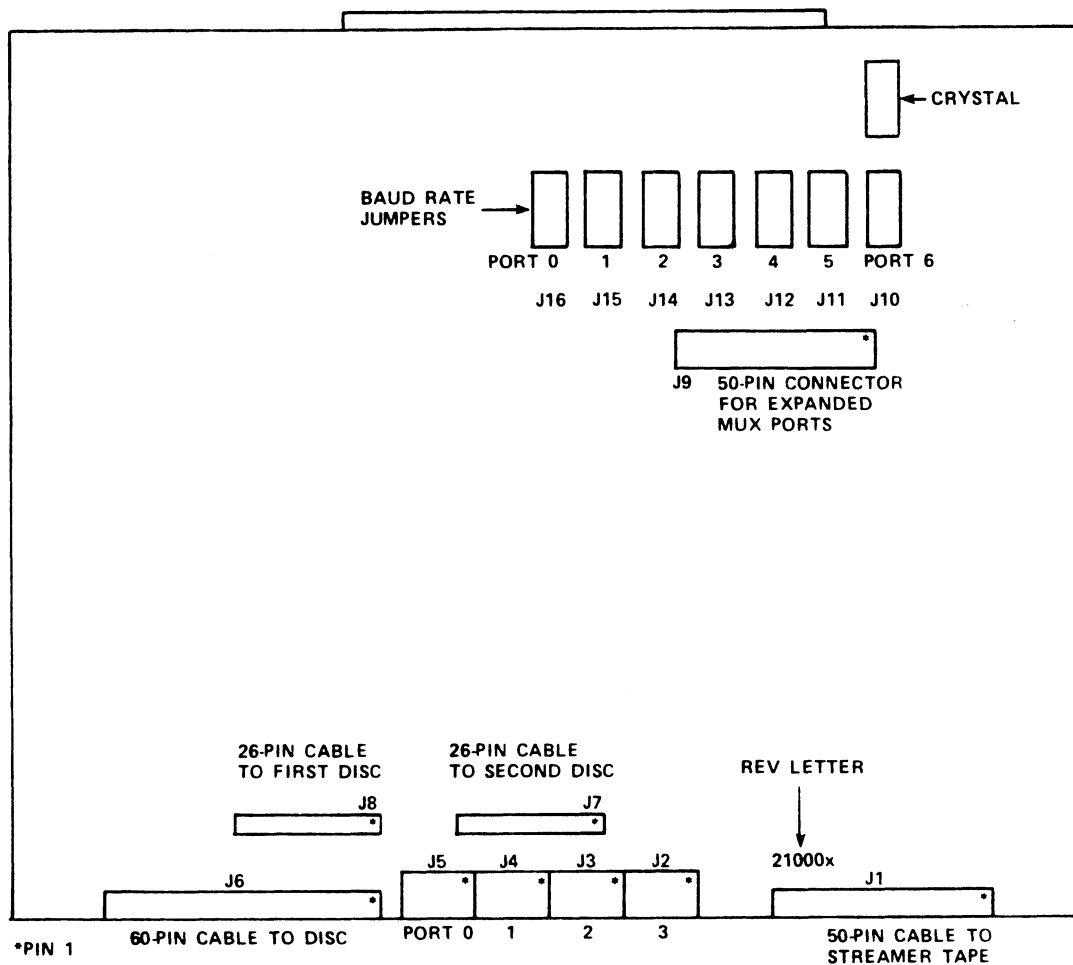
The MARK 3B PIB is set up for 8-inch and 5-1/4-inch floppy disc drives. Because of a difference in cabling and timing requirements, only one size of floppy disc drive can be used at one time.

The layout of a MARK 3B PIB is shown in Figure 1-2.

The MARK 2 PIB has the following distinguishing features:

- MARK 2 PIB label at the lower right corner of the board
- WD1002-5 controller mounted on the left side of the board

The layout of a MARK 2 PIB is shown in Figure 1-3.

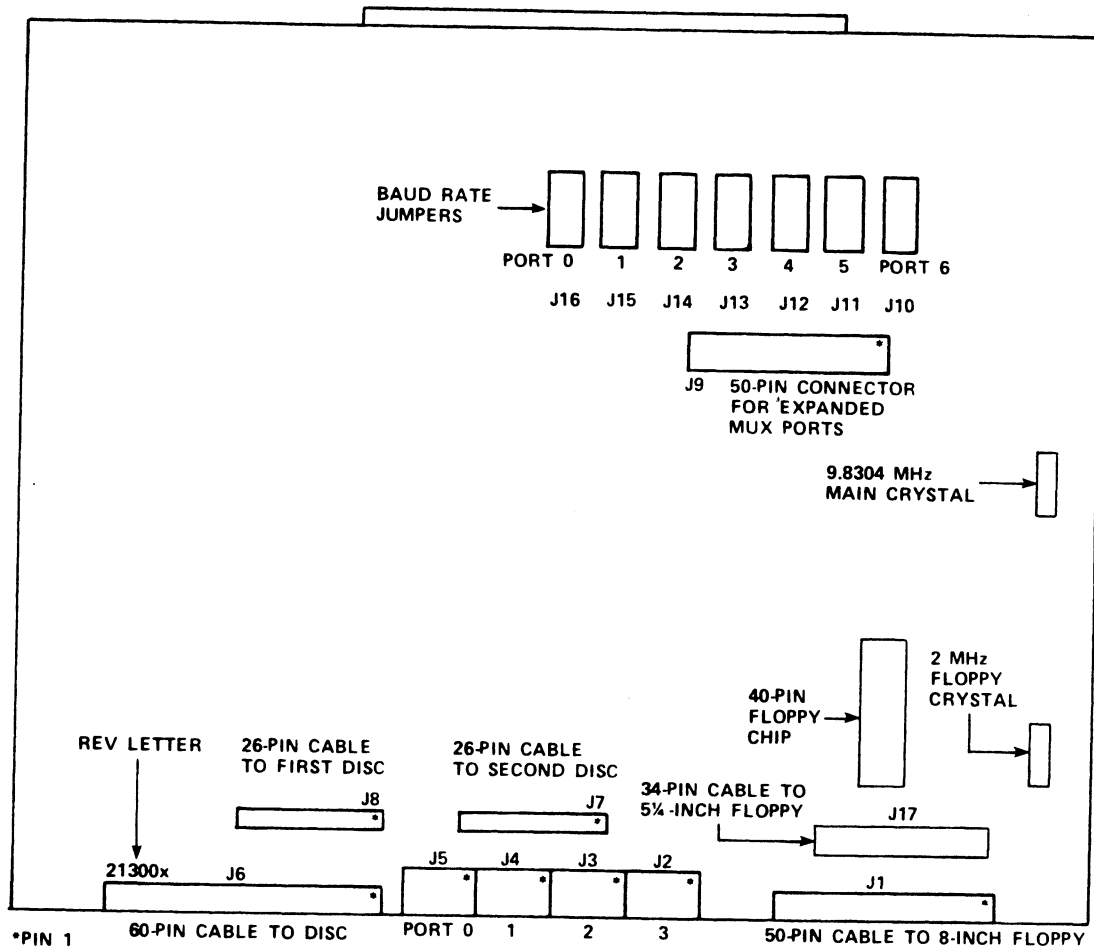


EXTERNAL CABLES WHICH PLUG INTO THE PIB BOARD:

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

081-8

Figure 1-1. POINT 4 MARK 3 Peripheral Interface Board



EXTERNAL CABLES WHICH PLUG INTO THE PIB BOARD:

8-INCH FLOPPY	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
5 1/4-INCH FLOPPY CABLE	J17

081-9

Figure 1-2. POINT 4 MARK 3B Peripheral Interface Board

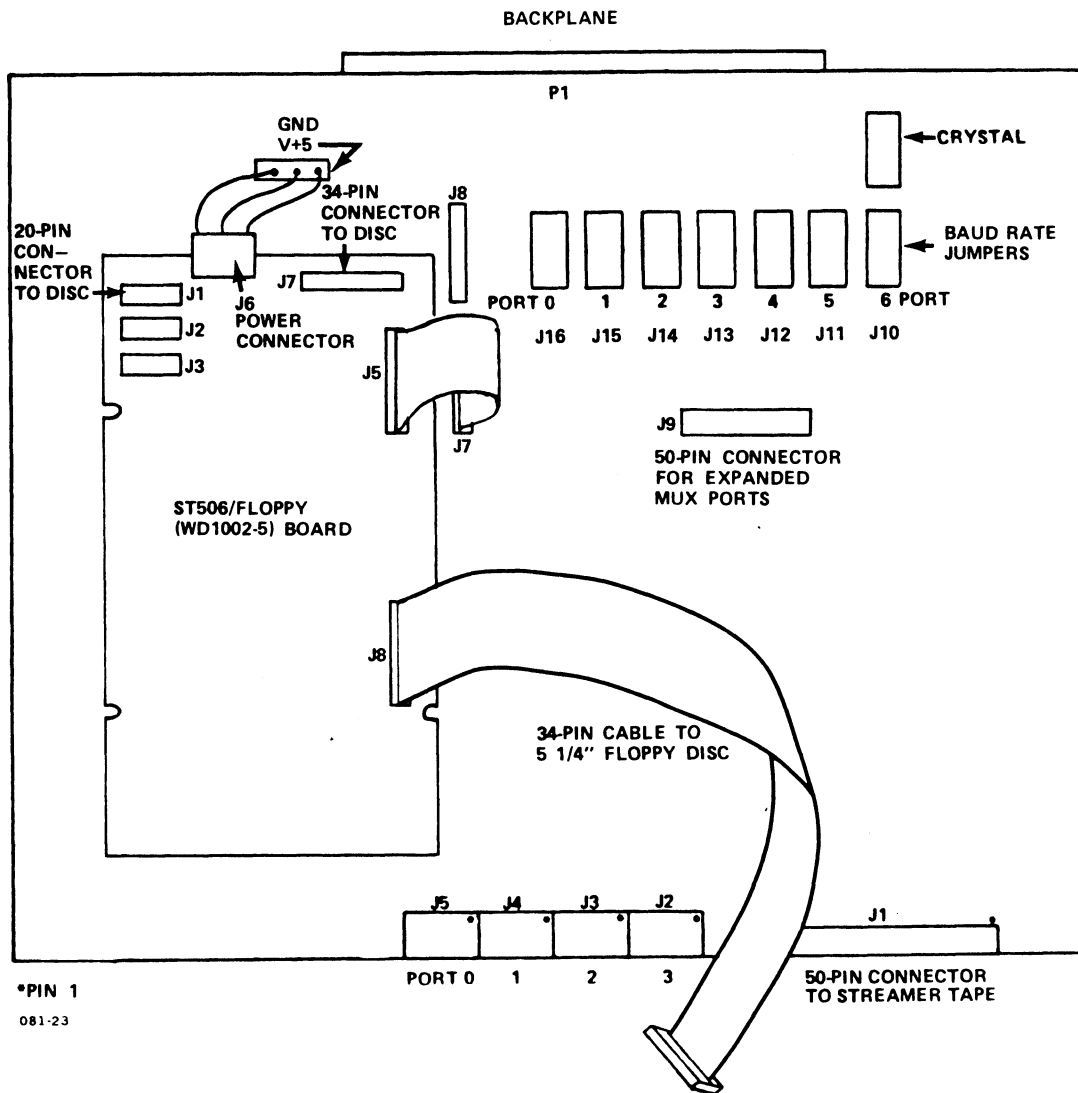


Figure 1-3. POINT 4 MARK 2 Peripheral Interface Board

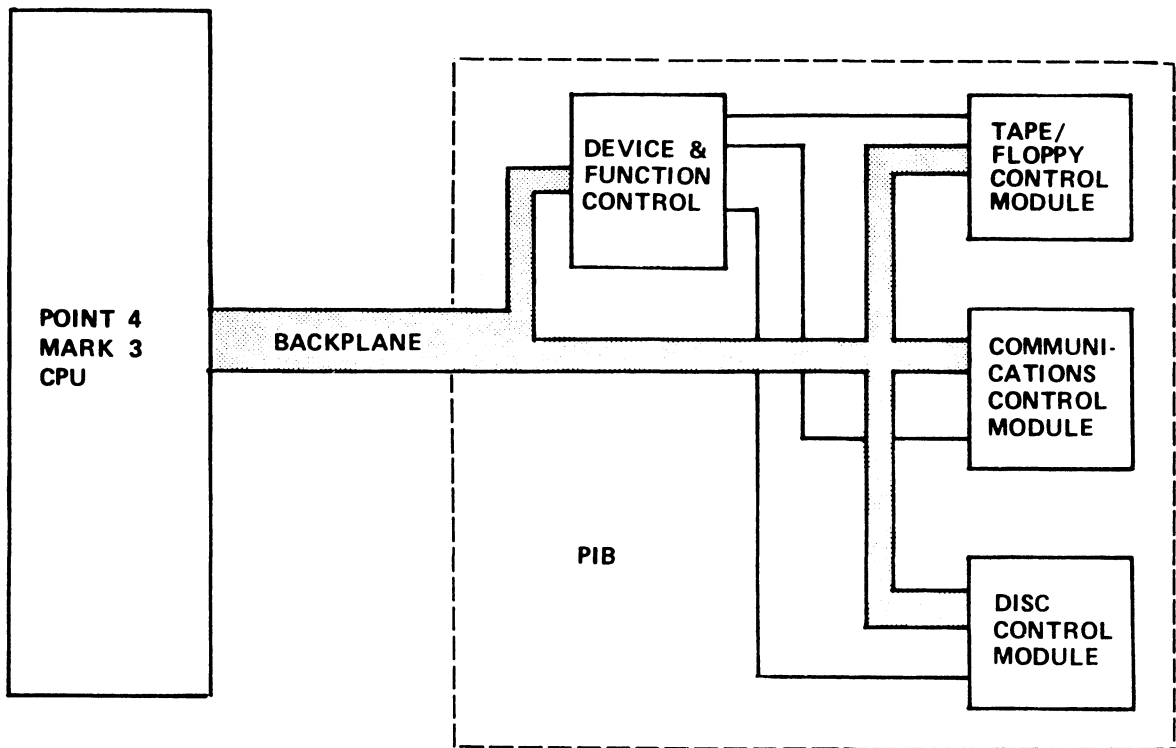
1.1.1 SYSTEM DESCRIPTION

A 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.

Each PIB contains three main control modules:

- Asynchronous Communications Control Module (see Section 2)
- CMD/SMD/LMD Disc Drive Control Module (see Section 3)
or
ST506 Disc Drive Control Module (see Section 6)
- Tape Drive Control Module (see Section 4)
or
Floppy Disc Drive Control Module (see Section 5)

Figure 1-4 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.



081-24

Figure 1-4. 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.

Cables are normally supplied by the drive vendor. POINT 4 does not supply the Mux cables. For Mux cabling specification and installation, refer to the POINT 4 MARK 3 Computer System Manual.

1.1.2.1 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

Serial Line Printer:
May be connected to any port

Baud Rates (hardware strappable):
110, 150, 300, 600, 1200, 2400, 4800, 9600

1.1.2.2 Disc Drive Interface on MARK 3 and MARK 3B PIBs

Drives per Controller:
Two

Drive Type:
SMD/CMD/LMD

DMA Transfer Rate:
1.25 megabytes per second

Sector Size:
Header - 8 bytes
Data - 512 bytes

1.1.2.3 Disc Drive Interface on a MARK 2 PIB

Drives per Controller:
One (limited by packaging only)

Drive Type:
ST506 drive

DMA Transfer Rate:
3 microseconds per word (approximate)

Sector Size:
Data - 512 bytes

1.1.2.4 Tape Drive Interface on MARK 3 and MARK 2 PIBs

Drives per Controller:
MARK 3 PIB - One
MARK 2 PIB - One

Drive Type:
QIC-02 Streaming Tape

Tape Data Rate:
90 kilobytes/second

Tape Speed:
90 inches/second

Tape Capacity:
20 or 45 megabytes

1.1.2.5 Floppy Disc Drive Interface on a MARK 3B PIB

Drives per Controller:
Four

Drive Type:
Shugart-compatible (soft-sectored) drives supporting single and double-sided, single and double-density diskettes with head-load-on-select jumper. The drives can be either

- 8-inch Floppy Disc Drive
- 5-1/4-inch Floppy Disc Drive

DMA Transfer Rate:
Single Density = 250 kilobits/second
Double Density = 500 kilobits/second

Rotational Speed:
360 RPM

Diskette Capacity:
Depends on size of diskette, format, and sector size.

For example, an 8-inch diskette, double sided, double density, with 512-byte sectors (required for IRIS) = 1.24 megabytes

Formats Supported:
IBM 1 Frequency Modulation (FM)
IBM 2 Modulated Frequency Modulation (MFM)
IBM 2D MFM

1.1.2.6 Floppy Disc Drive Interface on a MARK 2 PIB

Drives per Controller:
Four (currently one per MARK 2 system)

Drive Type:
5-1/4-inch 460-type supporting 96 tracks per inch, double-sided, double-density diskettes

DMA Transfer Rate:
250 kilobits/second

Diskette Capacity:
640 kilobytes formatted

Format:
MFM

NOTE

Diskettes should be 5-1/4-inch, 96TPI-certified, double-sided, double-density.

1.1.2.7 Input/Output Instructions

Input - DIA
Output - DOA

Device Codes:

10-17 - MUX Ports 0-3
20-25 - Expansion Ports 4-6
30-37 - Floppy Disc 0-3
50-55 - SMD Disc 0&1
50-57 - WD1002-5
60-62 - Tape 0-3

DMA (Device Code 77):

DOBS/DIBS - MUX
DOBC/DIBC - Disc
DOBP/DIBP - Tape or Floppy

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

1.1.3 PIB INTERNAL ARCHITECTURE

Four types of control functions are performed by a PIB:

- General Control Functions
- Multiplexer Control Functions
- Disc Control Functions
- Tape or Floppy Disc Control Functions

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

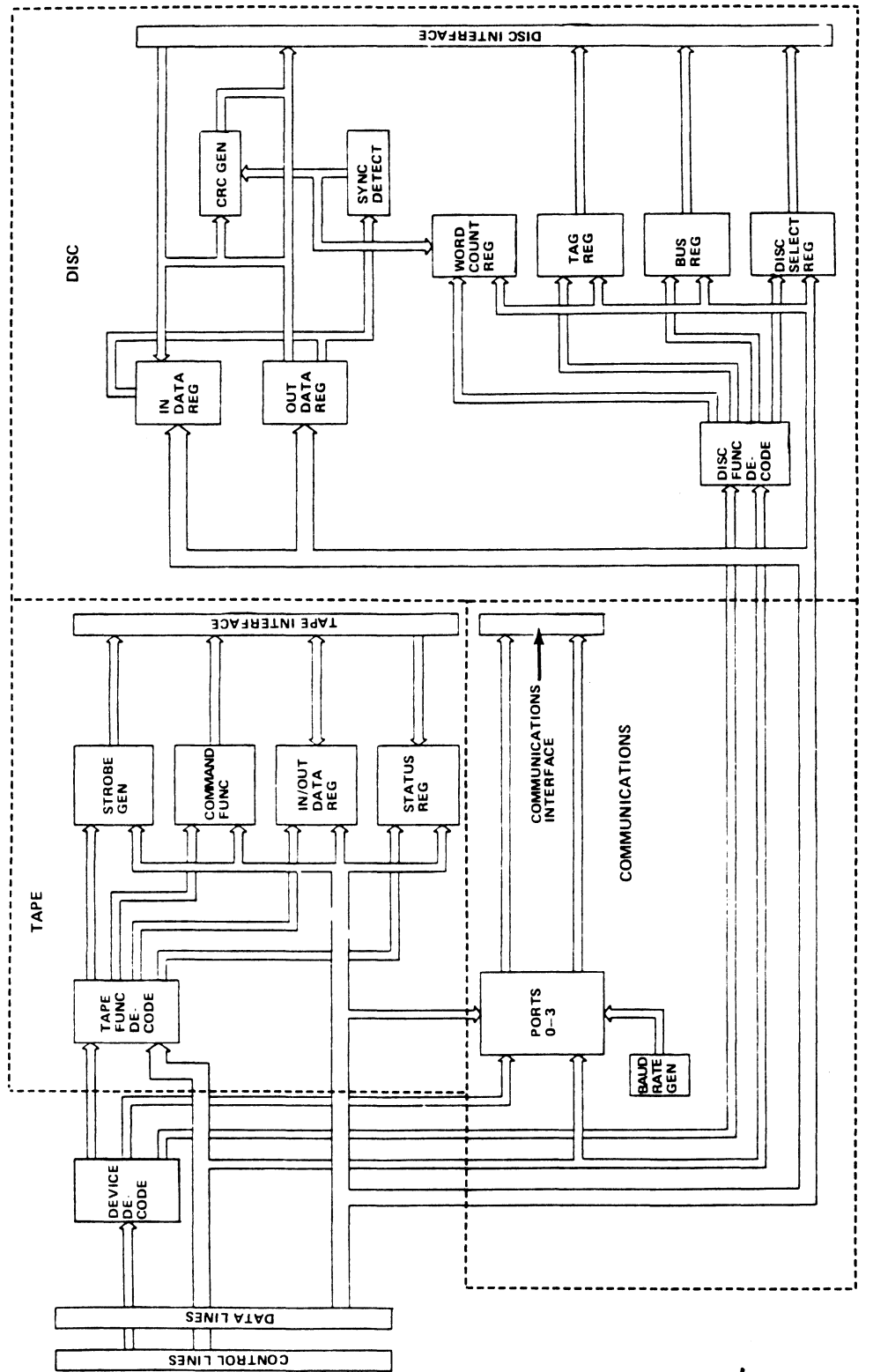
Multiplexer Control Functions include jumper-selectable baud rate, data transfer synchronization, port selection, data format manipulation, and the real-time clock.

Disc Control Functions include disc selection, bus control for DMA transfer, word count, CRC generation, tag line control (MARK 3 configurations only), and ECC (MARK 2 only).

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

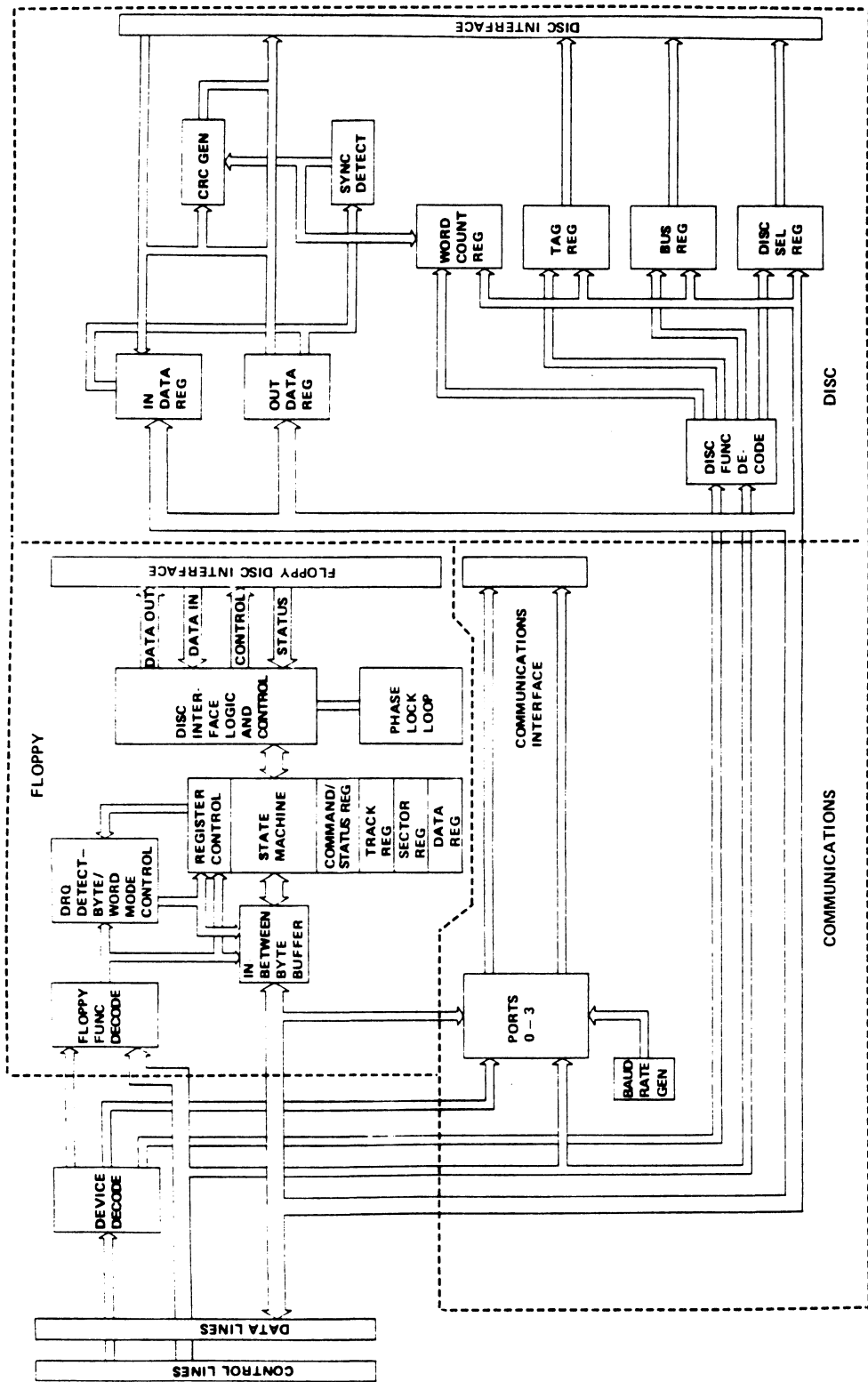
Floppy Disc Control Functions include side and unit selection, bus control for DMA transfer, CRC generation and checking, and data transfer control.

Figure 1-5 shows the internal architecture of a MARK 3 PIB, Figure 1-6 a MARK 3B PIB, and Figure 1-7 a MARK 2 PIB.



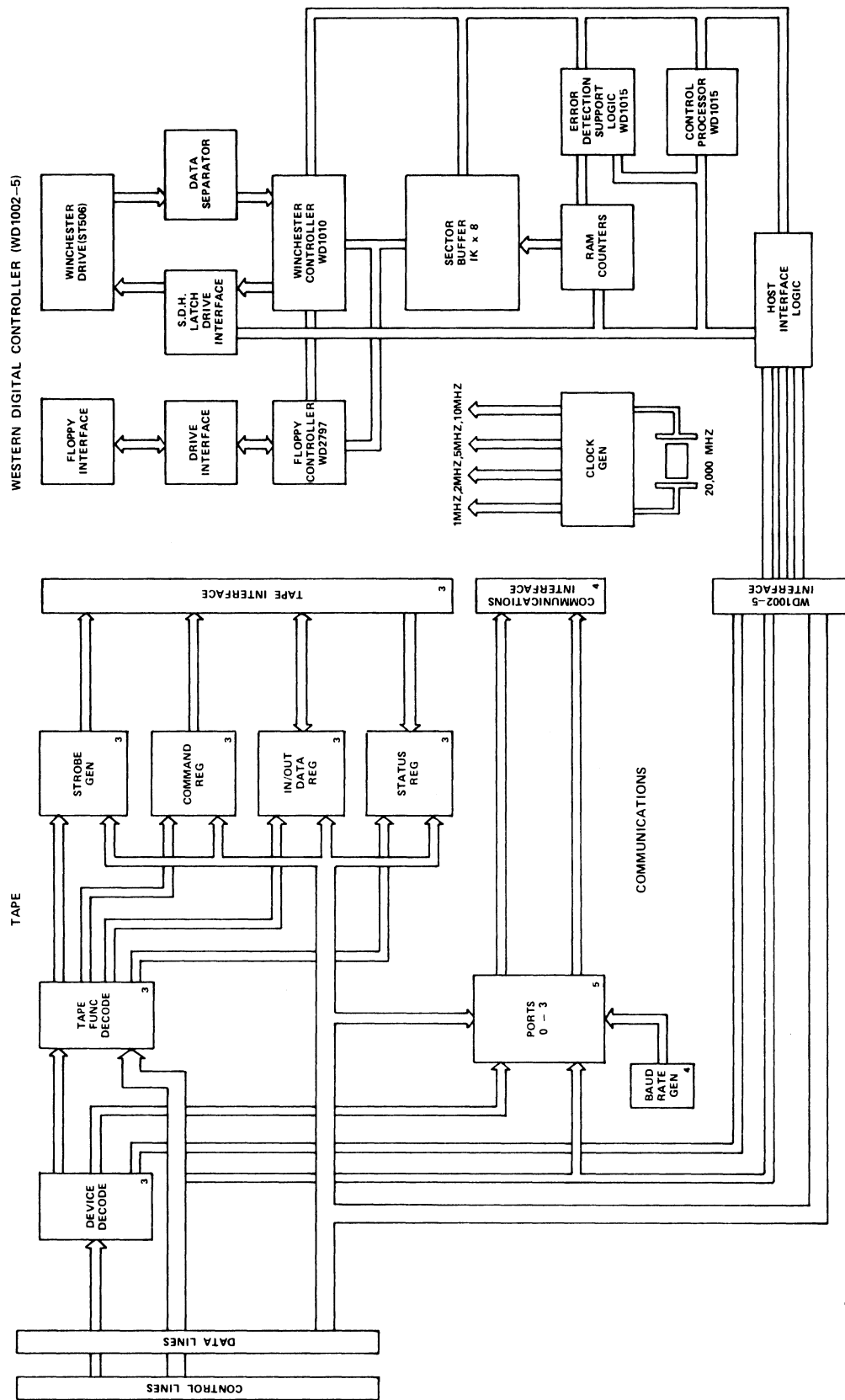
081-25

Figure 1-5. MARK 3 PIB Internal Architecture



081-27

Figure 1-6. MARK 3B PIB Internal Architecture



081-26

Figure 1-7. MARK 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 value of the accumulator can be interpreted as follows:

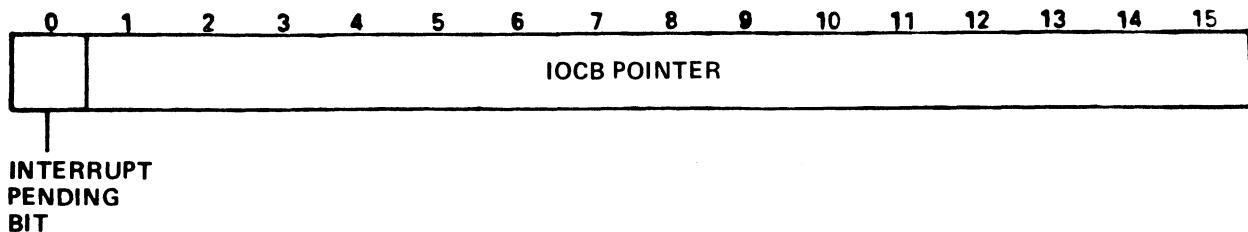
Accumulator Value	Function
Odd ($\neq 1$)	Masks out (disables) all interrupts
Even ($\neq 0$)	Masks in (enables if ION=1) all interrupts
= 1	Disables DMA to controllers
= 0	Enables DMA to controllers

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) as shown in Figure 1-8. Because the MSB of the address is used as the interrupt-pending bit, IOCBs must be in the lower 32K of memory on MARK 3 configurations. For a MARK 2, the MSB of the IOCB address is not used for interrupt-pending. Therefore, the full 16 bits can be used for addressing.

The DIB instruction with a device code of 77 (CPU) is used to read the IOCB pointers (registers 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 or floppy 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 or Floppy pointer



081-28

Figure 1-8. IOCB Pointer Format

For MARK 3 configurations, 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 either when 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.

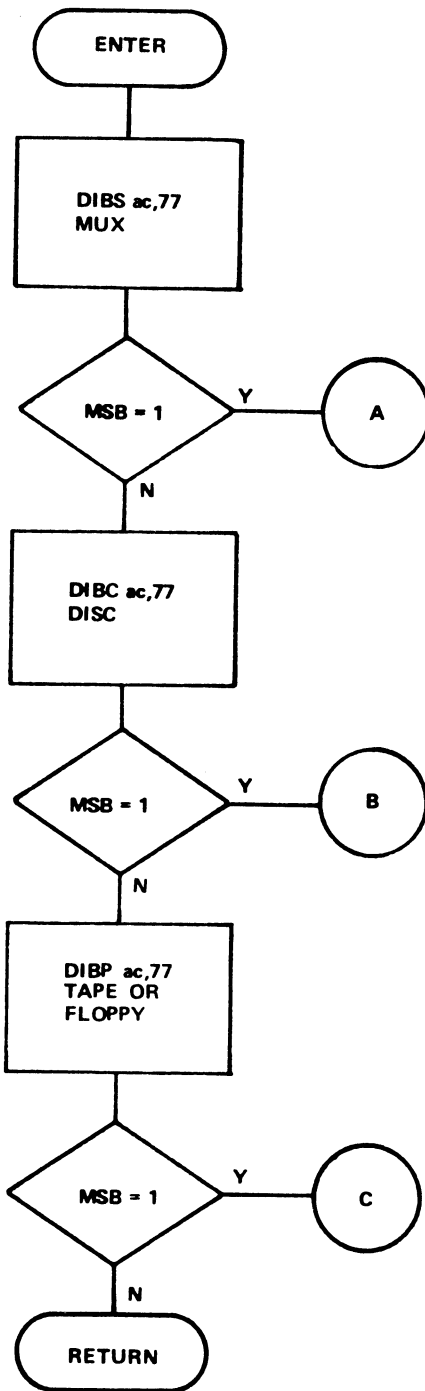
Resetting the interrupt-pending bit at that time ensures the detection of a second interrupt-pending condition while the first one is being serviced.

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 interrupt-pending flag for all 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 multiplexer IOCBs (see Section 2.4) to determine which port needs service (bit 0 set to 1). More than one port may be 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.

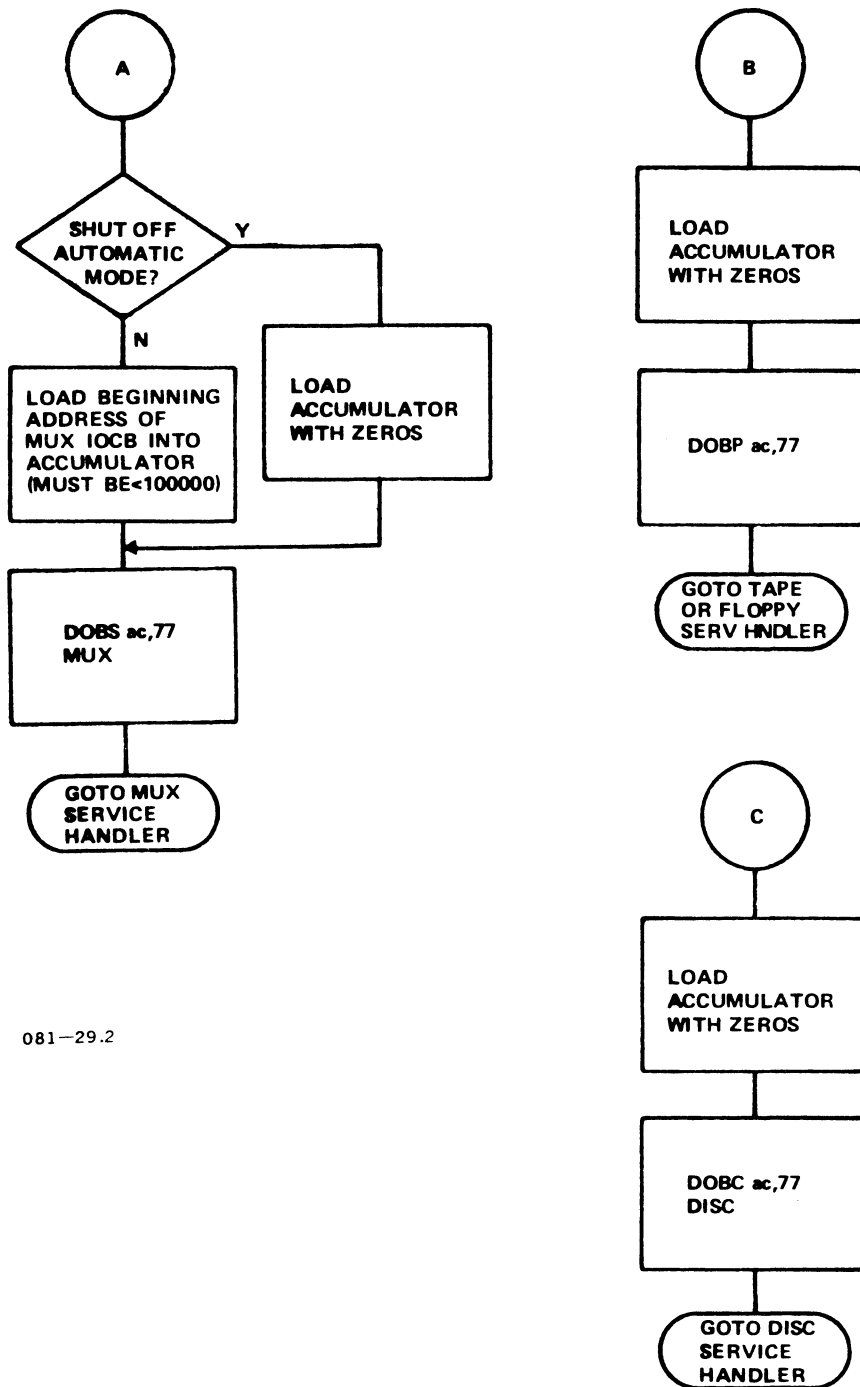
For an SMD configuration, in addition to an operation-complete bit (bit 0, word 7), the disc controller IOCB also contains a summary-error bit in the termination status (bit 1, word 7) which is used to flag the existence of an error condition in disc transfer. Like the operation-complete bit, the summary-error 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 1-9 is a flowchart showing the sequence of steps that must be taken into account when programming polling and interrupt processing procedures for a MARK 3 configuration. Refer to Section 6 for the MARK 2 procedure. Note that interrupts other than those caused by the Peripheral Interface Board (i.e., power-fail interrupt) are not covered in this flowchart. Figure 1-9 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.



081-29.1

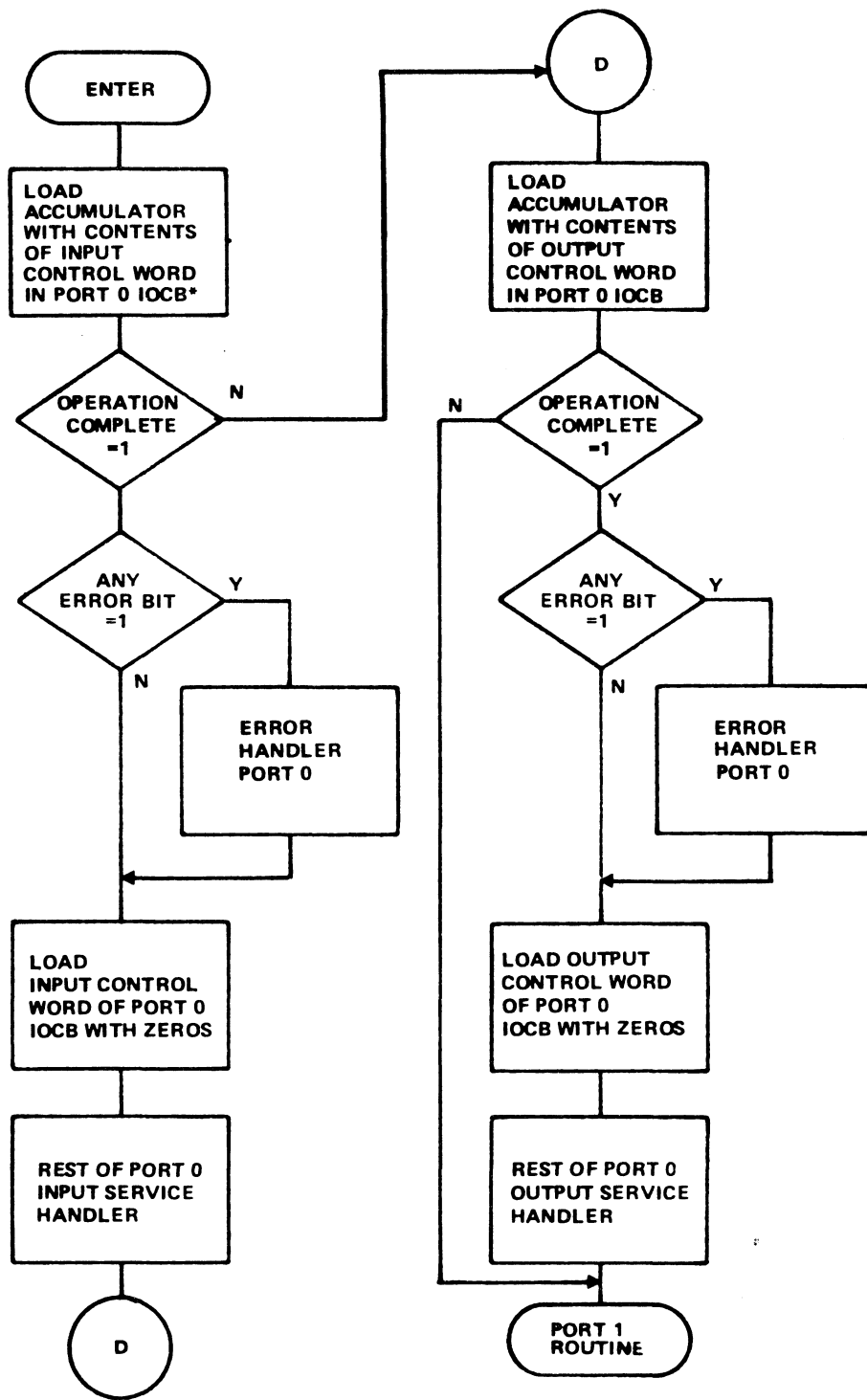
Figure 1-9. Software Polling/Interrupt Handling Flowchart (1 of 2)



081-29.2

Figure 1-9. Software Polling/Interrupt Handling Flowchart (2 of 2)

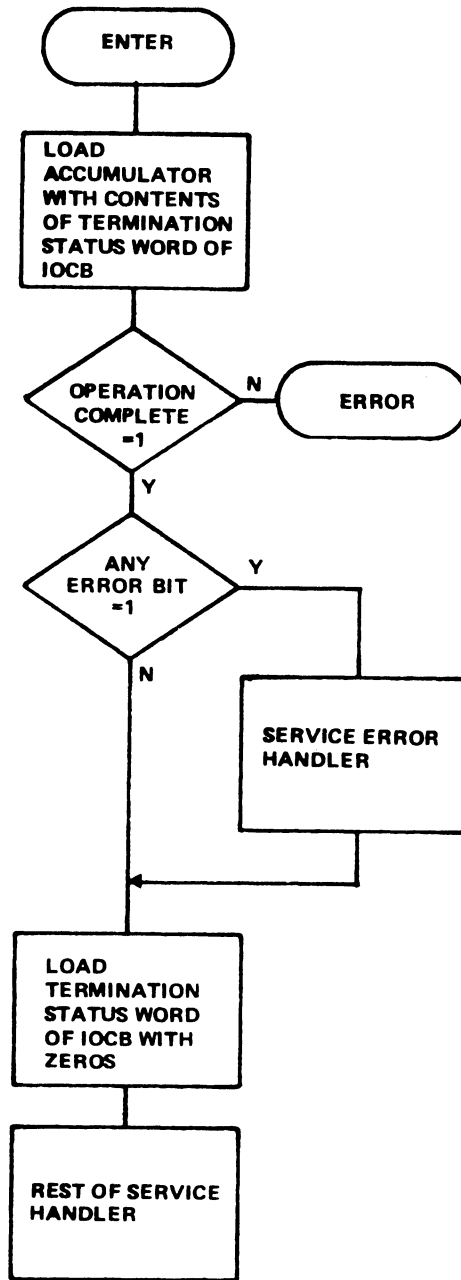
| Figures 1-10 and 1-11 are flowcharts of parts of the I/O service routines for the multiplexer and for disc, tape or floppy. The part of the service routines shown only takes into account testing of operation-complete and various error bits. The remainder of the service routine is determined by the programmer.



*PORTS 1, 2, AND 3 ARE HANDLED IN SAME MANNER AS PORT 0

081-30

Figure 1-10. I/O Service Routine Multiplexer Handling Procedures



081-31

Figure 1-11. I/O Service Routine Disc/Tape/Floppy 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, floppy disc or disc drives) 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:

<u>Accumulator Value</u>	<u>Function</u>
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:

<u>Instruction</u>	<u>Accumulator Value</u>	<u>Function</u>
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)

NOTE

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 streaming tape come to a stop due to data underrun. 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 or Floppy 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:

<u>Accumulator</u>	<u>Device Controller IOCB</u>
13	Multiplexer
14	Tape or Floppy
15	Disc

If an IOCB pointer contains a value of 0, the device controller is not being used in the system. On MARK 3 configurations, the most significant bit (MSB) of this accumulator is set to 1 when an interrupt is pending. For DMA transfers on MARK 3 configurations, the MSB must be reset to 0 after interrupt processing to avoid erroneous interrupt processing. Refer to Section 6 for information on the MARK 2.

When issued to the disc, floppy, 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. The following is an example of a MARK 3 SMD disc DMA operation:

IOCB Address: 075000

Instruction: DOBC 1,77

Accumulator 1 Value:

A1 = 0111101000000000

Memory address 75007 (termination word):

Must be set to 0

(Reserved for termination code)

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 each port on which input or output is to be enabled. 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, Tape or Floppy

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

Instruction:
DOBC 1,77 for disc
DOBP 1,77 for tape or floppy

Accumulator 1 Value: A1 = 0

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

NOTE

Except on a MARK 2, 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 BAUD-RATE SELECTION

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

110	1200
150	2400
300	4800
600	9600

Header jumpering for each baud rate is given in Table 2-1.

Header-port correspondence is given in Table 2-2.

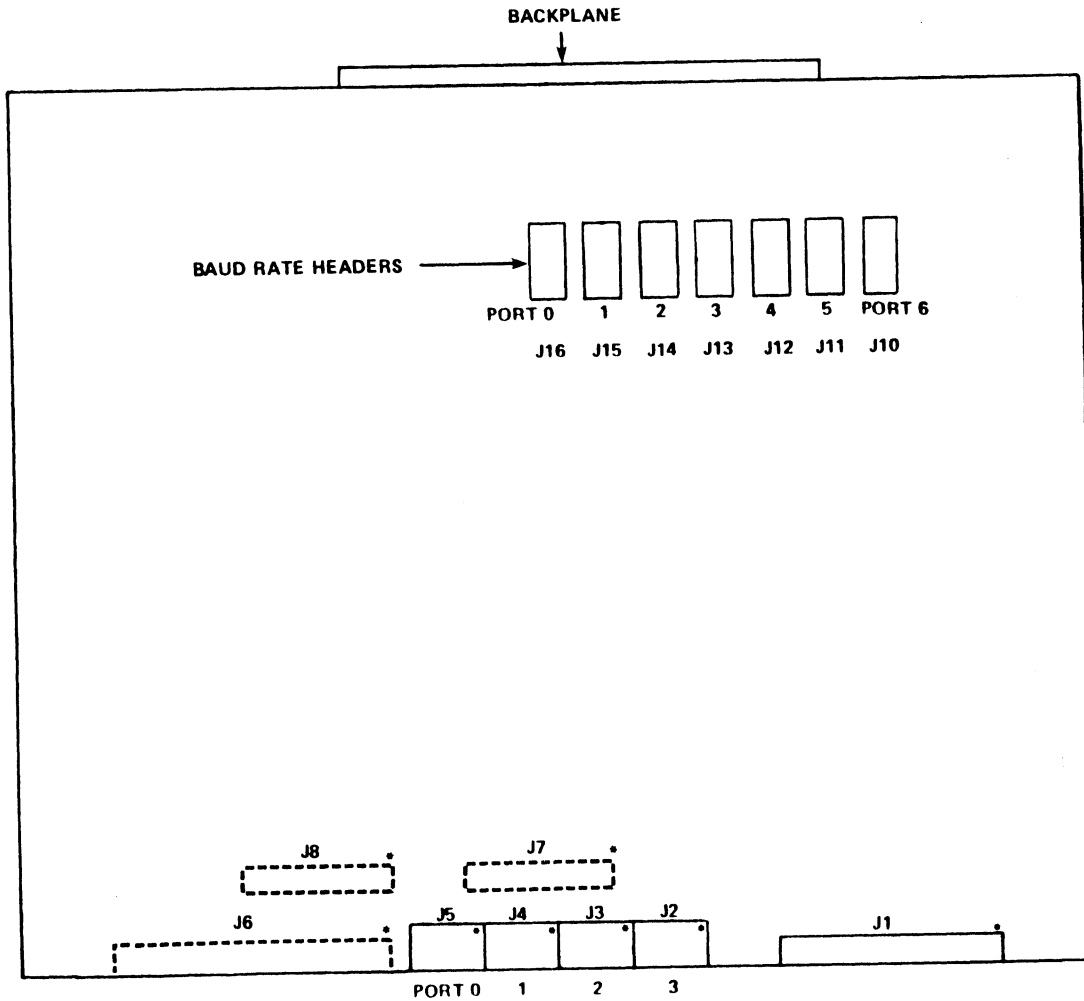
Locations of the baud rate headers for jumpering (which are the same for both PIBs) are shown in Figure 2-1.

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		



*PIN 1

081-32

Figure 2-1. Baud Rate Header 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)
10 11	PORT 0 Status Register Data Register	Command Register Data Register
12 13	PORT 1 Status Register Data Register	Command Register Data Register
14 15	PORT 2 Status Register Data Register	Command Register Data Register
16 17	PORT 3 Status Register Data Register	Command Register Data Register
20 21	PORT 4 (optional) Status Register Data Register	Command Register Data Register
22 23	PORT 5 (optional) Status Register Data Register	Command Register Data Register
24 25	PORT 6 (optional) Status Register Data Register	Command 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.

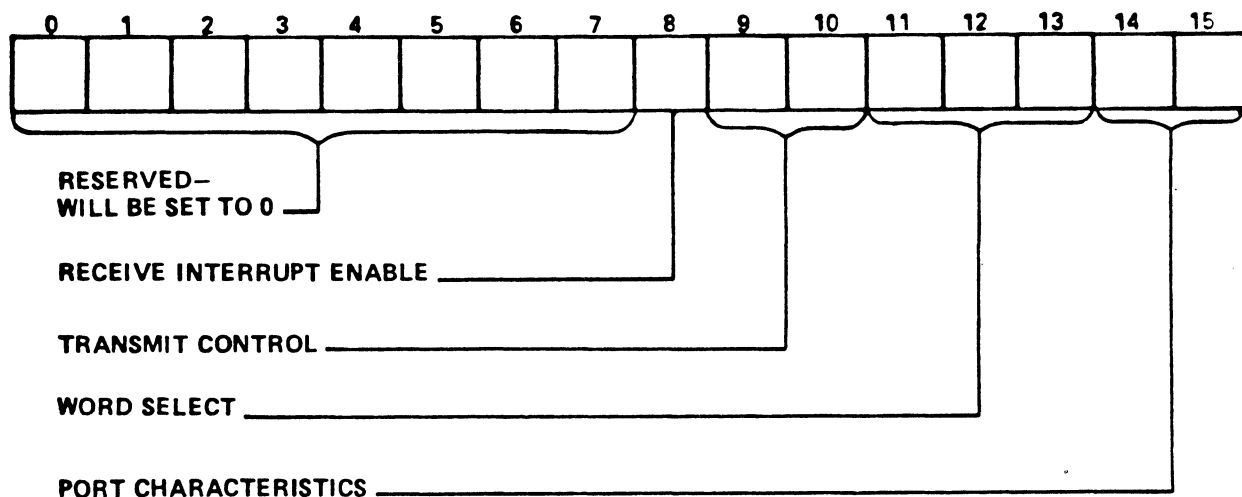


Figure 2-2. Command Register

TABLE 2-4. COMMAND REGISTER BIT FUNCTIONS

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

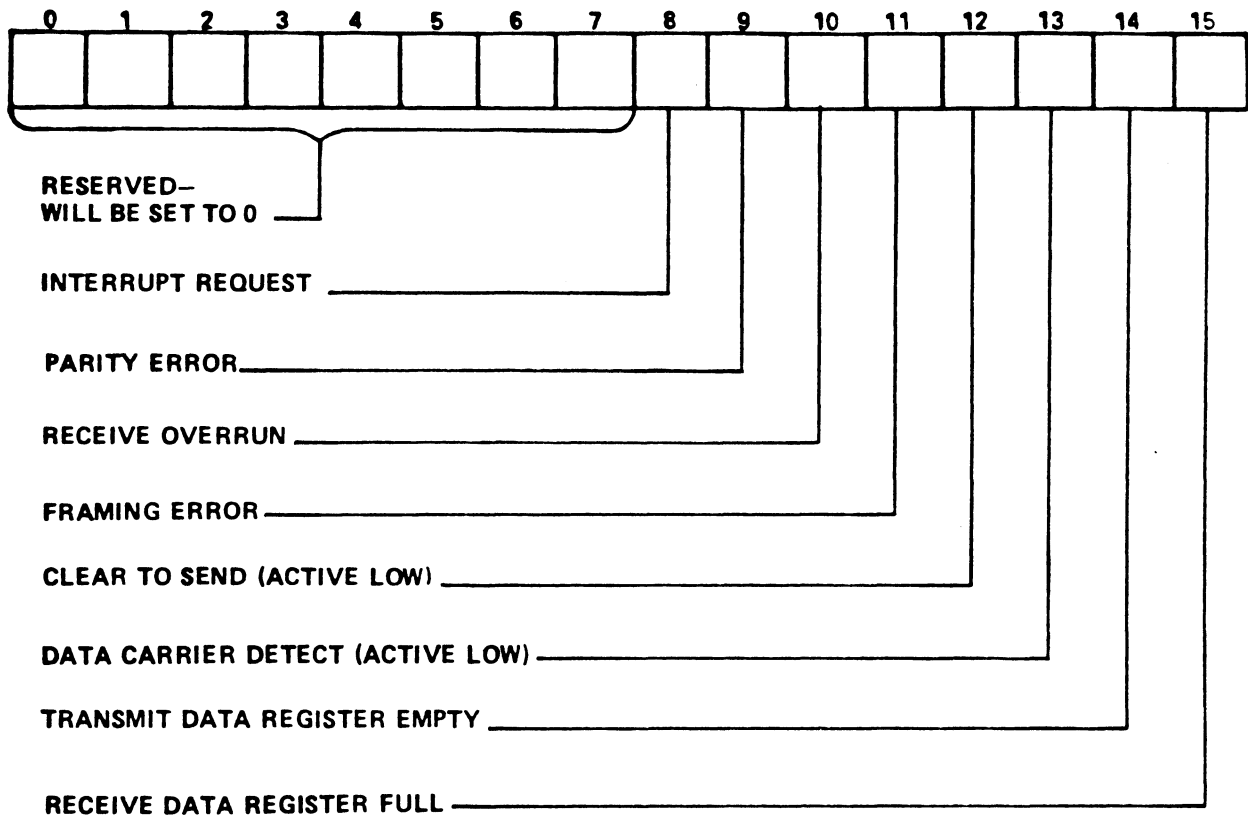


Figure 2-3. Status Register

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-)	<p>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.</p> <p>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.</p>
14	Transmit Data Register Empty (TDRE)	<p>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.</p>
15	Receive Data Register Full (RDRF)	<p>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.</p>

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.

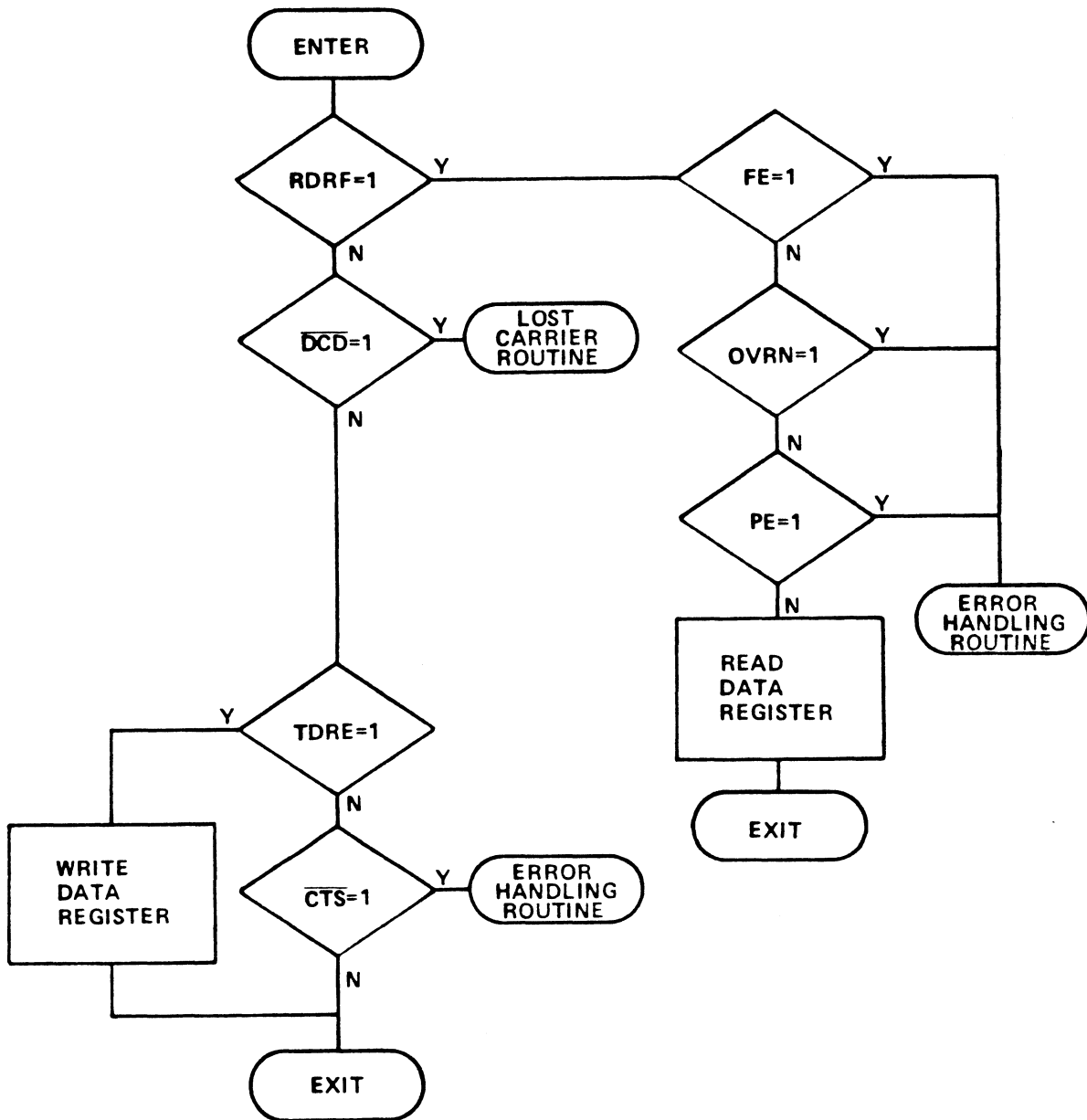


Figure 2-4. Software Polling Flowchart For Output Operations

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.

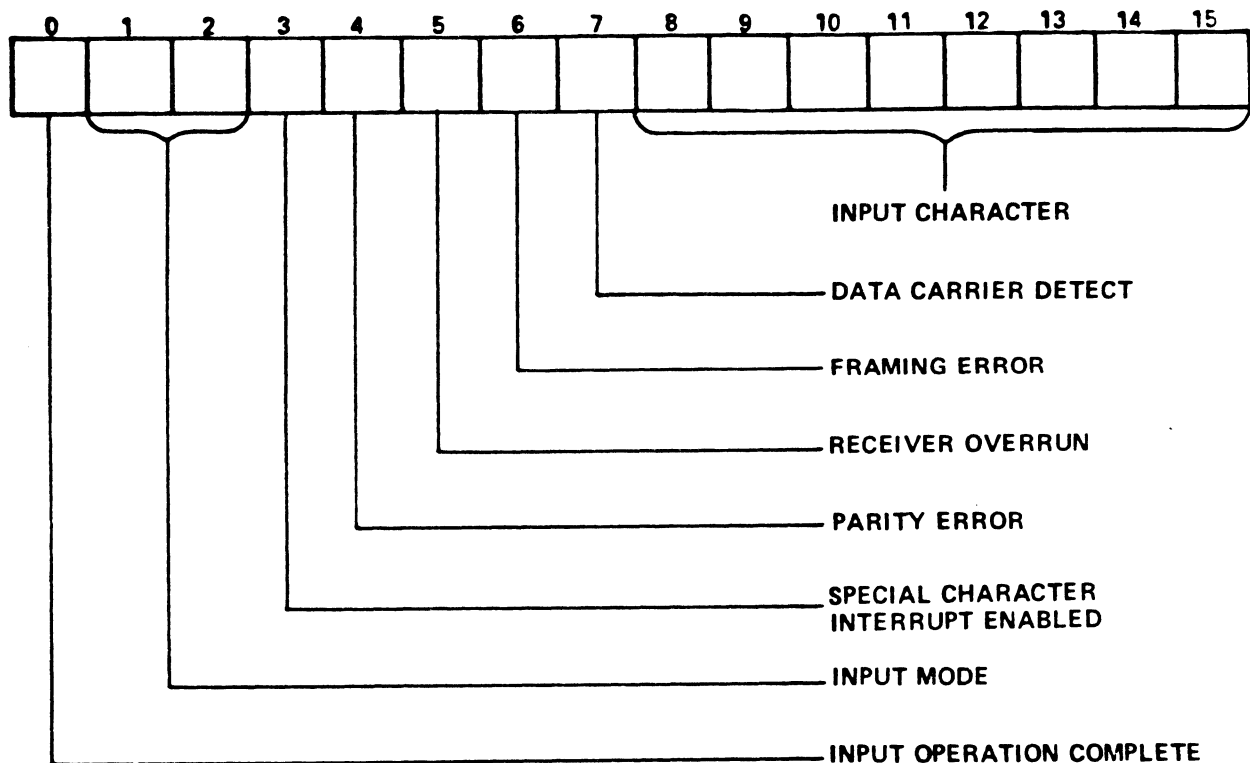


Figure 2-6. Input Control Word - Word 0

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

Bit	Instruction	Operation
0	IOC (Input Operation Complete)	<p>Set to one (1) if any of the following conditions occurs:</p> <ol style="list-style-type: none"> 1. Input buffer is full 2. The receiver detects: <ol style="list-style-type: none"> 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. <p>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.</p> <p>If another input character is received before the IOC is cleared, the original character will be overwritten in bits 8-15.</p>

**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:
<u>Value</u>		
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.
		<p align="center">NOTE</p> <p>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.</p>

**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	<p>Each incoming character is stored in bits 8-15. Characters that cause the IOC bit to be set are:</p> <ol style="list-style-type: none"> 1. The last character stored in input buffer, causing an input buffer-full condition. <p style="text-align: center;">EXCEPTION</p> <p style="text-align: center;">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.</p> <ol style="list-style-type: none"> 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 $IBP \geq LIB$, 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.

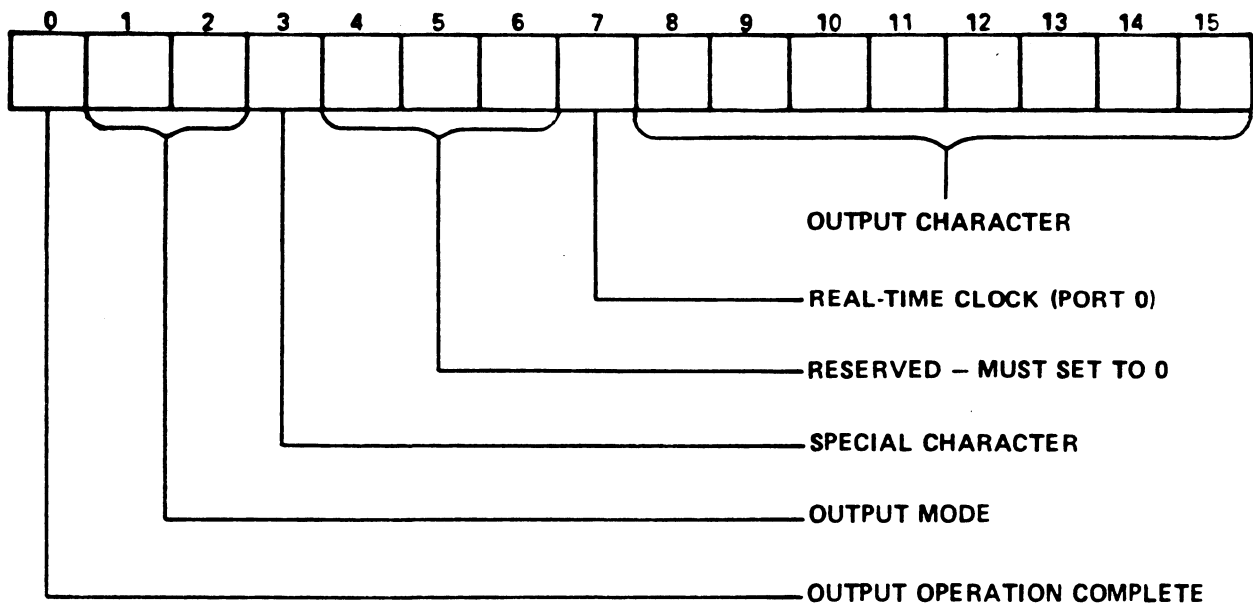


Figure 2-7. Output Control Word - Word 1

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

Bit	Instruction	Operation
0	OOC (Output Operation Complete)	<p>Set to one (1) if the operation has been terminated due to one of the following conditions:</p> <ol style="list-style-type: none"> 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. <p>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.</p>

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

Bit	Instruction	Operation
1 & 2 <u>Value</u>	Output Mode	Written by the program and read by MUX to determine the type of output.
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	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.
3		
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	<p>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.)</p> <p>The interrupt-pending bit is set, and an interrupt request is generated. Does not set the OOC bit, or terminate automatic operation.</p>
8-15	Output Character	<p>In single-character mode, this byte is sent to the port for transmission.</p> <p>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:</p> <ol style="list-style-type: none"> 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:

<u>Memory Location</u>	<u>Port</u>
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.

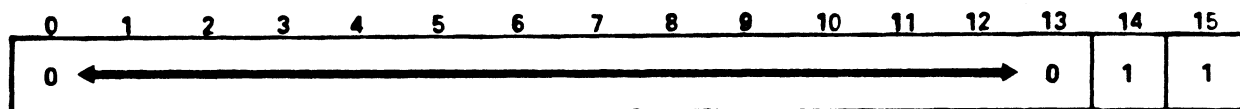


Figure 2-8. Master Reset Instruction Format

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 2-4).

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

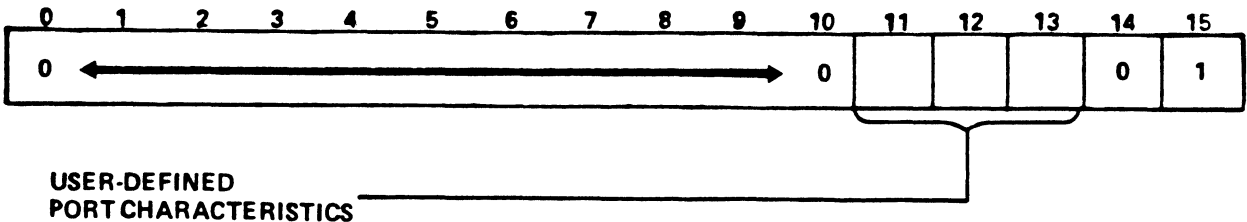


Figure 2-9. Port Initialization Instruction Format

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.

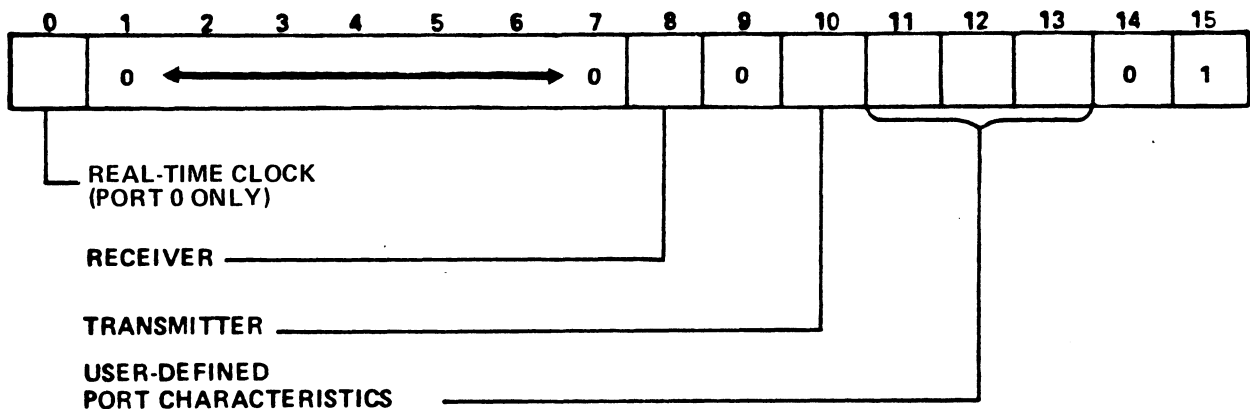


Figure 2-10. Transmitter/Receiver Instruction Format

Section 3

DISC DRIVE INTERFACE

3.1 INTRODUCTION

This section describes the interface for the SMD/CMD/LMD disc controller which is available on the MARK 3 and MARK 3B versions of the Peripheral Interface Board (PIB).

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 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

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.

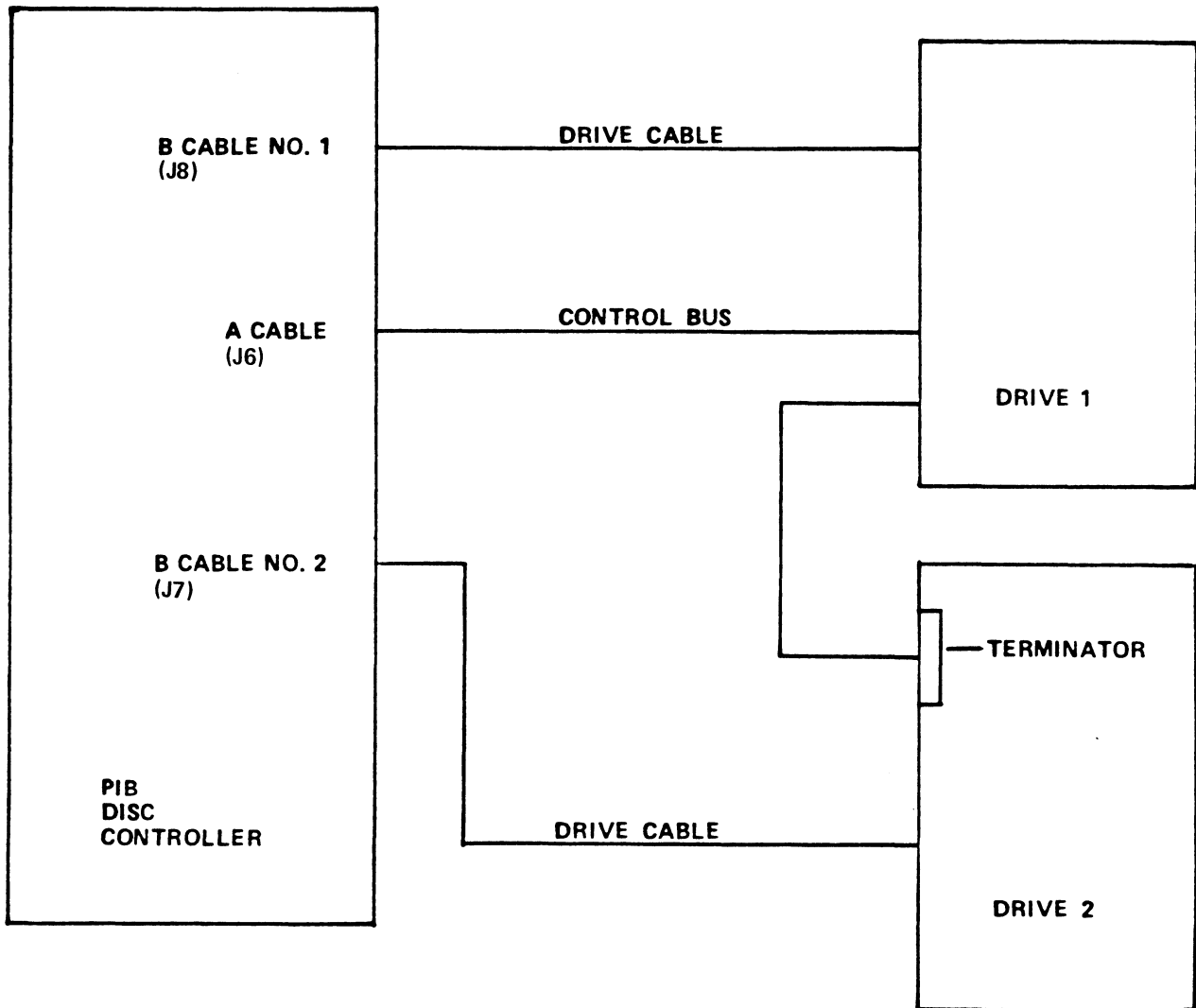


Figure 3-1. Daisy-Chain Drive Connection

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 J6 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.

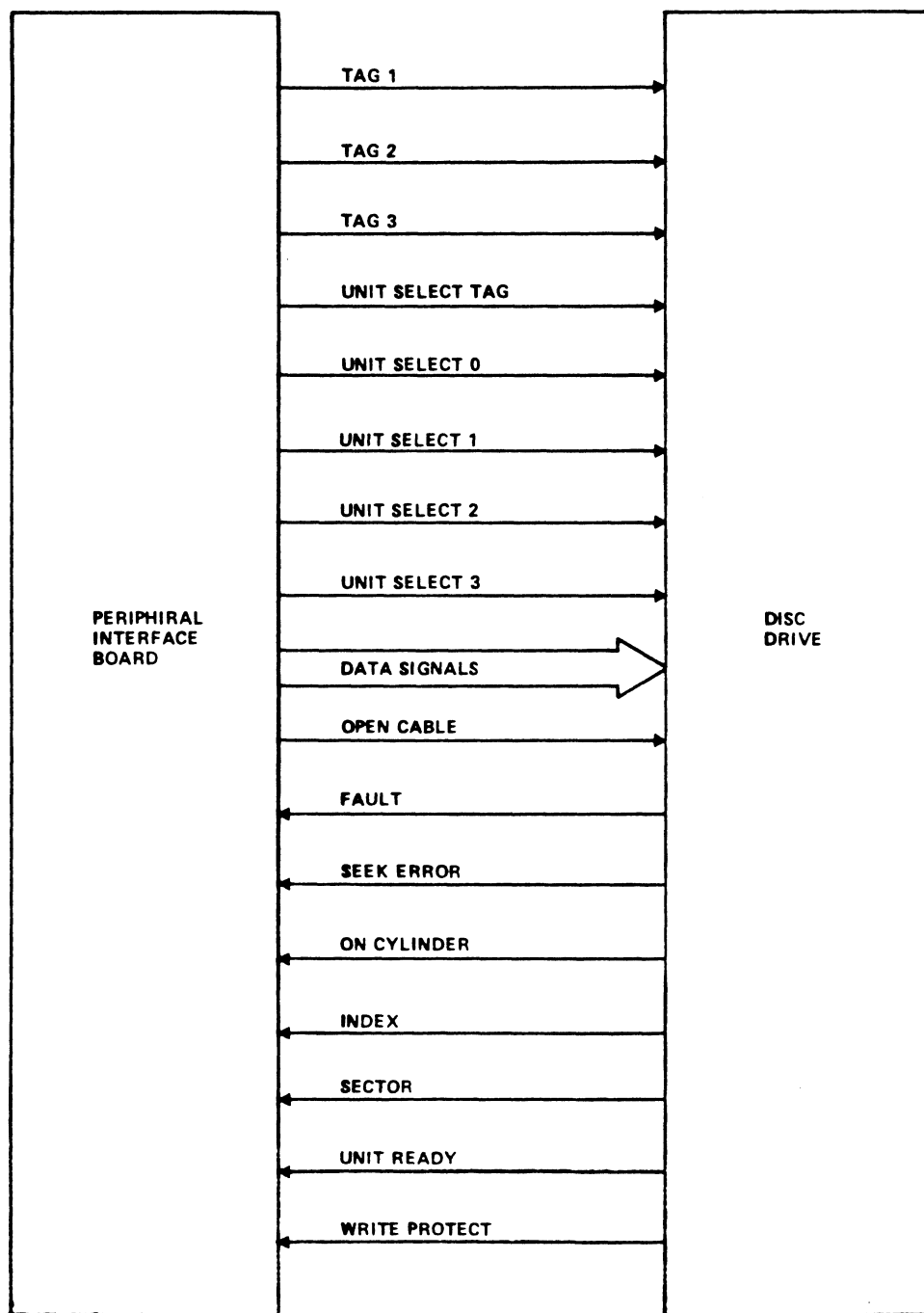


Figure 3-2. PIB/Disc Drive Differential Bus Signals

**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.

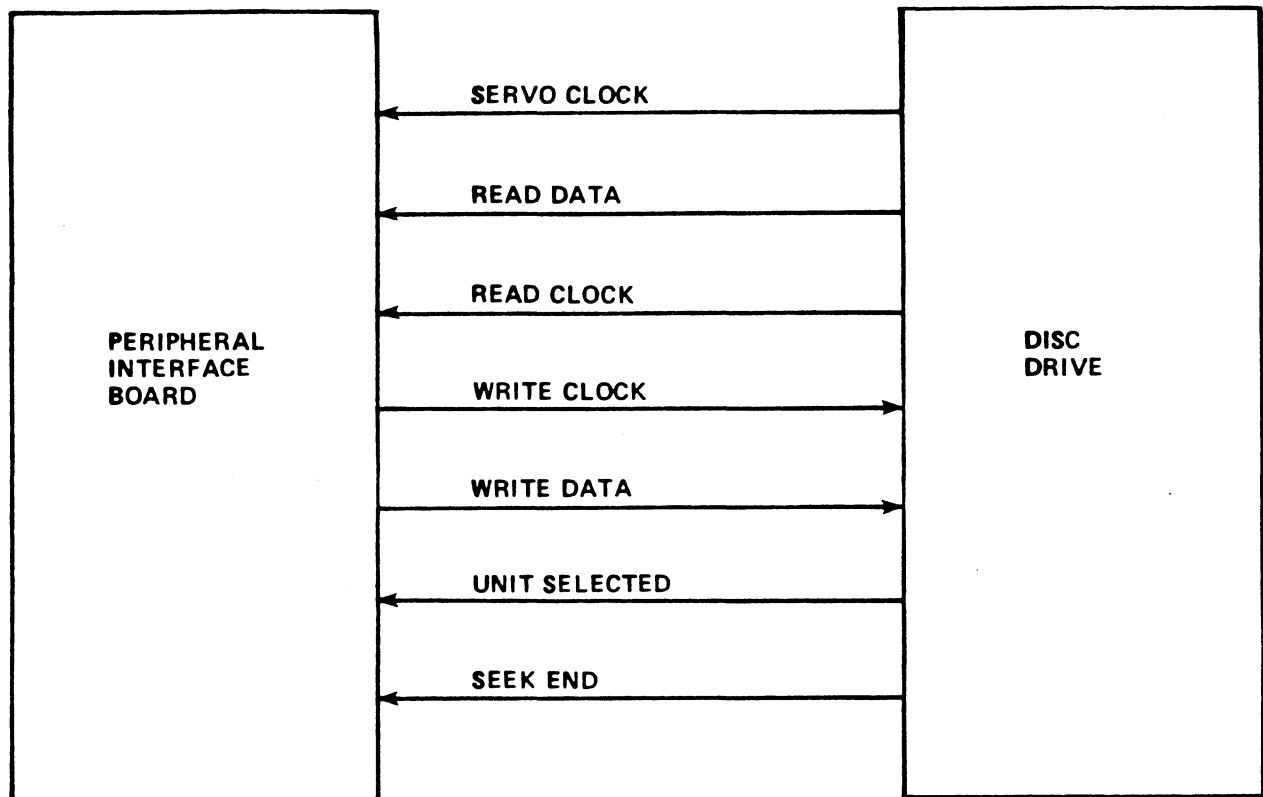


Figure 3-3. PIB/Disc Drive Differential Synchronization Cable Signals

**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.

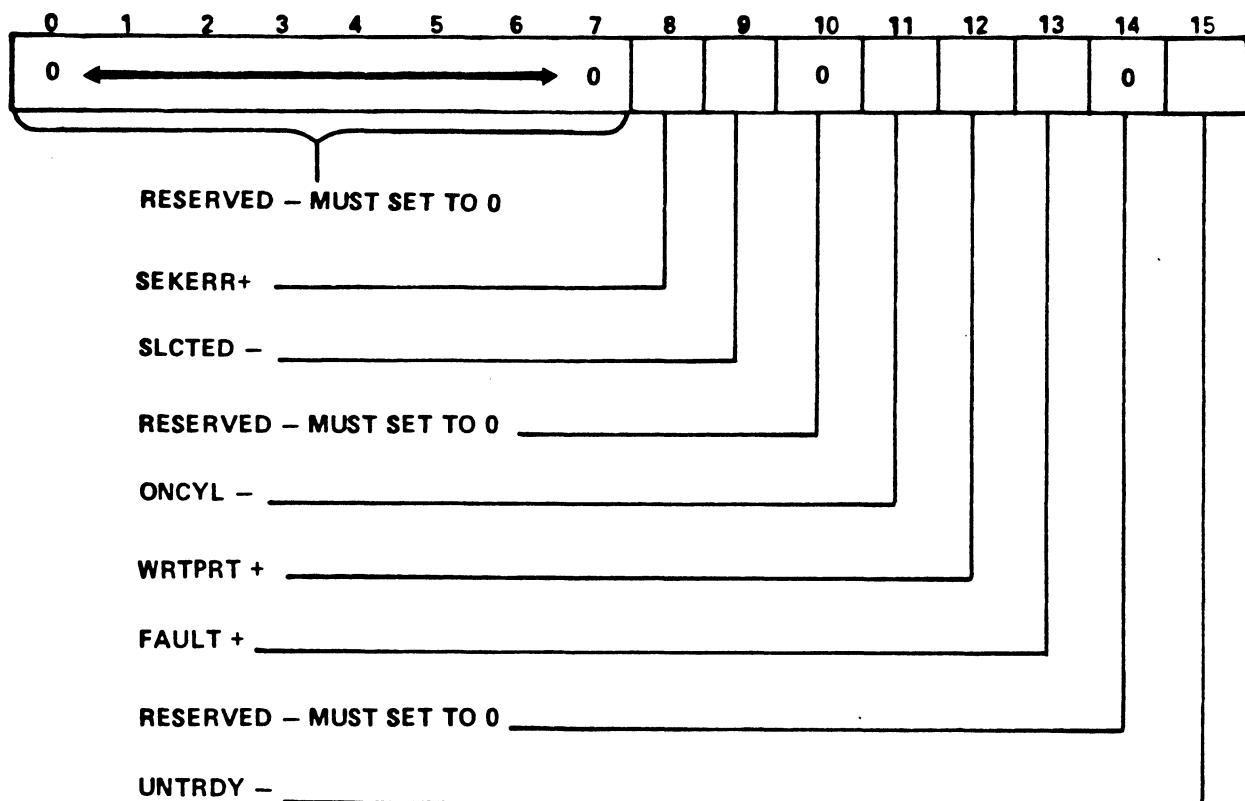


Figure 3-4. Drive Status

TABLE 3-3. DRIVE STATUS DEFINITIONS

Bits	Name	Function
0-7		Reserved, set to 0
8	SEKERR+	<p>Seek Error - When set to 1, indicates a seek error has occurred. A seek error can result from the following:</p> <ul style="list-style-type: none"> ● 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 <p>A seek error can only be cleared by issuing a return-to-zero command. See Section 3.4.3.</p>
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+	<p>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:</p> <ul style="list-style-type: none"> ● 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.

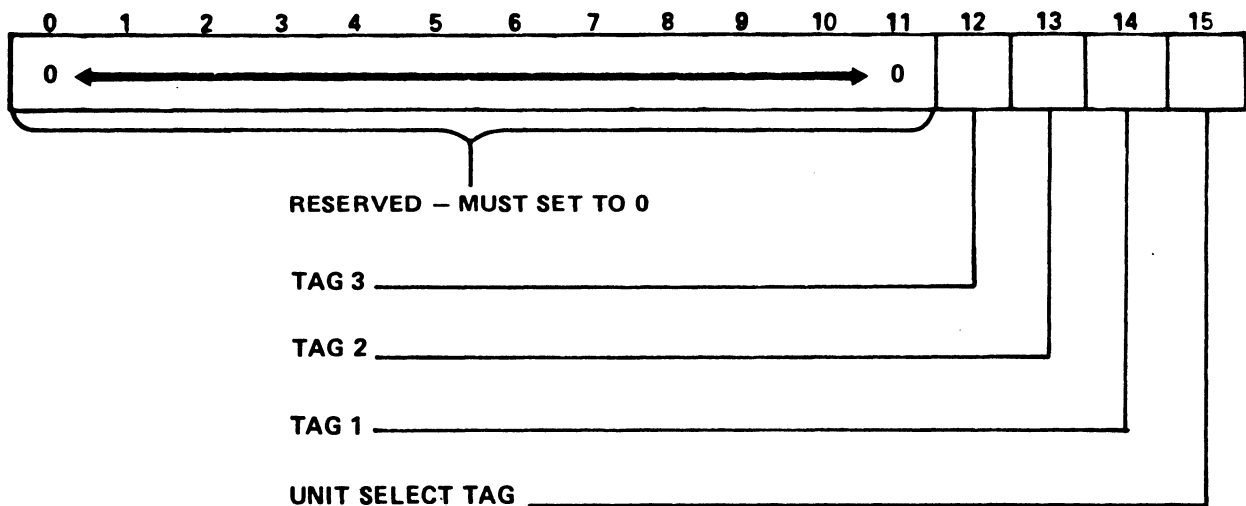


Figure 3-5. Contents of DOA Instruction Accumulator

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.

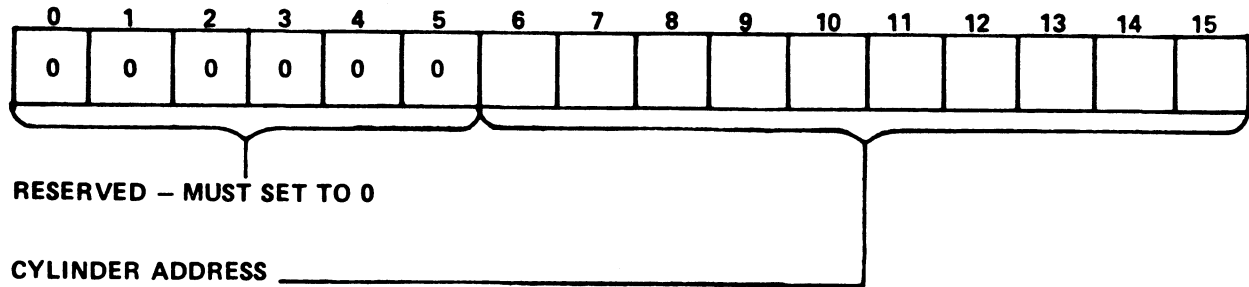


Figure 3-6. Tag 1 Instruction Format

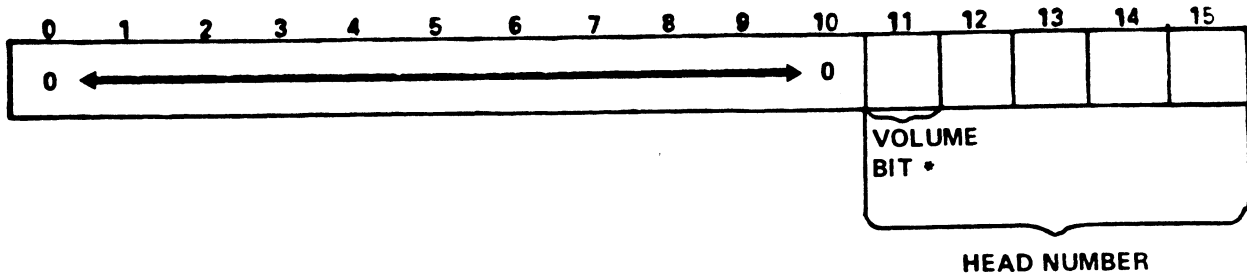
TABLE 3-5. DATA BUS BIT FUNCTIONS WITH TAG 1

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.



*CMD DRIVE ONLY

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.

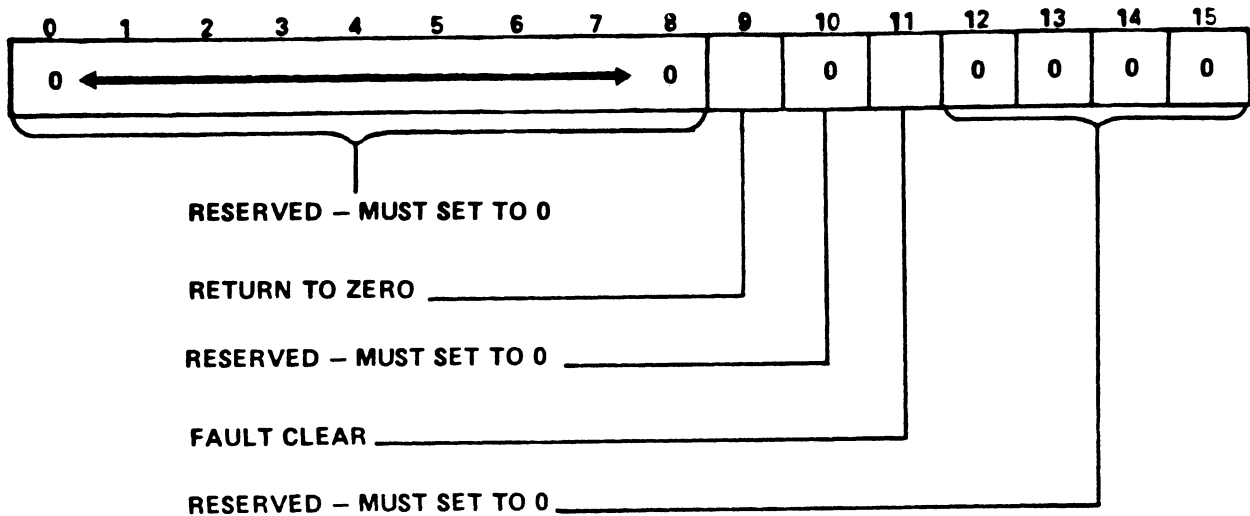


Figure 3-8. Tag 3 Instruction Format

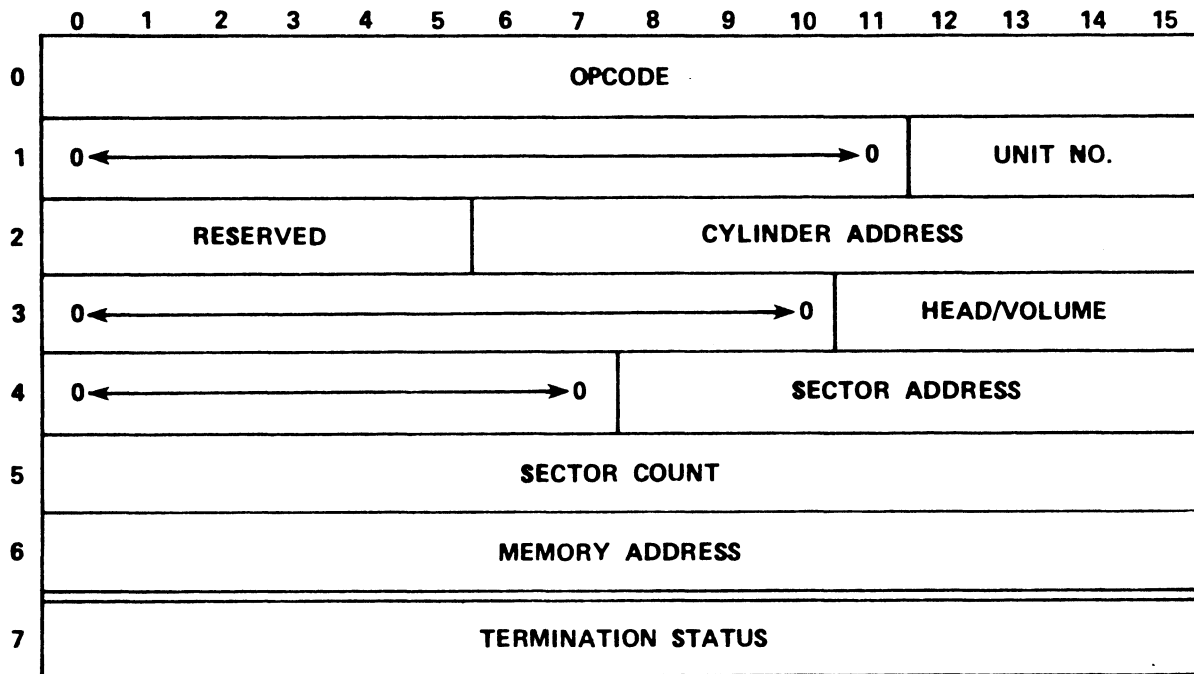
TABLE 3-7. DATA BUS BIT FUNCTIONS WITH TAG 3

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.



081-159

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.

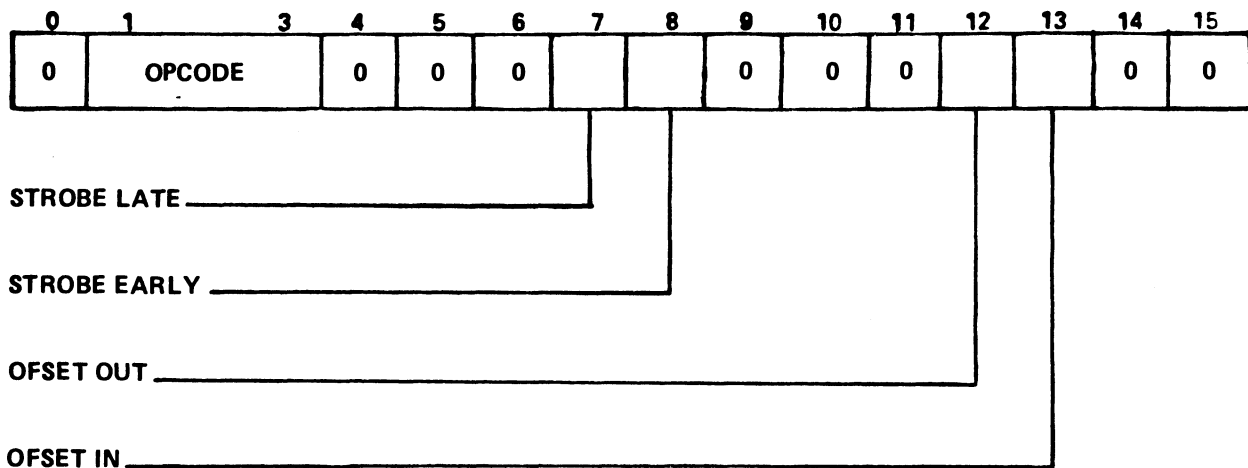


Figure 3-10. Opcode Word Format

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 OFFSET out
13	Set to 1 for OFFSET 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.

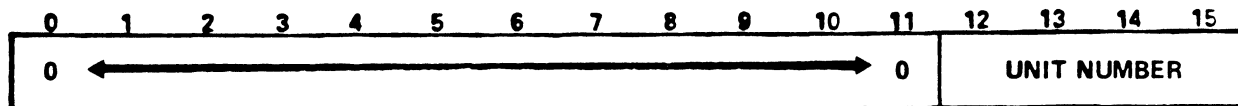


Figure 3-11. Unit Number Word Format

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.

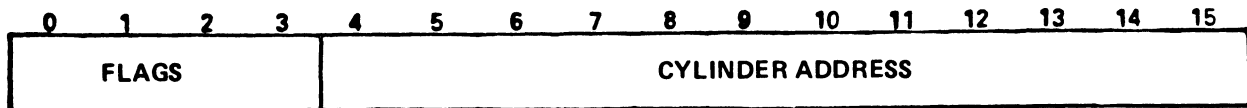


Figure 3-12. Cylinder Address Word Format

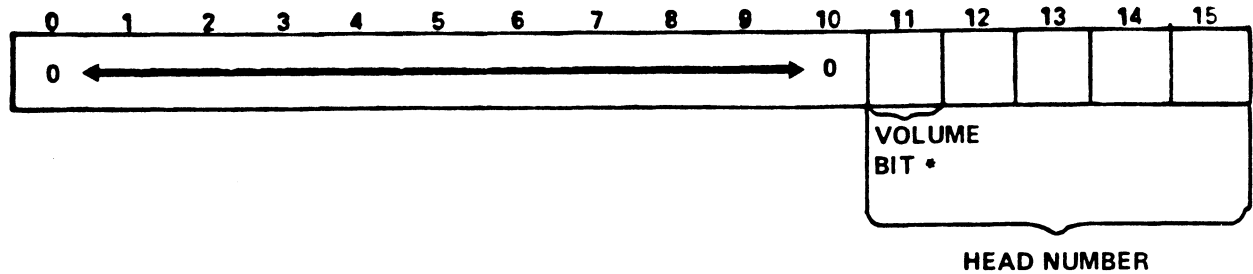
TABLE 3-10. DISC IOCB WORD 2

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:																		
	<table border="0"> <thead> <tr> <th>Bits</th> <th>Drive</th> </tr> </thead> <tbody> <tr> <td><u>11*12 13 14 15</u></td> <td><u>CMD Drives</u></td> </tr> <tr> <td>0 0 0 0 0</td> <td>Head number 0; Removable</td> </tr> <tr> <td>1 0 0 0 0</td> <td>Head number 0; Fixed</td> </tr> <tr> <td>1 0 0 0 1</td> <td>Head number 1; Fixed</td> </tr> <tr> <td><u>11 12 13 14 15</u></td> <td><u>SMD Drives</u></td> </tr> <tr> <td>0 0 0 0 0</td> <td>Head number 0</td> </tr> <tr> <td>0 0 0 0 1</td> <td>Head number 1</td> </tr> <tr> <td>0 0 0 1 0</td> <td>Head number 2</td> </tr> </tbody> </table>	Bits	Drive	<u>11*12 13 14 15</u>	<u>CMD Drives</u>	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	<u>11 12 13 14 15</u>	<u>SMD Drives</u>	0 0 0 0 0	Head number 0	0 0 0 0 1	Head number 1	0 0 0 1 0	Head number 2
Bits	Drive																		
<u>11*12 13 14 15</u>	<u>CMD Drives</u>																		
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																		
<u>11 12 13 14 15</u>	<u>SMD Drives</u>																		
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.

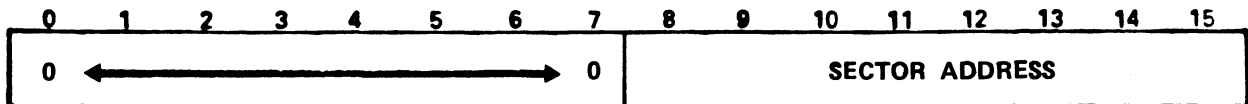


Figure 3-14. Sector Address Word Format

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.

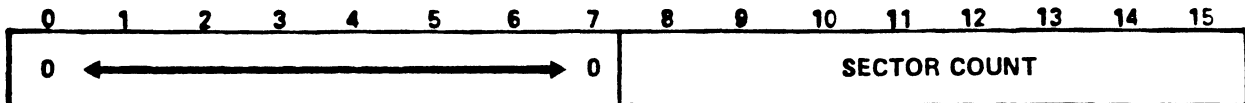


Figure 3-15. Sector Count Word Format

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.

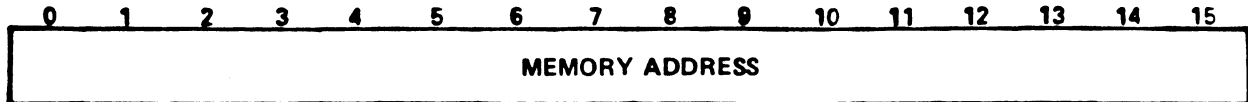


Figure 3-16. Memory Address Word Format

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.

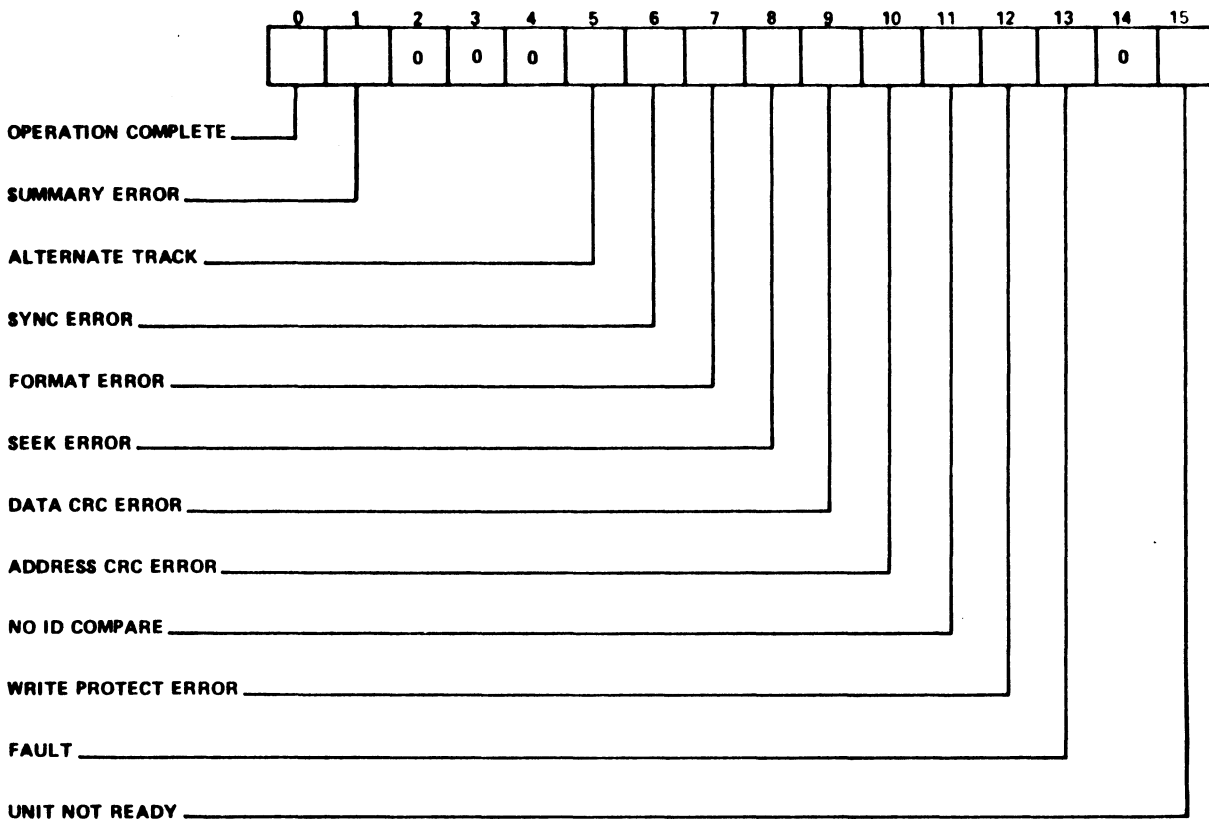


Figure 3-17. Termination Status Word Format

TABLE 3-15. DISC TERMINATION STATUS WORD 7

Bit	Definition
0	<p>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.</p>
1	<p>Summary Error - Indicates that one or more of the controller error bits is set.</p>
2-4	<p>Reserved - set to 0 by PIB.</p>
5	<p>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.</p> <p>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.</p> <p>When software detects the alternate-track bit set, a new seek should be issued based on the new information in the IOCB.</p>
6	<p>Synchronization Error - Indicates that the controller failed to synchronize with data (could not find any sectors on the track).</p>
7	<p>Format Error - Indicates that during a format, a track overrun occurred.</p>
8	<p>Seek Error - Indicates that a seek error occurred on the selected drive.</p>

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	<p>Fault - Indicates that a fault has occurred in the selected drive. A drive fault means that one or more of the following has occurred:</p> <ul style="list-style-type: none"> ● 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.

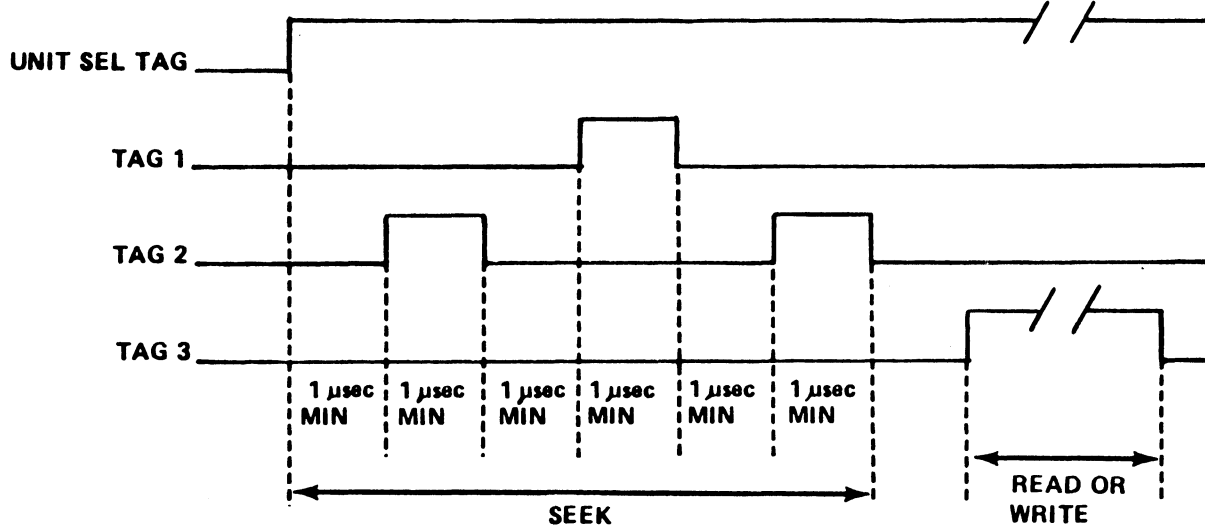
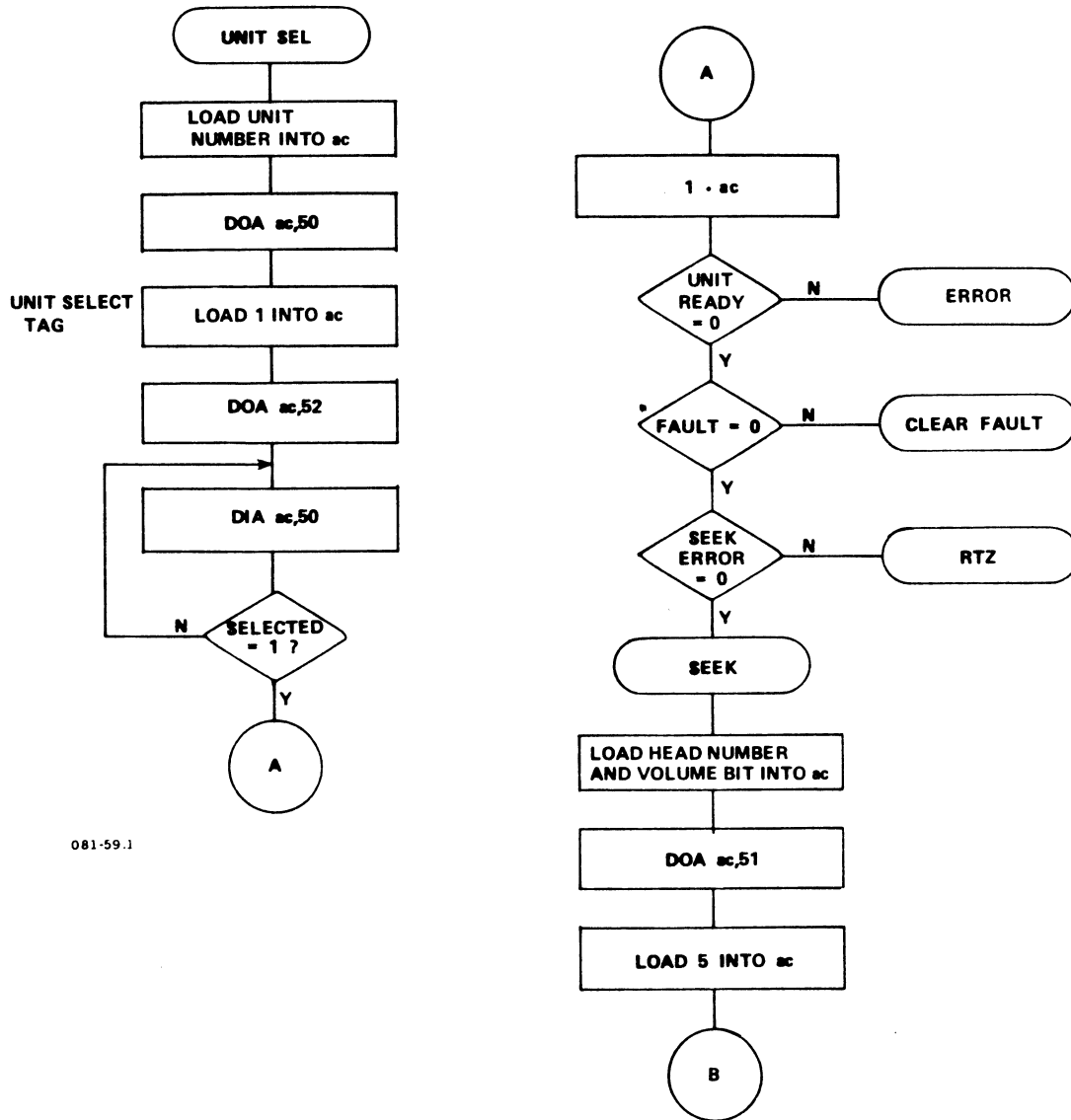


Figure 3-18. Seek Operation Timing



081-59.1

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

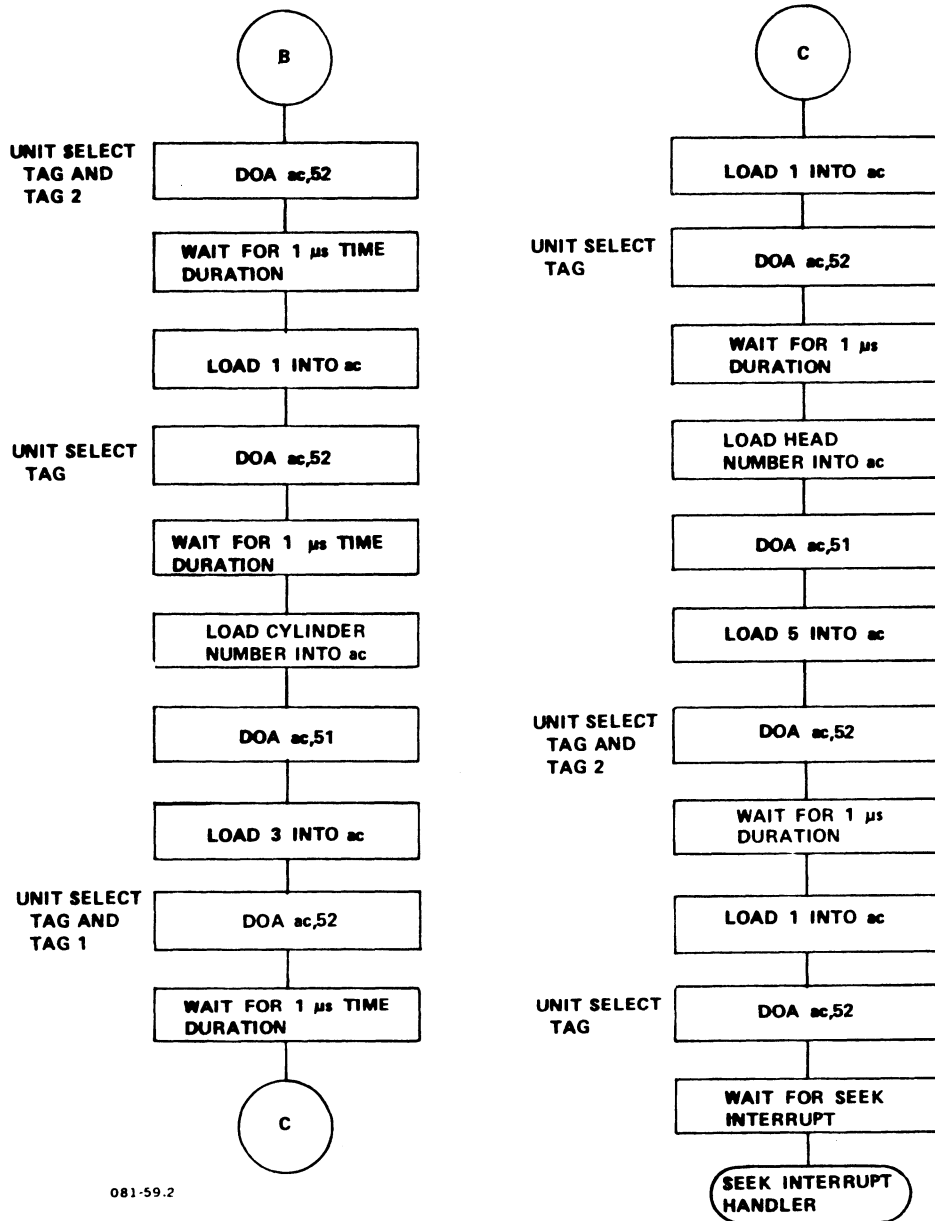
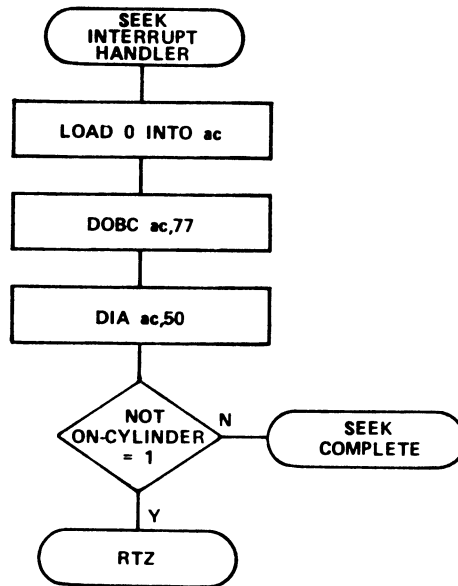


Figure 3-19. Drive Select and Seek Initiation (2 of 3)



081-59.3

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.

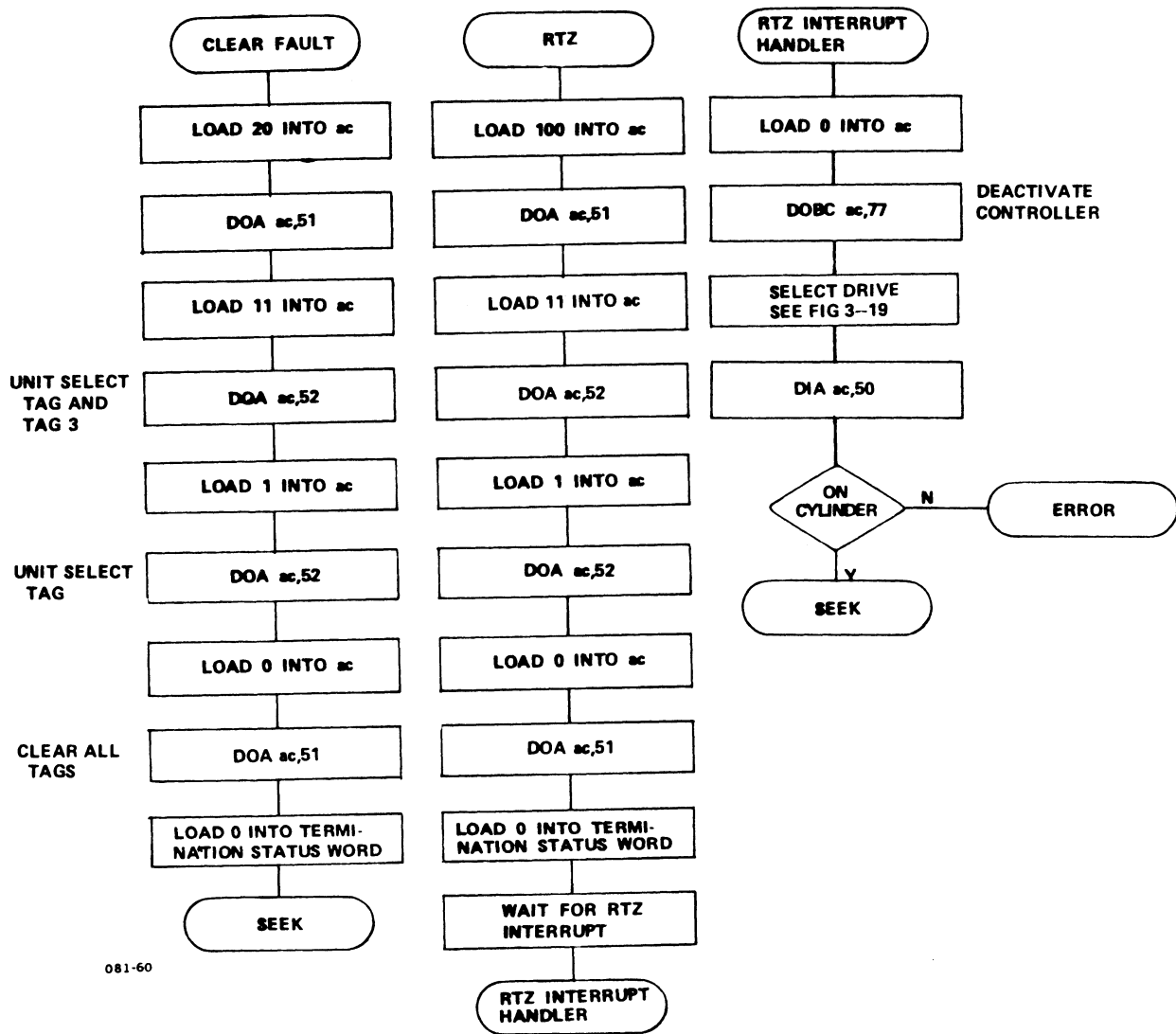


Figure 3-20. Seek Error Recovery Flowchart

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.

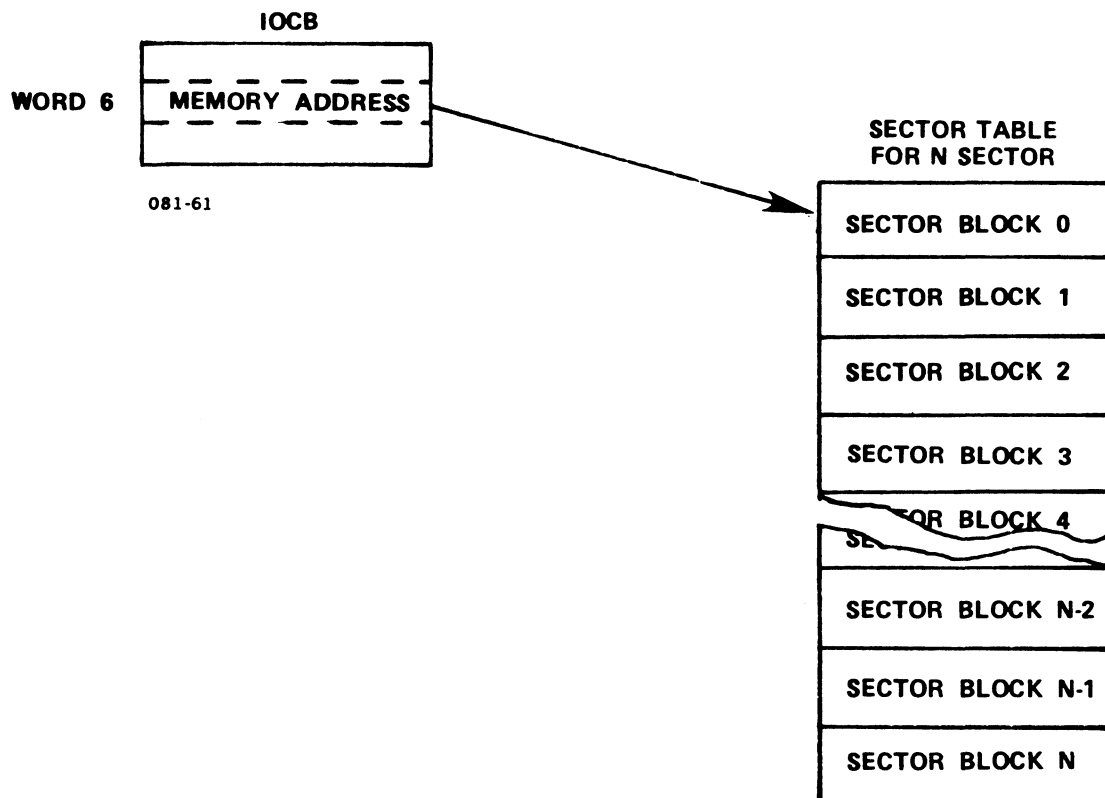


Figure 3-21. Sector Table in Memory

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.

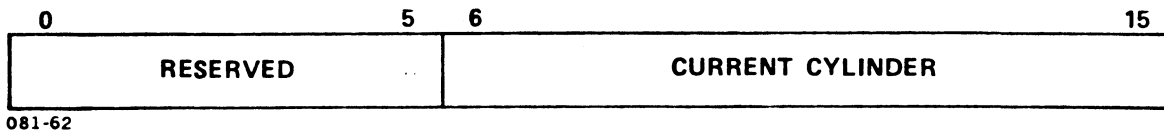


Figure 3-22. Sector Block Format - Word 0 Contents

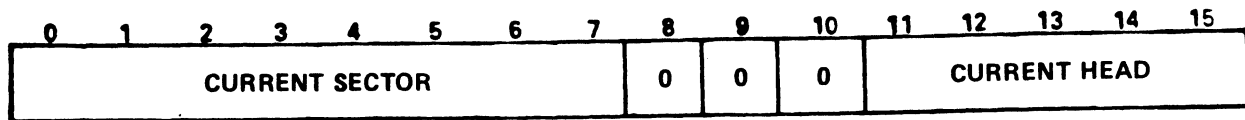
TABLE 3-16. SECTOR BLOCK FORMAT - WORD 0 FUNCTIONS

Bits	Function
0-5	Reserved.
6-15	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.



081-63

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

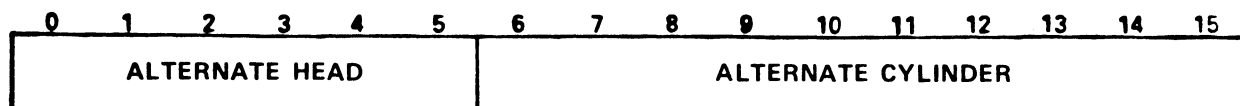
TABLE 3-17. SECTOR BLOCK FORMAT - WORD 1 FUNCTIONS

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-head/alternate-cylinder address of the sector. Figure 3-24 shows the contents of word 2.

Table 3-18 defines the fields in word 2.



081-64

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

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

Bits	Function
0-5	Defines the alternate-head (two's complement) address of the sector.
6-15	<p>Defines the alternate-cylinder address of the sector.</p> <p style="text-align: center;">NOTE</p> <p>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-head/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-head/alternate-cylinder word of each sector block in the sector table (every sector on the track) should contain the same value, <u>since the entire track will be flagged as bad.</u></p>

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.

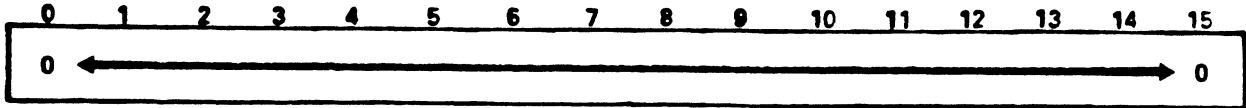


Figure 3-25. Sector Block Format - Word 3 Contents

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	0	1	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.

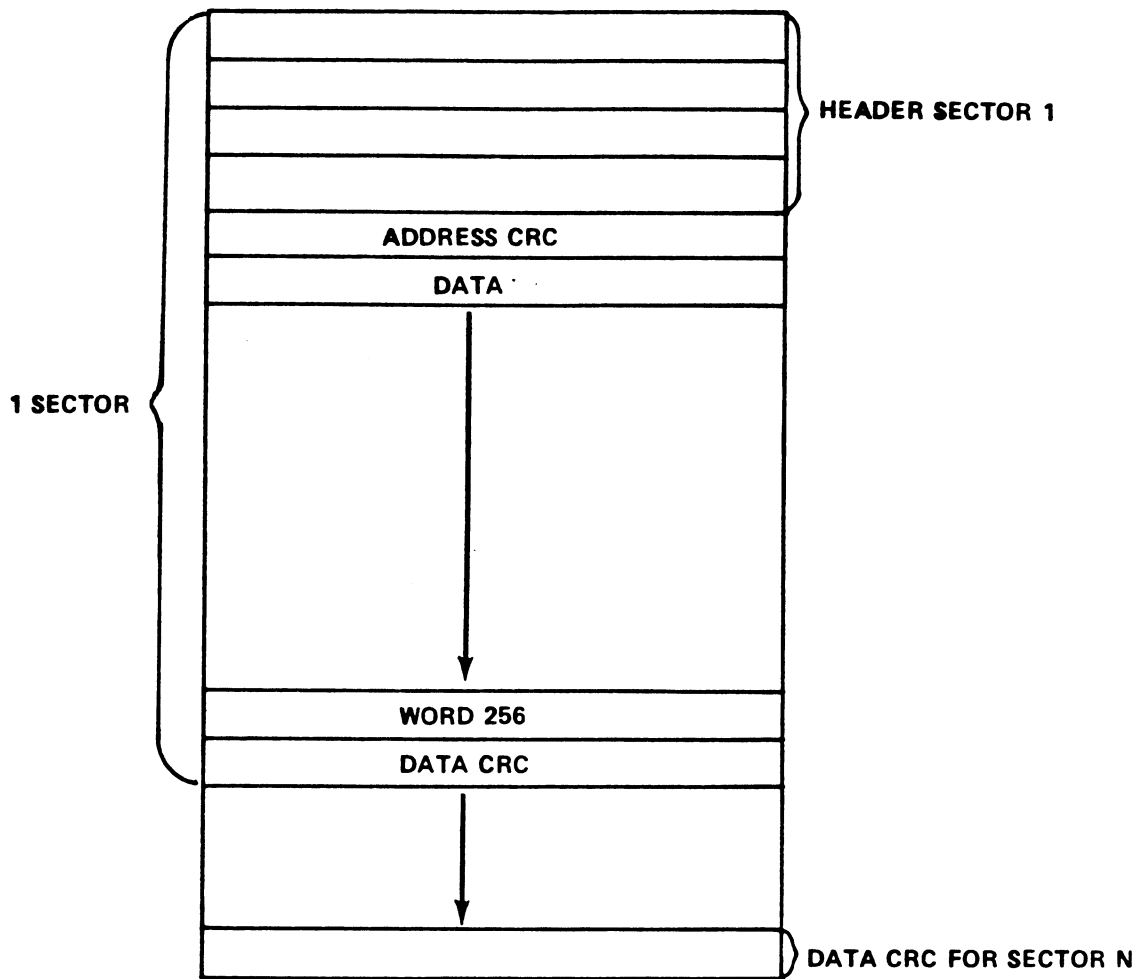


Figure 3-28. Read-Regardless Operation

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

QIC-02 STREAMING TAPE DRIVE INTERFACE

4.1 INTRODUCTION

This section describes the interface for the MARK 2/3 Streaming Tape Controller which is available on the MARK 2 and MARK 3 versions of the Peripheral Interface Board (PIB).

The POINT 4 MARK 2/3 Streaming Tape Controller is designed to handle one QIC-02-compatible Intelligent Interface. The Tape Controller provides high speed, direct memory access between the tape drive(s) 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 2/3 Tape Controller has the following features:

- Handles one Intelligent Interface
- Tape transfers at 90 kilobytes per second
- Accommodates tape speeds of 90 inches per second

4.1.2 PERFORMANCE CHARACTERISTICS

Tape Controller

Drive type:
QIC-02-compatible 1/4-inch streaming tape

Tape Transfer Rate:
90 kilobytes per second

Controller Device Codes:
60-63

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

Tape stop:
300 milliseconds maximum for 90 inches per second

Formatted capacity:
20 or 45 megabytes

Recording tracks:
4 or 9

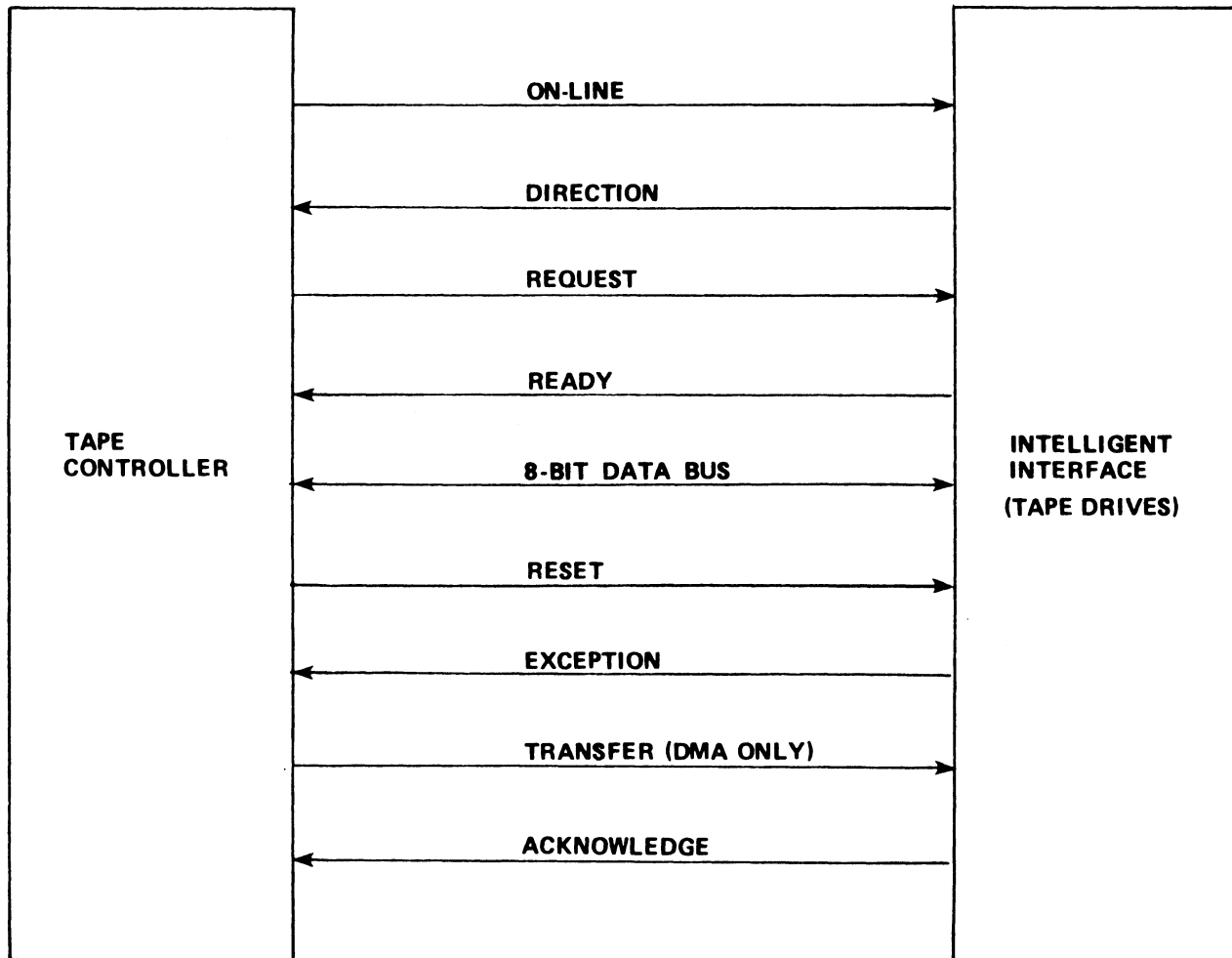
4.2 TAPE CONTROLLER INTERFACE

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.



081-69

Figure 4-1. Tape Controller Interface Signals

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

Pin	Signal	Direction	Description
1	GND	-	Ground
2	-	-	Reserved
3	GND	-	Ground
4	-	-	Reserved
5	GND	-	Ground
6	-	-	Reserved
7	GND	-	Ground
8	-	-	Reserved
9	GND	-	Ground
10	HBP-	Bidirec	Data Bus Odd Parity - reserved for optioned odd bus parity
11	GND	-	Ground
12	HB7-	Bidirec	Data Bus Bit 7 (MSB of 8-bit bidirectional data bus)
13	GND	-	Ground
14	HB6-	Bidirec	Data Bus Bit 6
15	GND	-	Ground
16	HB5-	Bidirec	Data Bus Bit 5
17	GND	-	Ground
18	HB4-	Bidirec	Data Bus Bit 4
19	GND	-	Ground
20	HB3-	Bidirec	Data Bus Bit 3
21	GND	-	GROUND
22	HB2-	Bidirec	Data Bus Bit 2
23	GND	-	Ground

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

Pin	Signal	Direction	Description
24	HB1-	Bidirec	Data Bus Bit 1
25	GND	-	Ground
26	HB0-	Bidirec	Data Bus Bit 0 (LSB of 8-bit bidirectional data bus)
27	GND	-	Ground
28	ONL-	OUT	Tape Controller is on-line to Intelligent Interface
29	GND	-	Ground
30	REQ-	OUT	Request - used by Tape Controller to initiate command sequence, provided RDY- or EXC- is asserted by drive
31	GND	-	Ground
32	RST-	OUT	Reset - resets Intelligent Interface, causes default selection to device 0, asserts Exception
33	GND	-	Ground
34	XFR-	OUT	Transfer - indicates that data is on bus, ready for transfer
35	GND	-	Ground
36	ACK-	IN	Acknowledge - indicates that data has been received by device
37	GND	-	Ground
38	RDY-	IN	Ready - Intelligent Interface is ready for new operation
39	GND	-	Ground
40	EXC-	IN	Exception - indicates exception condition in Intelligent Interface/drive operation

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

Pin	Signal	Direction	Description
41	GND	-	Ground
42	DIR-	IN	Direction - indicates direction of data flow
43	GND	-	Ground
44	-	-	Reserved
45	GND	-	Ground
46	-	-	Reserved
47	GND	-	Ground
48	-	-	Reserved
49	GND	-	Ground
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.

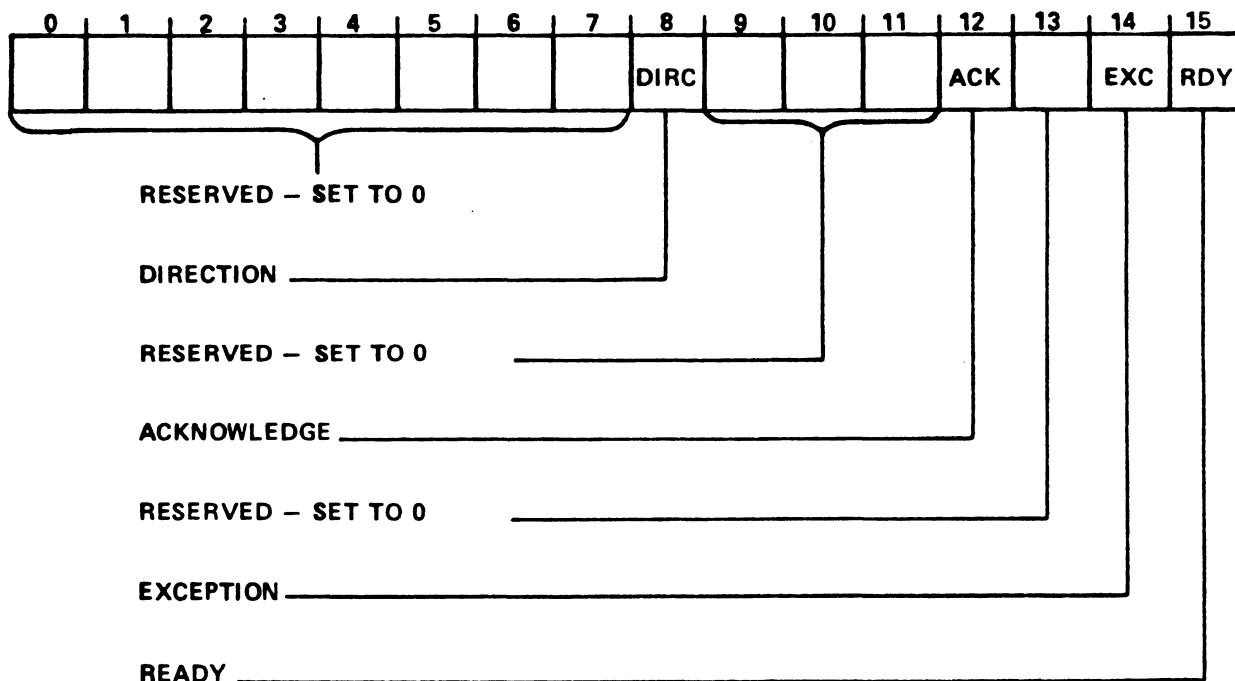


Figure 4-2. Format of DIA Instruction Accumulator

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	<p>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.</p> <p style="text-align: center;">NOTE</p> <p style="text-align: center;">This bit is used by the Tape Controller and need not concern the programmer.</p>
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)

Bits	Signal	Function																
15	RDY	<p>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.</p> <table border="0"> <thead> <tr> <th data-bbox="596 516 678 548"><u>Mode</u></th> <th data-bbox="984 516 1295 548"><u>RDY Bit Function</u></th> </tr> </thead> <tbody> <tr> <td data-bbox="596 579 753 642">Command Transfer</td> <td data-bbox="868 579 1430 674">Command has been taken from the data bus by the Intelligent Interface</td> </tr> <tr> <td data-bbox="596 705 712 768">Status Input</td> <td data-bbox="868 705 1430 800">Status byte has been placed on the data bus by the Intelligent Interface</td> </tr> <tr> <td data-bbox="596 831 811 863">Positioning</td> <td data-bbox="868 831 1430 905">Position command has been completed</td> </tr> <tr> <td data-bbox="596 936 736 968">Writing</td> <td data-bbox="868 936 1430 1062">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 data-bbox="596 1094 695 1188">Write File Mark</td> <td data-bbox="868 1094 1430 1157">Write-file-mark command is completed</td> </tr> <tr> <td data-bbox="596 1220 736 1251">Reading</td> <td data-bbox="868 1220 1430 1314">A buffer is ready to be emptied by the Tape Controller</td> </tr> <tr> <td data-bbox="596 1346 670 1377">Idle</td> <td data-bbox="868 1346 1430 1419">Tape Controller is ready to receive a new command</td> </tr> </tbody> </table>	<u>Mode</u>	<u>RDY Bit Function</u>	Command Transfer	Command has been taken from the data bus by the Intelligent Interface	Status Input	Status byte has been placed on the data bus by the Intelligent Interface	Positioning	Position command has been completed	Writing	An Intelligent Interface buffer is ready to be filled by the Tape Controller or a write-file-mark can be issued	Write File Mark	Write-file-mark command is completed	Reading	A buffer is ready to be emptied by the Tape Controller	Idle	Tape Controller is ready to receive a new command
<u>Mode</u>	<u>RDY Bit Function</u>																	
Command Transfer	Command has been taken from the data bus by the Intelligent Interface																	
Status Input	Status byte has been placed on the data bus by the Intelligent Interface																	
Positioning	Position command has been completed																	
Writing	An Intelligent Interface buffer is ready to be filled by the Tape Controller or a write-file-mark can be issued																	
Write File Mark	Write-file-mark command is completed																	
Reading	A buffer is ready to be emptied by the Tape Controller																	
Idle	Tape Controller is ready to receive a new command																	

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.

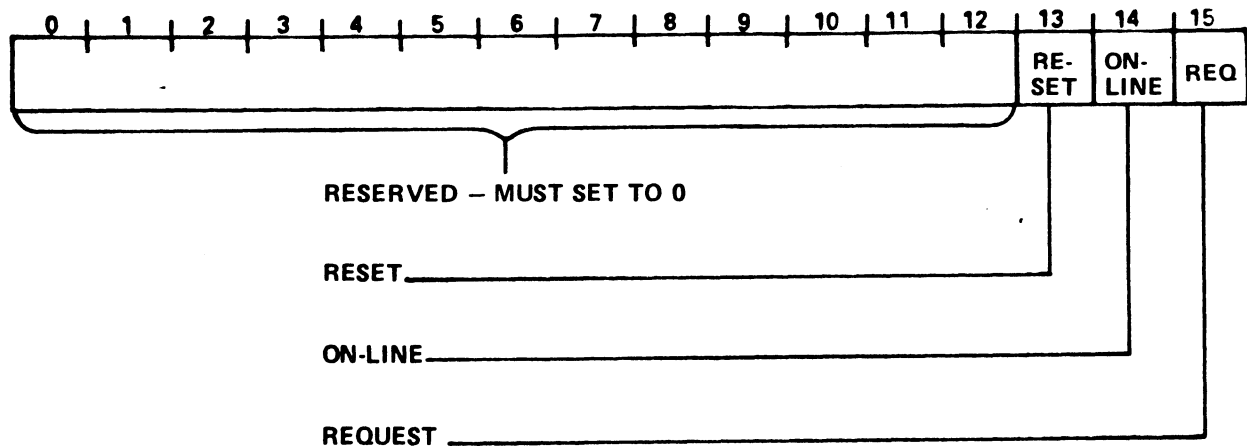


Figure 4-3. Contents of DOA Instruction Accumulator

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	<p>This bit <u>must</u> be set for the following command modes:</p> <ul style="list-style-type: none"> ● Writing ● Reading ● Write File Mark ● Read File Mark <p>The on-line bit performs the following functions depending upon the operational mode.</p> <table border="0" style="width: 100%;"> <thead> <tr> <th style="text-align: left;"><u>Mode</u></th> <th style="text-align: left;"><u>On-line Bit Function</u></th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;">Write</td> <td>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).</td> </tr> <tr> <td style="vertical-align: top;">Read</td> <td>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.</td> </tr> </tbody> </table>	<u>Mode</u>	<u>On-line Bit Function</u>	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.
<u>Mode</u>	<u>On-line Bit Function</u>							
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
		<p style="text-align: center;"><u>Mode</u> <u>On-line Bit Function</u></p> <p>Write File Mark The Intelligent Interface must be on-line to the Tape Controller to execute the write-file-mark command.</p> <p>Read File Mark The Intelligent Interface must be on-line to the Tape Controller to execute the read-file-mark command.</p>
15	REQ	<p>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.</p>

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.

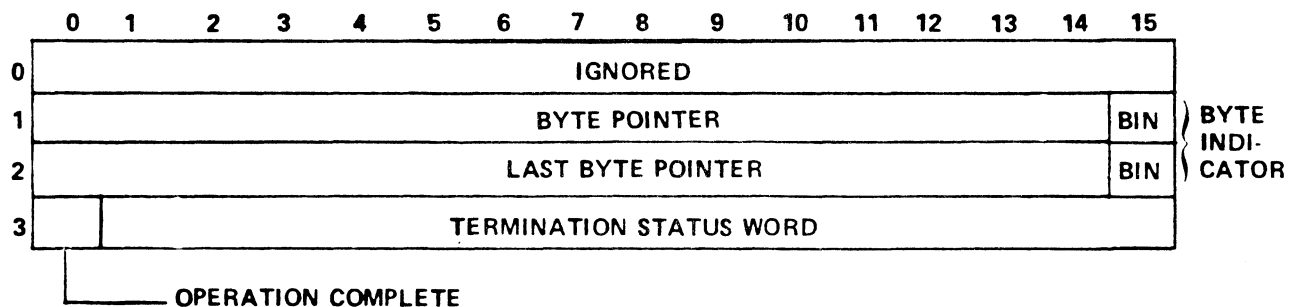


Figure 4-4. Tape Controller IOCB Format

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.

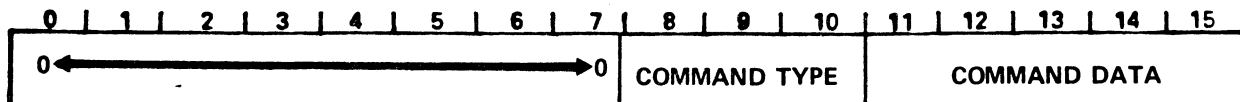


Figure 4-5. Tape Controller Command Format

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.

Type	Bit Position 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.

<u>Bit</u>	<u>Description</u>
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.

<u>Bit</u>	<u>Description</u>
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.

<u>Bit</u>	<u>Description</u>
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.

<u>Bit</u>	<u>Description</u>
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.

<u>Bit</u>	<u>Description</u>
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.

<u>Bit</u>	<u>Description</u>
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:

<u>Bit</u>	<u>Description</u>
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.

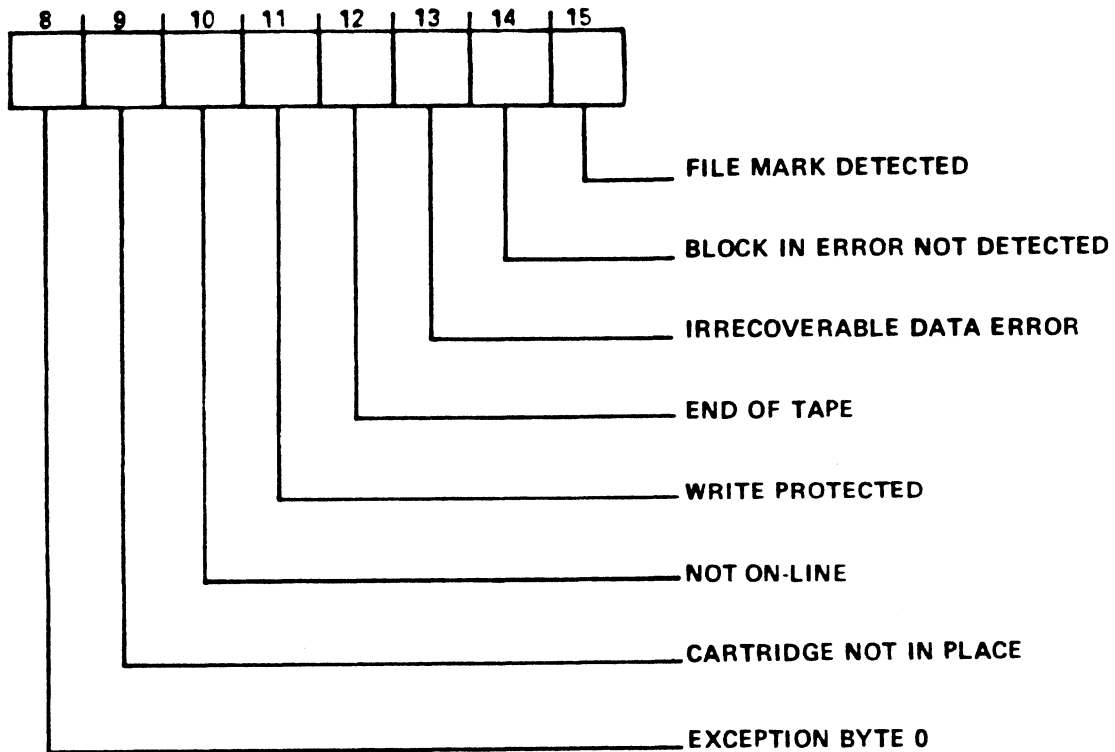


Figure 4-6. Exception Byte 0

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.

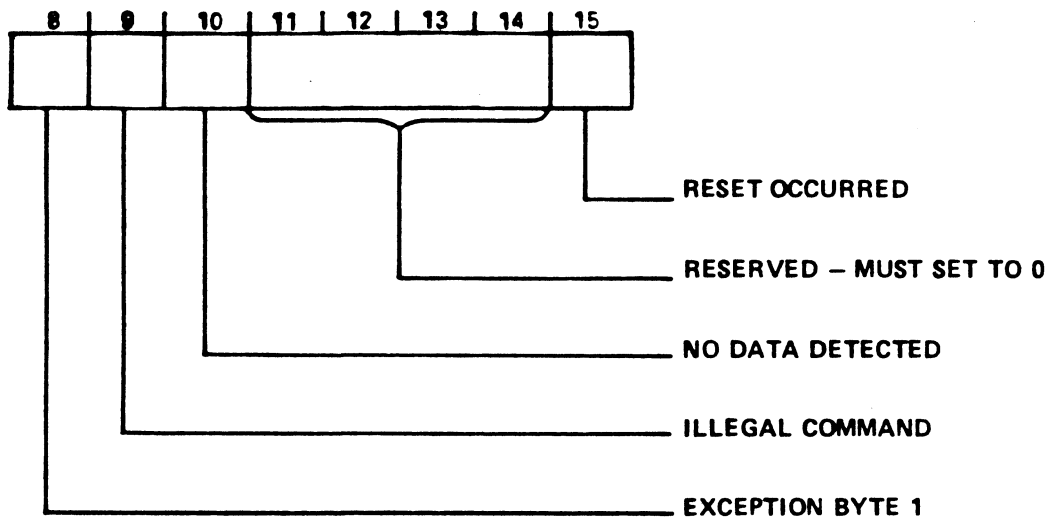


Figure 4-7. Exception Byte 1

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: <ol style="list-style-type: none">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.



Section 5

FLOPPY DISC DRIVE INTERFACE

5.1 INTRODUCTION

This section describes the interface for the MARK 3 Floppy Disc controller which is available on the MARK 3B version of the Peripheral Interface Board (PIB).

The POINT 4 MARK 3B PIB utilizes a state-of-the-art Western Digital 1797 controller chip. The Floppy Disc Controller is designed to handle up to four Shugart-compatible drives with single or double-sided, single or double-density capabilities. It provides high-speed, direct-memory access to the MARK 3 CPU. Multiple drives are accommodated by daisy chaining using connector J1 for 8-inch or J17 for 5-1/4-inch floppy disc drives. Drives used with the MARK 3B system must have soft sectoring and head-load-on-select capabilities.

Diskettes are an inexpensive means of mass storage. They are small and easily transportable. If security is a consideration, they are easily stored in a locked cabinet or may even be carried in a brief case. Diskettes are most frequently used for word processing and inputting data records.

The MARK 3B Peripheral Interface Board (PIB) supports either 8-inch or 5-1/4-inch drives. Both sizes cannot be run on the same PIB because each size drive has different cabling and timing requirements.

The Interface supports the following IBM formats:

IBM 1 FM	(single-sided, single-density Frequency Modulation)
IBM 2 MFM	(single-sided, double-density Modified Frequency Modulation)
IBM 2D MFM	(double-sided, double-density Modified Frequency Modulation)

This section describes the MARK 3 floppy disc drive interface in general terms. Some discussions, such as those dealing with diskette formatting, are annotated for the IRIS user. Users of other operating systems may wish to consult Western Digital's FD 1797 Data Sheet for appropriate formatting information.

5.1.1 FLOPPY DISC CONTROLLER FEATURES

THE POINT 4 MARK 3B Floppy Disc Controller has the following features:

- Supports up to four Shugart-compatible (soft sectored) drives
- 8-inch or 5-1/4-inch diskettes
- FM and MFM encode and decode logic
- Side select
- Address mark detection
- Control-block-oriented instructions
- Software control of sector addressing
- Read/Write head position verification
- Floppy disc operation handled via an I/O control block (IOCB)
- I/O Instructions using device codes 30 to 37
- Data transfer is performed in DMA mode (device code 77)
- Status and error reporting on completion of operation
- Write protect
- CRC (cyclic redundancy check) generation and detection
- Controller device codes range from 30 to 37
- Drive port assignments for floppy disc:
 - J1 for 8-inch
 - J17 for 5-1/4-inch

5.1.2 TYPICAL DRIVE PERFORMANCE CHARACTERISTICS

Performance depends on the type of drive used. The performance characteristics given here are typical for most drives.

- DMA transfer at 250 kilobits per second for single density and 500 kilobits per second for double density diskettes.
- Average latency is 83 milliseconds.
- The capacity of the diskettes depends on the format used:

<u>Unformatted</u>	<u>Single Density</u>	<u>Double Density</u>
Per Disc	800 kilobits	1600 kilobits
Per Surface	400 kilobits	800 kilobits
Per Track	5.2 kilobits	10.4 kilobits

<u>IBM Format (128-Byte Sectors)</u>	<u>Single Density</u>	<u>Double Density</u>
Per Disc	500 kilobits	1000 kilobits
Per Surface	250 kilobits	500 kilobits
Per Track	3.3 kilobits	6.66 kilobits

- Typical access times are as follows:

Track-to-track	3 milliseconds
Settling time	15 milliseconds
Head load time	50 milliseconds
Average (including settling)	90 milliseconds

5.1.3 DISKETTE SPECIFICATIONS

Diskettes may be either 8-inch or 5-1/4-inch depending on the size of the drive. Specifications are as follows:

8-inch Diskettes:

- | ● Single or double sided
- | ● Single or double density
- Sectors may be formatted as follows:

128 bytes

256 bytes

512 bytes*

1024 bytes

*IRIS requires that diskettes be double density and be formatted into 15 sectors of 512 bytes.

5-1/4-inch Diskettes:

- | ● Single or double sided
- | ● Single or double density
- Sectors may be formatted as follows:

128 bytes

257 bytes

512 bytes**

1024 bytes

**IRIS requires that diskettes be double density, be formatted into sectors of 512 bytes, and have a capacity of at least 750K bytes after formatting.

5.2 FLOPPY DISC DRIVE INTERFACE

The Floppy Disc Controller supports up to four diskette drives. The drive(s) must be Shugart-compatible (soft sector), and have head-load-on-select capability. If two or more drives are used, they must be daisy-chained. Figure 5-1 illustrates the daisy-chain connection.

Figure 5-2 shows the Floppy Disc Drive interface signals. Table 5-1 lists the J1 signal functions (for 8-inch floppy drives) and Table 5-2 the J17 signal functions (for 5-1/4-inch floppy drives).

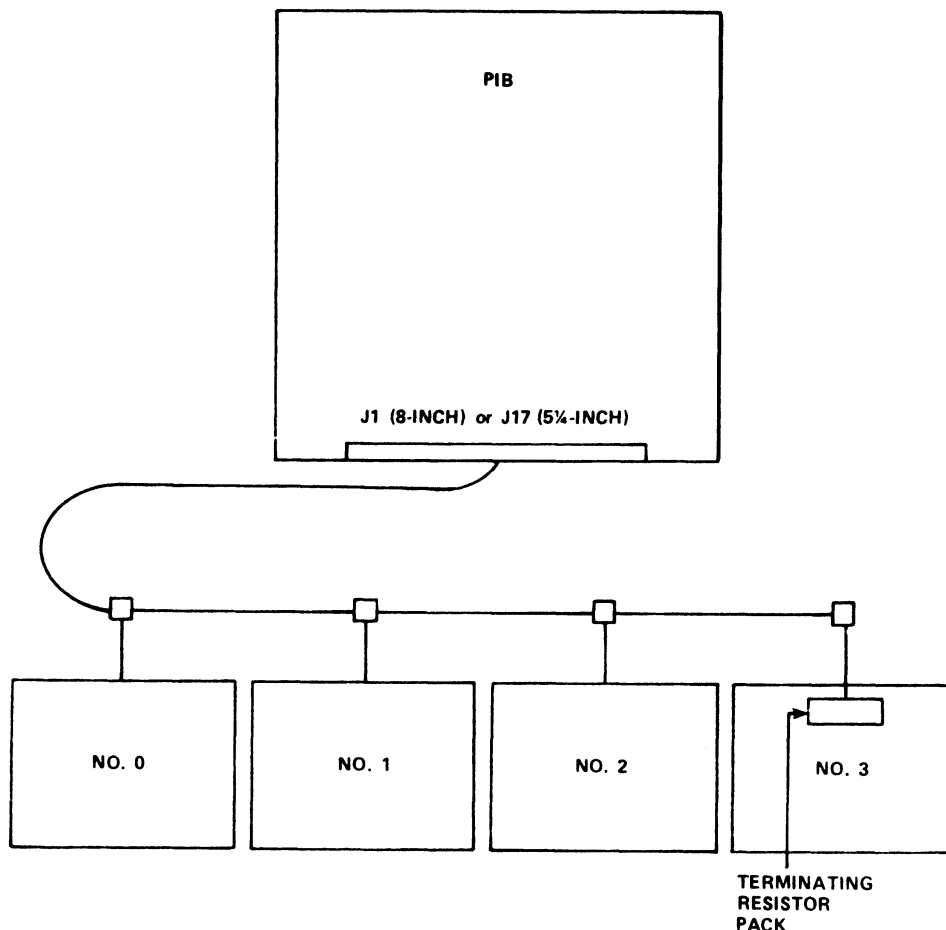


Figure 5-1. Daisy-Chain Drive Connection

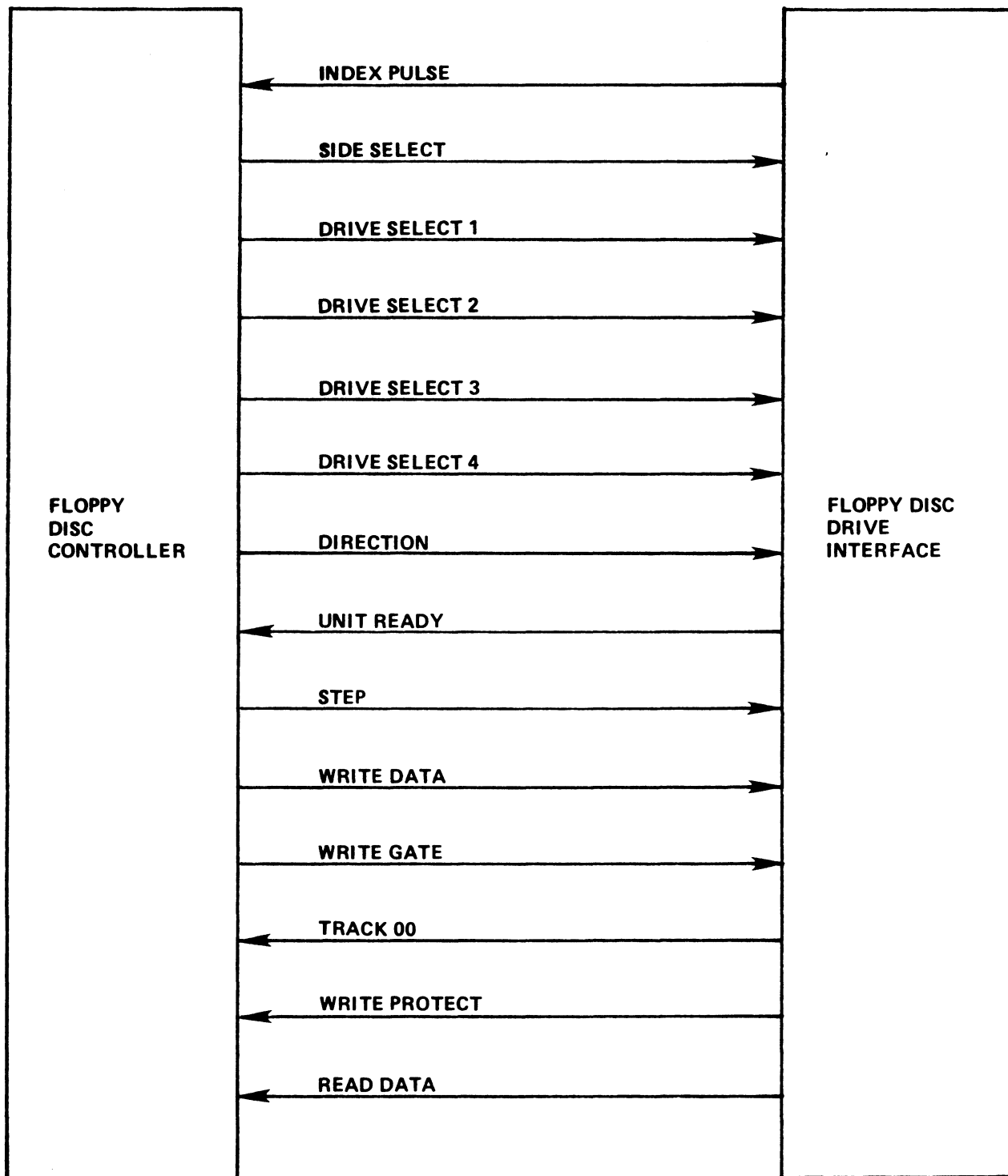


Figure 5-2. PIB/Floppy Disc Drive Interface Signals

**TABLE 5-1. 8-INCH FLOPPY DISC CONTROLLER INTERFACE
(J1 SIGNAL FUNCTIONS)**

Pin	Signal	Direction	Description
1			
2			
3	GND		Ground
4			
5	GND		Ground
6			
7			
8			
9	GND		Ground
10			
11	GND		Ground
12			
13	GND		Ground
14	SIDE 1 SELD-	OUT	Side 1 Select - Indicates which side of a two-sided diskette to use for reading or writing. 0 = Side 0 1 = Side 1
15	GND		Ground
16			
17	GND		Ground
18			
19	GND		Ground

**TABLE 5-1. 8-INCH FLOPPY DISC CONTROLLER INTERFACE (Cont)
(J1 SIGNAL FUNCTIONS)**

Pin	Signal	Direction	Description
20	IP-	IN	Index Pulse - Provided by a drive once each revolution of the diskette (166.67 milliseconds) to indicate the beginning of the track.
21	GND		Ground
22	READY-	IN	Unit Ready - Indicates that two index holes were sensed after proper insertion of diskette and closing the door; or two index holes were sensed following the application of +5V power to the drive. Three holes must be sensed for two-sided diskettes. READY is activated for a two-sided diskette when either side is selected. For a one-sided diskette READY is activated when side 0 is selected.
23	GND		Ground
24			
25	GND		Ground
26	DRVSEL 1-	OUT	Select drive 1
27	GND		Ground
28	DRVSEL 2-	OUT	Select drive 2
29	GND		Ground
30	DRVSEL 3-	OUT	Select drive 3
31	GND		Ground
32	DRVSEL 4-	OUT	Select drive 4
33	GND		Ground

**TABLE 5-1. 8-INCH FLOPPY DISC CONTROLLER INTERFACE (Cont)
(J1 SIGNAL FUNCTIONS)**

Pin	Signal	Direction	Description
34	DIR-	OUT	Direction Select - Defines which direction the Read/Write heads will move.
35			
36	STEP-	OUT	Step - Causes the Read/Write heads to move as defined by DIR-.
37	GND		Ground
38	WDATA-	OUT	Write Data - Provides the data to be written to the diskette.
39	GND		Ground
40	WGATE-	OUT	Write Gate - When activated, enables WDATA to write to diskette.
41	GND		Ground
42	TR00-	IN	Track 00 - Indicates when Read/Write heads are positioned on track zero.
43	GND		Ground
44	WPRT-	IN	Write Protect - Inhibits writing if a write-protected diskette is installed.
45	GND		Ground
46	RDATA-	IN	Read Data - Provides the raw data (clock and data together).
47	GND		Ground
48			
49	GND		Ground
50			

**TABLE 5-2. 5-1/4-INCH FLOPPY DISC CONTROLLER INTERFACE
(J17 SIGNAL FUNCTIONS)**

Pin	Signal	Direction	Description
1	GND		Ground
2			
3	GND		Ground
4			
5	GND		Ground
6	DRVSEL 4-	OUT	Select drive 4.
7	READY-	IN	Forced low by interface. Not normally supported by 5-1/4-inch drives.
8	IP-	IN	Index Pulse - Provided by a drive once each revolution of the diskette (166.67 milliseconds) to indicate the beginning of the track.
9	GND		Ground
10	DRVSEL1-	OUT	Select drive 1
11	GND		Ground
12	DRVSEL2-	OUT	Select drive 2
13	GND		Ground
14	DRVSEL3-	OUT	Select drive 3
15	GND		Ground
16			
17	GND		Ground
18	DIR-	OUT	Direction Select - Defines which direction the Read/Write heads will move when the step line is pulsed.
19	GND		Ground

**TABLE 5-2. 5-1/4-INCH FLOPPY DISC CONTROLLER INTERFACE (Cont)
(J17 SIGNAL FUNCTIONS)**

Pin	Signal	Direction	Description
20	STEP-	OUT	Step - Causes the Read/Write heads to move as defined by DIR-.
21	GND		Ground
22	WDATA-	OUT	Write Data - Provides the data to be written to the diskette.
23	GND		Ground
24	WGATE-	OUT	Write Gate - When activated, enables WDATA to write to diskette.
25	GND		GROUND
26	TR00-	IN	Track 00 - Indicates when Read/Write heads are positioned on track zero.
27	GND		GROUND
28	WPRT-	IN	Write Protect - Inhibits writing if a diskette is write-protected.
29	GND		GROUND
30	RDATA-	IN	Read Data - Provides the raw data (data and clock together).
31	GND		GROUND
32	SIDE 1 SELD-	OUT	Side 1 Select - Indicates which side of a two-sided diskette to use for reading or writing. 0 = Side 0 1 = Side 1
33	GND		Ground
34			

5.3 PROGRAMMED INPUT/OUTPUT

Two modes of operation are standard with the Floppy Disc Controller: programmed I/O and DMA (Direct Memory Access).

Three types of command are used to communicate with the controller:

```
DOBP ac,77  (initiates a floppy data transfer)
DIA ac,3x   (Read from Floppy Controller register)
DOA ac,3x   (Write to Floppy Controller register)
```

where

ac = accumulator

3x = device code

The device codes used by the Floppy Disc Controller range from 30 to 37:

<u>Device Code</u>	<u>DOA Instruction</u>	<u>DIA Instruction</u>
30/32	Select Register	*
31/33	Intermediate Buffer	Intermediate Buffer**
34	Command Register	Status Register
35	Track Register	Track Register
36	Sector Register	Sector Register
37	Data Register	Data Register

*The Select Register can be written to but not read from.

**Reserved for firmware or test purposes.

A DOA instruction is used to write to and a DIA instruction is used to read from a Floppy Disc Controller register. The general command formats are:

```
DOA ac,3x
DIA ac,3x
```

where

ac = accumulator

3x = device code

The following subsections provide information on floppy disc programmed I/O and the use of registers.

5.3.1 SELECT REGISTER

The Select Register can be written to but may not be read from. It is loaded using the least significant six bits of the specified accumulator as shown in Figure 5-3.

The command format is

DOA ac,30

where ac = the accumulator

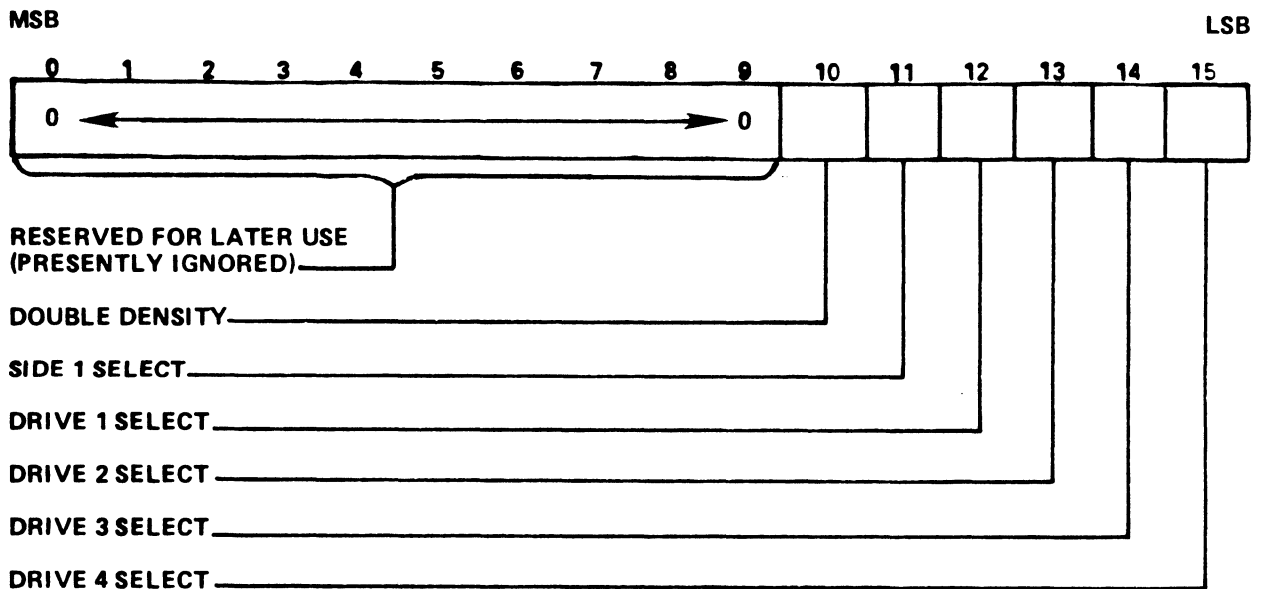


Figure 5-3. Select Register Word

5.3.2 COMMAND REGISTER

The Command and Status Registers (see Section 5.3.3) are physically the same register. As a Command Register, it holds the command currently being executed. A command is input with a DOA instruction. When execution of the command is completed, or if a halt occurs because of an error condition, status information may be retrieved with a DIA instruction.

Five instructions are available to software:

1. Restore
2. Seek
3. Step
4. Step In
5. Step Out

Of these instructions only Restore and Seek are required. The bit configuration is discussed in Sections 5.3.2.1 and 5.3.2.2 respectively. For information on the Step commands, refer to Western Digital's FD 1797 Data Sheet.

The Command Register should not be loaded when the device is busy.

The command format is

DOA ac,34

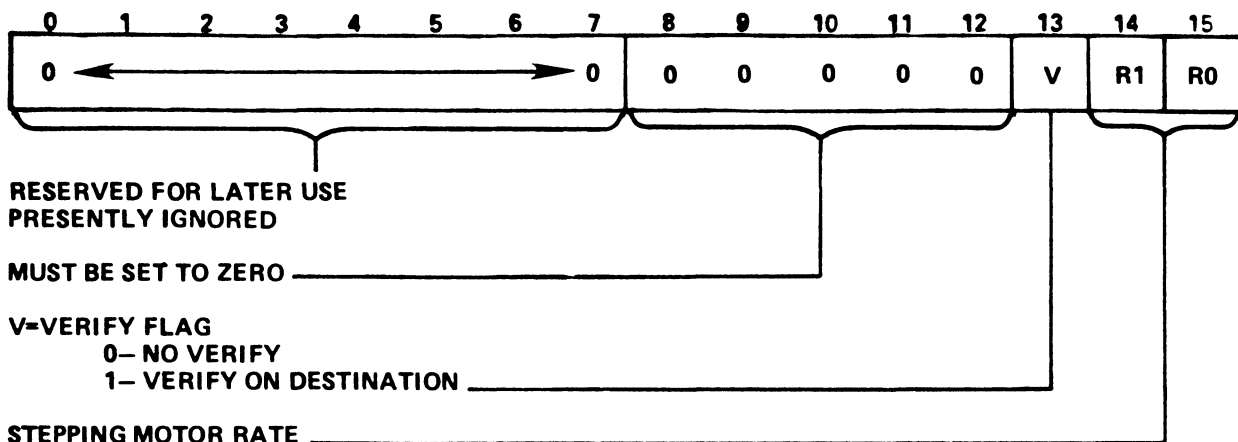
where ac = the accumulator.

5.3.2.1 Restore Command Bit Configuration

The Command Register uses the eight least significant bits of the accumulator word. Bits 8 through 12 are set to zero. Bit 13 is the Verify flag and bits 14 and 15 control the stepping motor rate. Figure 5-4 shows how these bits may be set.

NOTE

The setting of bits 14 and 15 depends on the type of drive used. Not all drives support the 3-millisecond stepping motor rate. See the specification supplied by the drive vendor.



BIT 14 (R1)	BIT 15 (R0)	TEST- = 1 (MILLISECONDS)	TEST- = 0 (MICROSECONDS)
0	0	3 ms	184 μ s
0	1	6 ms	190 μ s
1	0	10 ms	198 μ s
1	1	15 ms	208 μ s

081-79

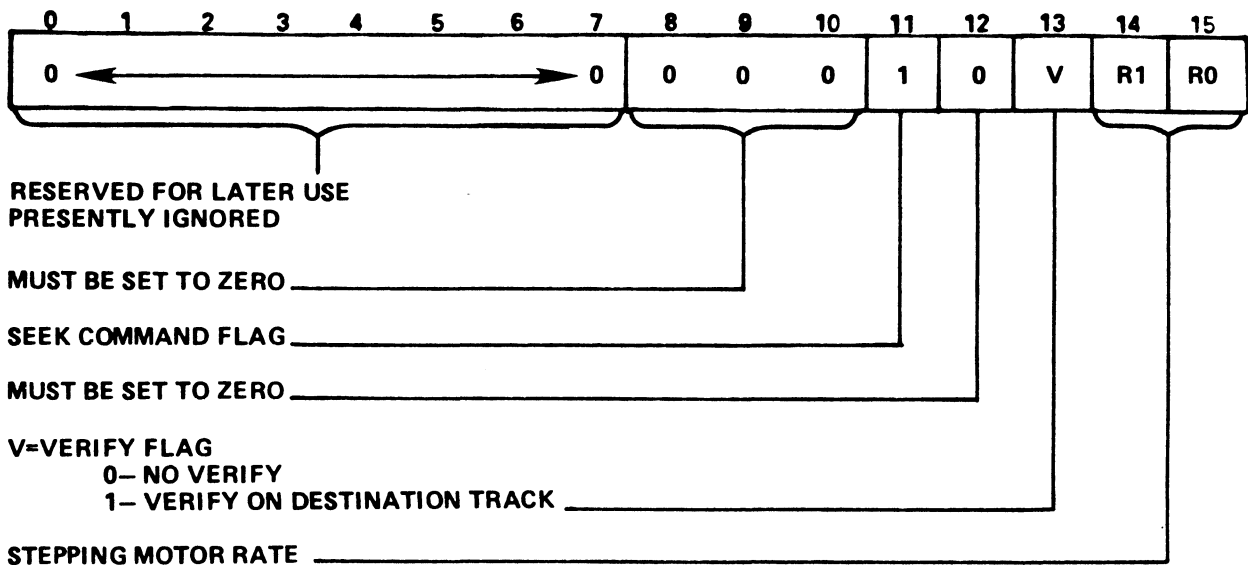
Figure 5-4. Restore Command Bit Configuration

5.3.2.2 Seek Command Bit Configuration

The Command Register uses the eight least significant bits of the accumulator word. Bits 8 through 10 are set to zero. Bit 11 is set to one, bit 12 to zero and bit 13 is the Verify flag. Bits 14 and 15 control the stepping motor rate. Figure 5-5 shows how these bits may be set.

NOTE

The setting of bits 14 and 15 depends on the type of drive used. Not all drives support the 3-millisecond stepping motor rate. See the specification supplied by the drive vendor.



BIT 14 (R1)	BIT 15 (R0)	TEST = 1 (MILLISECONDS)	TEST = 0 (MICROSECONDS)
0	0	3 ms	184 μ s
0	1	6 ms	190 μ s
1	0	10 ms	198 μ s
1	1	15 ms	208 μ s

081-80

Figure 5-5. Seek Command Bit Configuration

5.3.3 STATUS REGISTER

The Status Register contains information on the status of the drives after a command has been executed. It uses the same address as the Command Register (see Section 5.3.2). The eight least significant bits contain status information depending on the type of command. This status information is retrieved with a DIA instruction.

The instruction format is

DIA ac,34

where ac = the accumulator.

The format of the status word for data transfer commands is given in Figure 5-6; Table 5-3 defines the function of each bit.

The bit configuration for head movement commands is shown in Figure 5-7; Table 5-4 explains the bit functions.

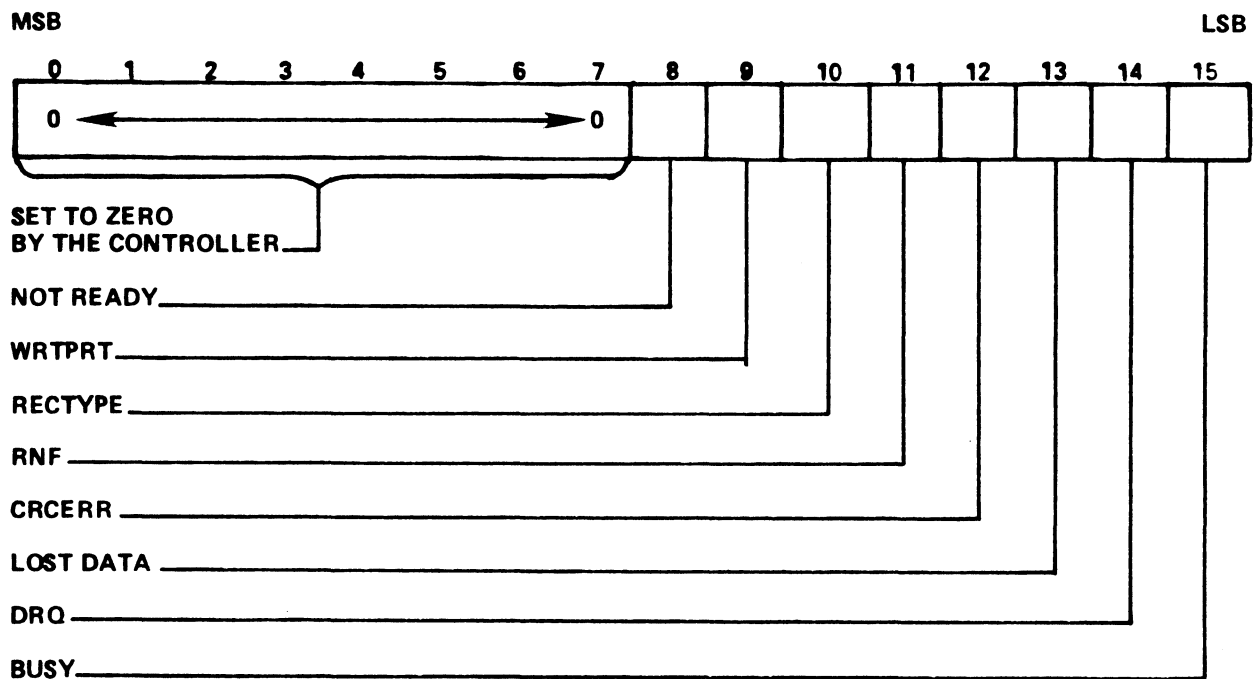


Figure 5-6. Drive Status Register Word For Data Transfer Commands

TABLE 5-3. DRIVE STATUS DEFINITIONS FOR DATA TRANSFER COMMANDS

Bits	Name	Function
0-7		Reserved, set to 0 by the controller
8	NOT READY	Unit Not Ready - When set to 0, the selected unit is ready. When set to 1, the unit is not ready. This bit is an inverted copy of the Ready input. Data transfer commands will not execute unless the requested drive is ready.
9	WRTPRT	Write Protect - Indicates that data may not be written into a file. Not used for Read Record or Read Track commands. Reset when the next command is received.
10	RECTYPE	Record Type - For Read Sector command only. A user-definable flag that indicates the record-type code from the Data Address Mark. 0 = Data Address Mark 1 = Deleted Data Address Mark Reset when the next command is received.
11	RNF	Record Not Found - Not used for Read Track or Write Track commands. When set to 1, indicates that the specified track, sector, or side was not found. Reset to 0 when the next command is received.
12	CRCERR	CRC Error - Not used for Read Track or Write Track commands. When set to 1 and bit 11 is also set to 1, indicates an error is found in one or more ID fields. When set to 1 and bit 11 = 0, indicates an error in the data field. Reset to 0 when the next command is received.
13	LOST DATA	Lost Data - When set to 1, indicates that the computer did not respond to the data request in one byte time. Reset to 0 when the next command is received.

**TABLE 5-3. DRIVE STATUS DEFINITIONS FOR DATA TRANSFER COMMANDS
(Cont)**

Bits	Name	Function
14	DRQ	Data Request - When set to 1, indicates that the Data Register is full on a read operation or it is empty on a write operation. This bit is reset to 0 when updated. This bit is a copy of the DRQ output.
15	BUSY	Busy - When set to 0, no command is being executed. When set to 1, a command is being executed.

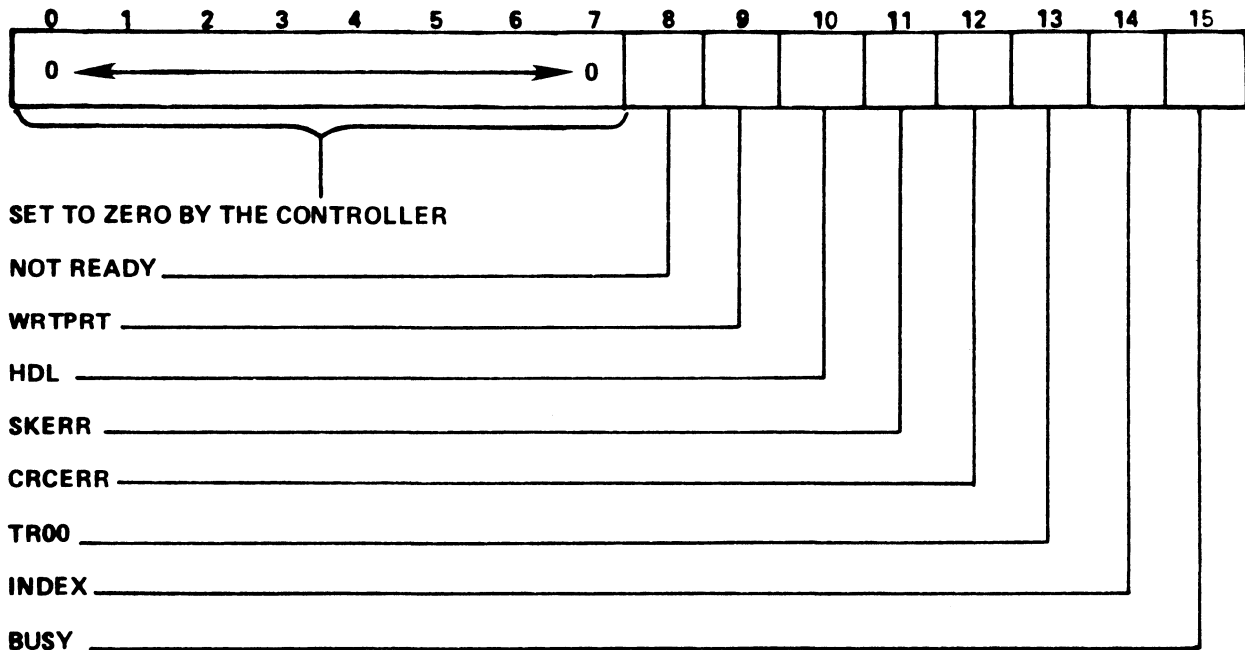


Figure 5-7. Status Register For Head Movement Commands

TABLE 5-4. DRIVE STATUS DEFINITIONS FOR HEAD MOVEMENT COMMANDS

Bits	Name	Function
0-7		Reserved, set to 0 by the controller
8	NOT READY	Unit Not Ready - When set to 0, the selected unit is ready. When set to 1, the unit is not ready. This bit is an inverted copy of the Ready input.
9	WRTPRT	Write Protect - When set to 1, indicates that Write Protect is activated. This bit is an inverted copy of WRTPRT-input.
10	HDL	Head Load - Not used. Select loads the heads at the drive end.
11	SKERR	Seek Error - When set to 1, indicates that the specified track was not verified. It is reset when updated.
12	CRCERR	CRC Error - When set to 1, an error was encountered in the ID field.
13	TR00	Track Number 00 - When set to 1, indicates that Read/Write heads are positioned on track zero. This bit is an inverted copy of TR00- input.
14	INDEX	Index - When set to 1, indicates that an Index Mark from the drive is detected. This bit is an inverted copy of the IP-input.
15	BUSY	Busy - When set to 0, no command is being executed. When set to 1, a command is being executed.

5.3.4 TRACK REGISTER

The Track Register holds the track number of the current Read/Write head position. It is incremented by one every time the head is stepped in (toward track 76) and decremented by one when the head is stepped out (toward track 00). The contents of the register are checked against the recorded track number in the ID field during disc Read, Write, and Verify procedures.

This register may be written to or read from. To write to the Track Register, use the format

DOA ac,35

To read from the register, use the format

DIA ac,35

where ac = the accumulator.

5.3.5 SECTOR REGISTER

The Sector Register contains a specified sector address. It may be written to or read from. The contents of this register are compared with the recorded sector number in the ID field during disc Read or Write operations. To write to the Sector Register, use the format

DOA ac,36

To read from the register, use the format

DIA ac,36

where ac = the accumulator.

5.3.6 DATA REGISTER

The Data Register is an 8-bit holding register used while disc Read or Write operations are in progress. When a Seek Command (see Section 5.3.2.2) is executed, the Data Register is used to hold the destination track of the Seek. During a Read operation, data is transferred into the register. During a Write operation, data is transferred from the register. All data transfers are handled by firmware in DMA mode. The Data Register may be written to and read from. To write to the register, use the format

DOA ac,37

To read from the register, use the format

DIA ac,37

where ac = the accumulator.

5.4 INPUT/OUTPUT CONTROL BLOCK (IOCB)

The input/output control block (IOCB) contains commands to be performed via direct memory access (DMA). The IOCB for the Floppy Disc Controller consists of three words located in the lower 32K of main memory. It is used to indicate the operational mode. The three words are set up by the software before an operation begins. The format of the Floppy Disc IOCB is shown in Figure 5-8.

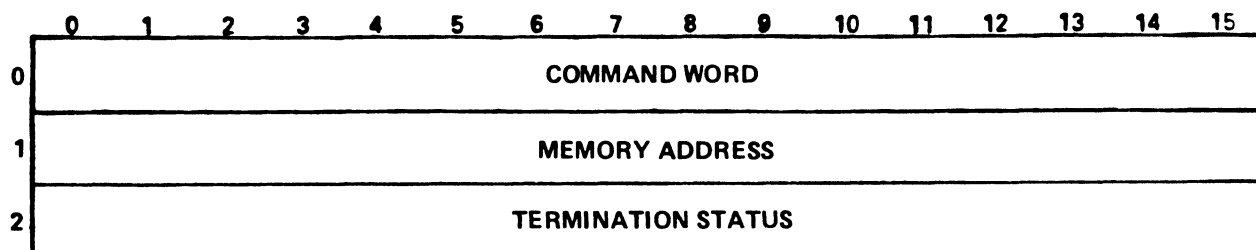


Figure 5-8. Floppy Disc IOCB Format

5.4.1 COMMAND WORD (WORD 0)

The command word (word 0 of the IOCB) is used to define five commands. These commands may be divided into three classes as follows:

1. Data Transfer Commands
 - Read Sector
 - Write Sector
2. Utility Commands
 - Read Address
 - Read Track
3. Format Command
 - Write Track

The following subsections discuss each command function and bit configuration.

5.4.1.1 Read Sector Command

The Read Sector command transfers data blocks from diskette to memory. The block size depends on the setting of the B-flag in the command word and the sector length code in the header (see Table 5-5). For IRIS-compatibility, the B-flag should be set to 1 and the sector length should be 02 (see Section 5.4.1.3). Upon receipt of the command, the Busy status bit is set. When an ID field is encountered with the correct track number, sector number, side number, and cyclic redundancy check (CRC), the data is transferred to memory.

TABLE 5-5. SECTOR LENGTH FLAG SETTING

B Flag Setting	Sector Length Code			
	00	01	02	03
0 - Single Density	256	512	1024	128
1 - Double Density	128	256	512*	1024

*Required for IRIS

The Data Address Mark must be found within 30 bytes (single density) or 43 bytes (double density) of the last ID field CRC byte. If it is not found, the Record Not Found (RNF) bit is set and the operation terminates.

Once the first byte reaches the Data Register, the Data Request bit (DRQ) is set. If the computer has not read the contents of the Data Register before a new byte is transferred, that byte is lost and the Lost Data status bit is set. If a CRC error occurred at the end of the data field, the CRC Error status bit is set and the operation terminates.

At the end of the Read operation, the type of Data Address Mark encountered in the data field is recorded in the Status Register (Deleted Data Mark = 1; Data Mark = 0).

The bit configuration and possible flag settings for the Read Sector command are shown in Figure 5-9. Bits 0-12 must be set as shown; bits 13 and 14 may be set as required; bit 15 is a user-defined flag. Table 5-6 defines the bit settings.

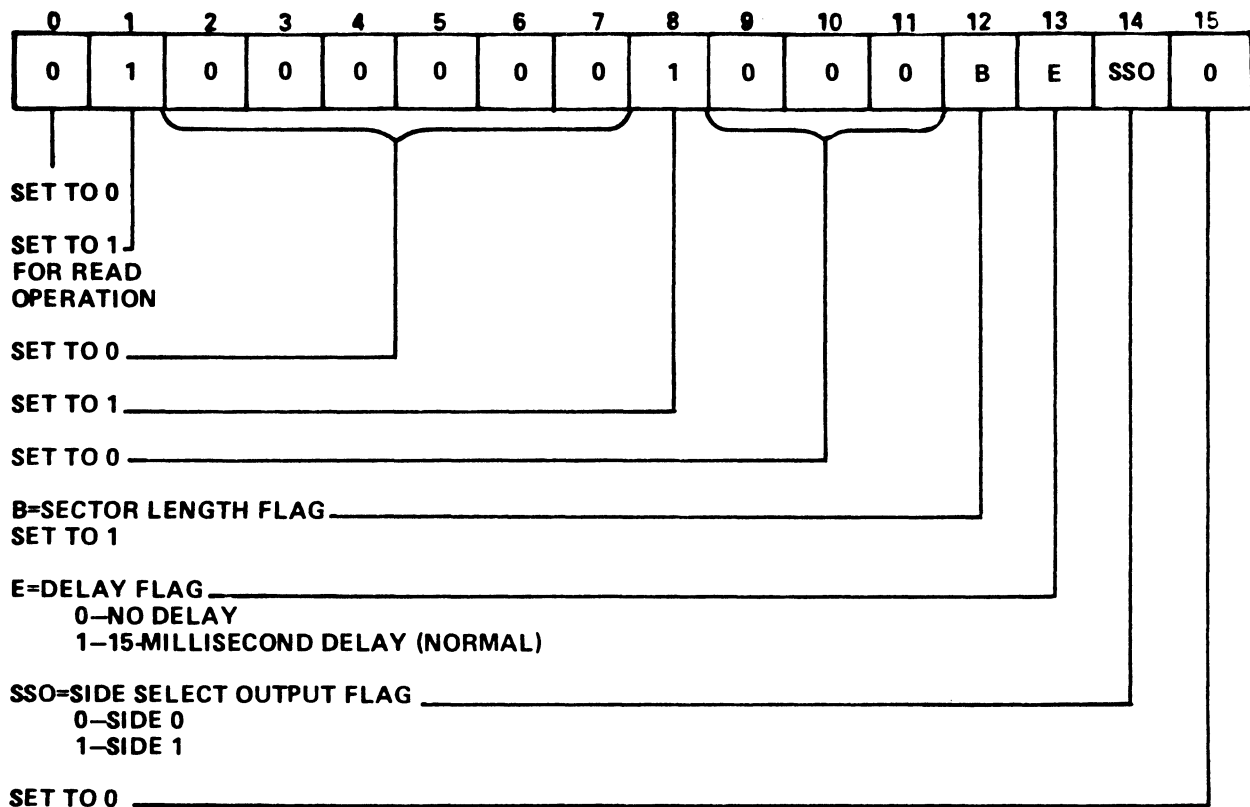


Figure 5-9. Read Sector Command Word Bit Configuration

TABLE 5-6. READ SECTOR COMMAND BIT CONFIGURATION

Bit	Definition
0	Set to 0
1	Set to 1 for Read operation
2-7	Set to 0
8	Set to 1
9-11	Set to 0
12	Side Select Output Flag (SSO). This flag is used in address verification. 0 - Side 0 1 - Side 1
13	Delay Flag (E). 0 - No delay 1 - 15-millisecond delay (normal)
14	Sector Length Flag (B). This flag, in conjunction with the sector length code in the header, signals the block size to be transferred. For IRIS compatibility, set to 1.
15	Data Address Mark Flag. This is a user-defined flag. Set to 0.

5.4.1.2 Write Sector Command

The Write Sector command transfers data blocks from memory to diskette. The block size depends on the setting of the B-flag in the command word and the sector length code in the header (see Table 5-5). For IRIS-compatibility, the B-flag should be set to 1 and the sector length should be 02 (see Section 5.4.1.3). Upon receipt of the command, the Busy status bit is set. When an ID field is encountered that has the correct track number, sector number, side, and cyclic redundancy check (CRC), a Data Request (DRQ) is generated. 11 bytes for single density or 22 bytes for double density are counted off and output is activated if the Data Register has been loaded. If the DRQ is not serviced, the operation terminates and the Lost Data bit is set. The Data Address Mark is written on diskette. It is set by bit 15 and is a user-defined flag (1 = Deleted Data Mark, 0 = Data Mark).

The data field is written and the DRQ is generated to the computer. If the DRQ is not serviced in time for continuous writing, the Lost Data bit is set and a byte of zeros is written to the diskette. The command is not terminated.

Writing is inhibited when the Write Protect logic is low. The command is terminated, an interrupt generated, and the Write Protect status bit is set.

The bit configuration and possible flag settings for the Write Sector command are shown in Figure 5-10 and defined in Table 5-7.

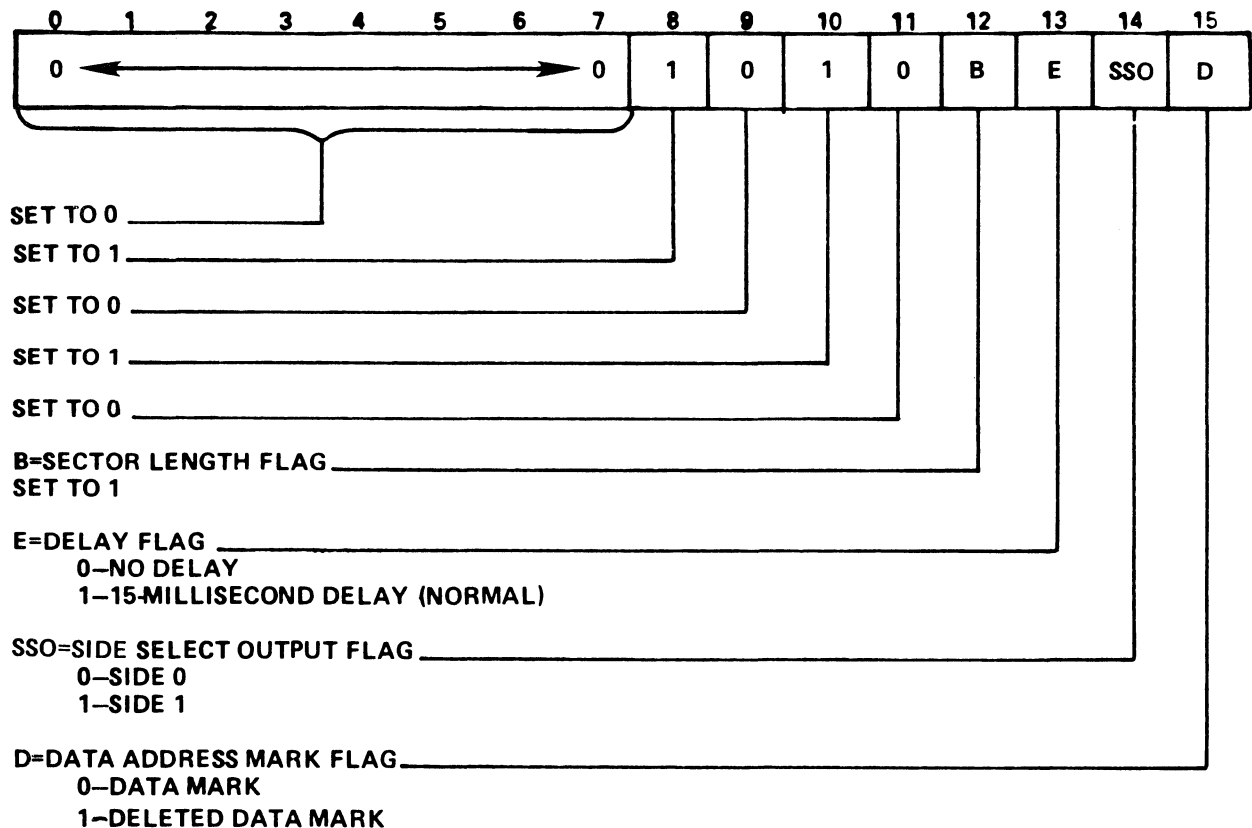


Figure 5-10. Write Sector Command Word Bit Configuration

TABLE 5-7. WRITE SECTOR COMMAND BIT CONFIGURATION

Bit	Definition
0-7	Set to 0
8	Set to 1
9	Set to 0
10	Set to 1
11	Set to 0
12	Side Select Output Flag (SSO). This flag is used in address verification. 0 - Side 0 1 - Side 1
13	Delay Flag (E). 0 - No delay 1 - 15-millisecond delay (normal)
14	Sector Length Flag (B). The B-flag, in conjunction with the sector size code in the header, signals the block size to be transferred. For IRIS compatibility, set to 1.
15	Data Address Mark Flag (D). A user-defined flag, it is set by this bit and recognized by Read Sector command. 0 - Data Mark 1 - Deleted Data Mark

5.4.1.3 Read Address Command

The Read Address command is used to check the sector header of the diskette. Upon receipt of the command, the Busy status bit is set. The next encountered ID field is read in from the diskette. Six data bytes of the ID field (see Figure 5-11) are assembled, transferred to the Data Register, and a Data Request (DRQ) is generated for each byte. Cyclic redundancy check (CRC) characters are transferred to the CPU but they are also checked for validity. The CRC Error bit is set if an error is detected. The track address is written to the Sector Register. When the operation is completed, an interrupt is generated and the Busy bit reset.

For the Read Address command, the bit configuration and possible flag setting for bit 13 are shown in Figure 5-12.

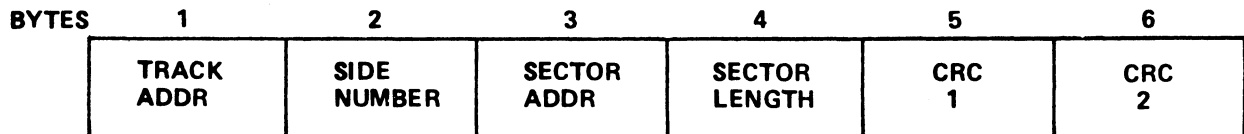


Figure 5-11. ID Field

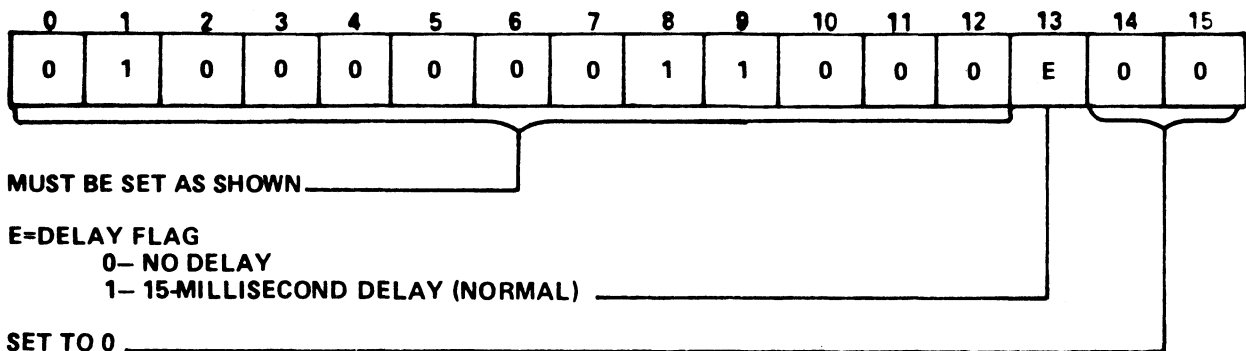


Figure 5-12. Read Address Command Word Bit Configuration

5.4.1.4 Read Track Command

The Read Track command is used to check the format of a diskette. Upon receipt of a Read Track command, the Busy status bit is set. The track is read starting with the first index pulse encountered and continues until the next index pulse is found. As each byte is assembled, it is transferred to the Data Register and the Data Request (DRQ) is generated for each byte. Cyclic redundancy check (CRC) is not performed. Gaps are included in the input data stream. The accumulation of bytes is synchronized to each Address Mark encountered. Upon completion the interrupt is activated.

For the Read Track command, the bit configuration and possible settings for bit 13 are shown in Figure 5-13.

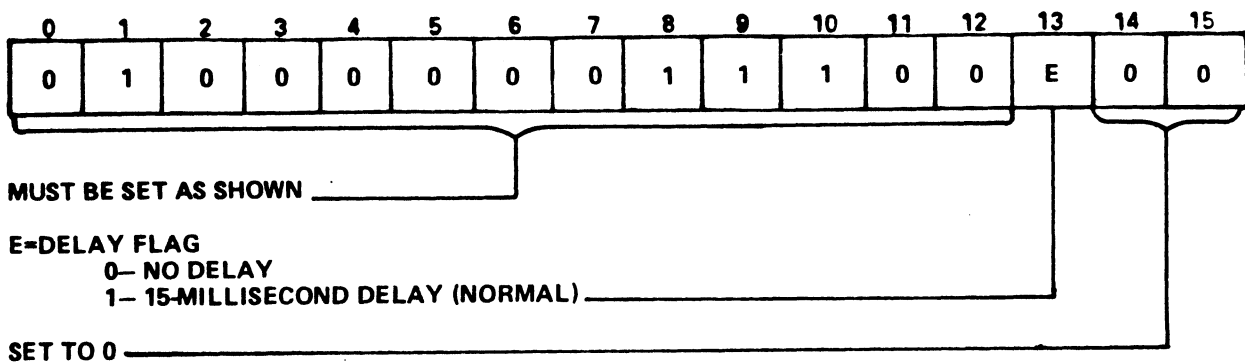


Figure 5-13. Read Track Command Word Bit Configuration

5.4.1.5 Write Track Command

The Write Track command is used to format a diskette by transferring an image of the memory buffer to the diskette. Upon receipt of the command, the Busy status bit is set. Writing starts with the first index pulse encountered and continues until the next one is found. Data Request (DRQ) is activated upon receipt of the command but writing does not start until the first byte is loaded into the Data Register. If the register is not loaded when the first index pulse is encountered, the operation terminates and the Busy bit is reset, the Lost Data bit is set, and interrupt is enabled.

If a byte is not in the Data Register when it is needed, a byte of zeros is substituted. Address Marks and CRC characters are written to the diskette when certain data byte patterns (see Section 5.6.1) are detected in the outgoing data stream.

The bit configuration and possible settings for bit 13 are shown in Figure 5-14.

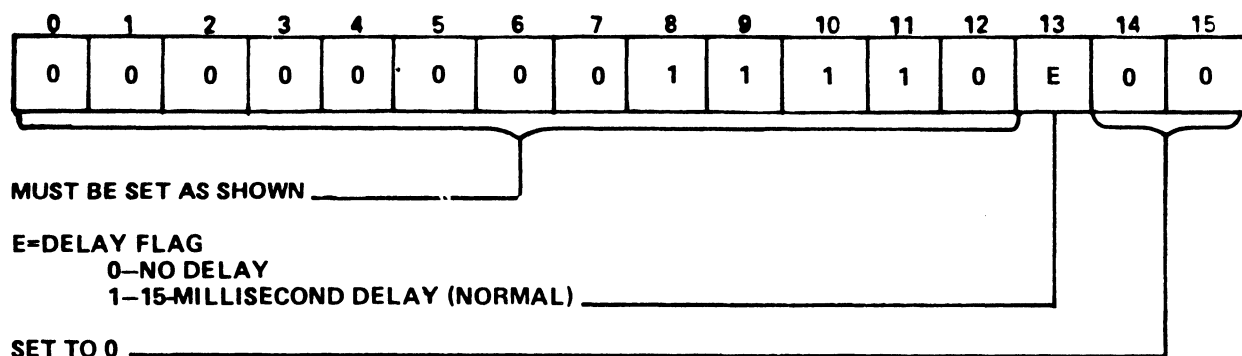


Figure 5-14. Write Track Command Word Bit Configuration

5.4.2 MEMORY ADDRESS (WORD 1)

The 16-bit memory address word is used to specify the word address of the memory buffer. The format of the memory address word is shown in Figure 5-15.

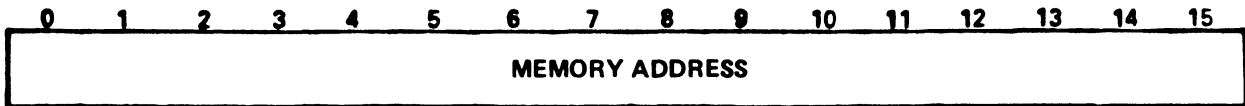


Figure 5-15. Memory Address Word Bit Configuration

5.4.3 TERMINATION STATUS WORD (WORD 2)

Upon completion of an operation, the controller writes the termination status into word 2 of the IOCB. The termination status provides information on the result of the operation. The information is presented in the eight least significant bits of word 2. The bit configuration depends on the type of command issued (see Figure 5-16).

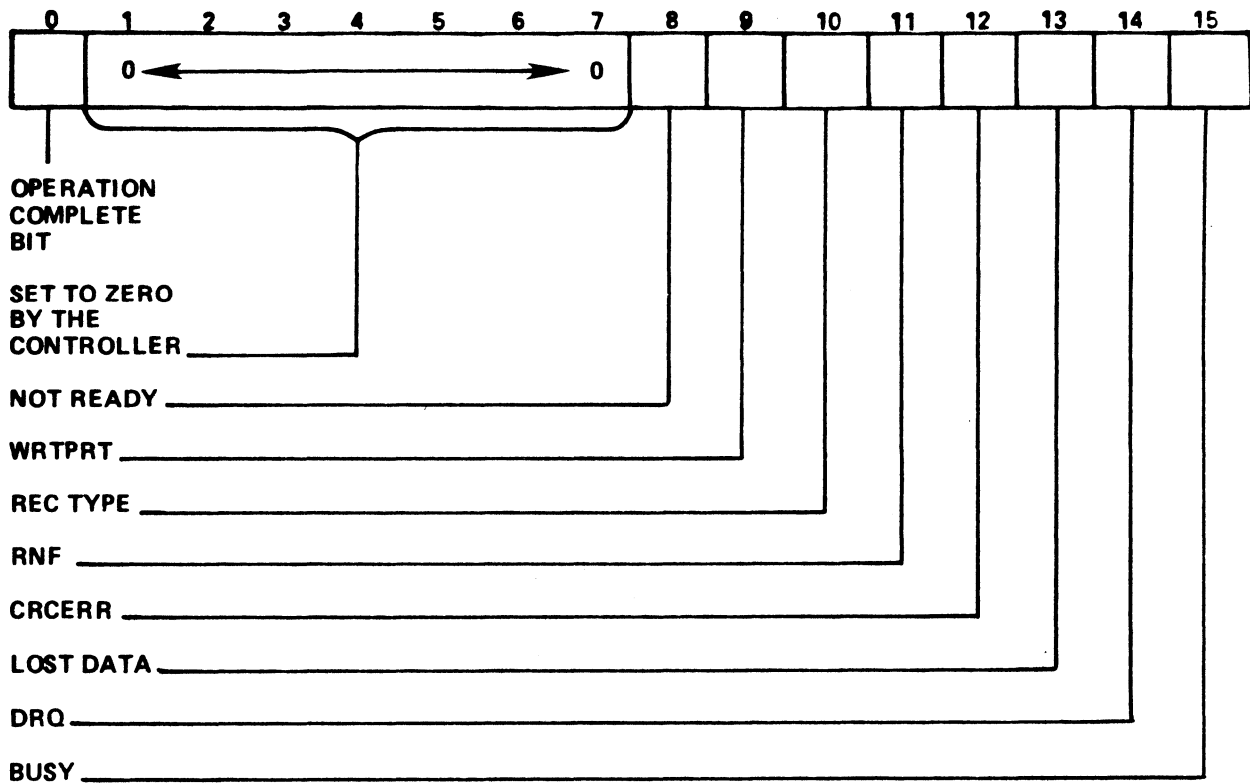


Figure 5-16. Termination Status Word Format

5.4.4 FLOPPY DISC CONTROLLER INPUT CONTROL

A dedicated register in the processor serves as a pointer to the floppy disc IOCB. This register may be written into or read from by using the DOBP or DIBP instructions, respectively. The P control code is used to specify command instructions directed to the floppy disc controller (rather than the multiplexer (S) or disc controller (C)). The format of the floppy disc controller input control instruction is

DIBP ac,77

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

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

5.4.5 FLOPPY DISC CONTROLLER OUTPUT CONTROL

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

DOBP ac,77

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

1. Activating the Floppy Disc Controller - The Floppy Disc Controller is activated by a DOBP 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 Floppy Disc Controller - After the Floppy Disc Controller completes an operation and generates an interrupt, it must be deactivated by the software. A DOBP instruction with the accumulator set to zero deactivates the controller. This resets the interrupt-pending bit and places the Floppy Disc Controller in an idle state.

5.5 READ/WRITE HEAD POSITIONING

The Read/Write heads may be positioned by five instructions:

1. Seek
2. Restore
3. Step
4. Step In
5. Step Out

For the MARK 3B, only the Seek and Restore commands are required and are discussed here. Users who wish to use the Step commands, refer to Western Digital's FD 1797 Data Sheet.

When a Seek or Restore is executed, an optional verification can be performed. This option depends on the setting of the Verify flag in bit 13 of the command word (see Figure 5-4 for Restore, Figure 5-5 for Seek). If bit 13 is set to 1, the verification option is executed.

Verification begins at the end of the 15-millisecond settling time after the head is loaded against the media. The track number of the first encountered ID field is compared against the contents of the Track Register. If the track numbers correspond and the ID field Cyclic Redundancy Check (CRC) is correct, the verify operation completes. If the track numbers are not the same, the ID field, or the correct CRC are not found within five revolutions of the diskette, a SEEK error and an interrupt are generated.

The Read/Write heads are loaded when the Drive Select Signal is received.

5.5.1 SEEK OPERATION

A Seek command is performed in two steps. First load the appropriate track number into the Data Register (device code = 37). Then load the Seek command into the Command Register (device code = 34). At the completion of the Seek, the Track Register (device code = 35) contains the current value of the head position and an interrupt is generated. The Seek command assumes that the initial value of the Track Register is a true reading of the Read/Write heads. If interrupts are not to be used, the completion of the Seek can be polled by testing the interrupt-pending bit as follows:

```
DIBP ac,77
```

where ac = accumulator.

Seek is completed when the interrupt-pending bit (bit 0) = 1.

5.5.1.1 Seek Command Initiation

An example of a Seek command sequence follows:

```
LDA 0,track number ;Load track no. into register 0
DOA 0,37 ;Issue track no. to Data Register
LDA 1,SEEK command ;Load SEEK command into register 1
DOA 1,34 ;Issue SEEK command to Command Register
DIBP 2,17 ;Read interrupt pending bit in register 2
SSN 2,2 ;Test if set
JMP .-2 ;Repeat until set
DIA 3,34 ;Read status
•
•
• ;Check for error
•
•
```

Figure 5-17 is a flowchart of Select and Seek operations.

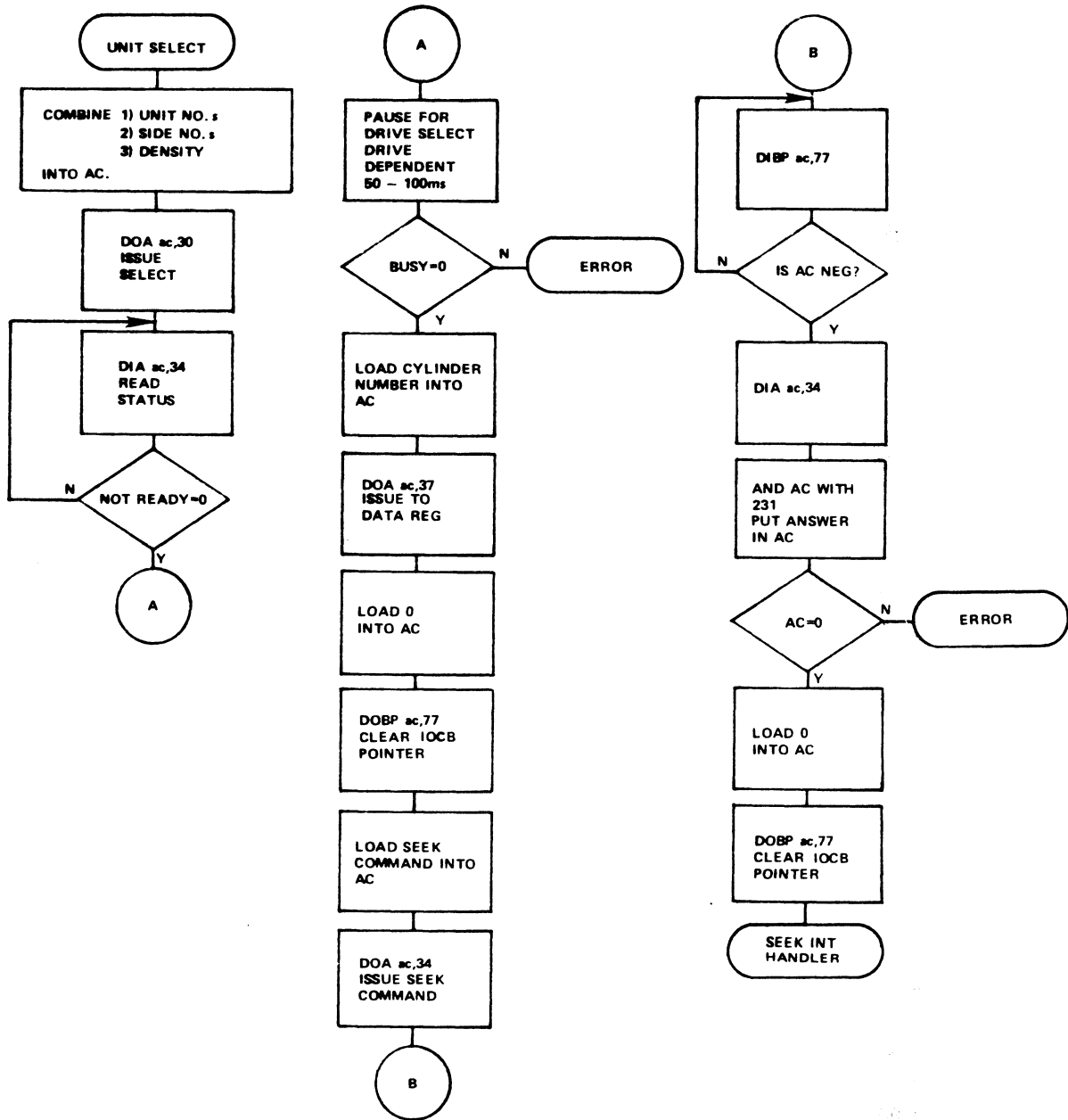


Figure 5-17. Select and Seek Operation Flowchart

5.5.1.2 Seek Error Recovery

A Seek error occurs if one of the following conditions exists:

- Wrong track number was issued
- Cyclic redundancy check is incorrect
- ID field was not found within 5 revolutions of the diskette

If a Seek error occurs (SKERR/RNF, bit 11 and/or CRCERR, bit 12 in the Status Register), restore and reissue the command. The bits are reset when the next command is received.

5.5.2 RESTORE OPERATION

A Restore command causes the Read/Write heads to be positioned over track 00 (TR00- flag is set in the Status Register).

A Restore (recalibrate, return to 0) is done automatically when the RESET button on the mini-panel is pressed.

To effect a Restore without use of the RESET button, issue the following command:

```
DOA ac,34
```

where

```
ac - accumulator where a valid restore command has been  
    loaded  
34 - Command Register device code
```

5.6 FORMATTING DISKETTES

The Write Track command initiates the formatting process. For IRIS-compatibility, 8-inch diskettes are formatted with 15 sectors for double density and 5-1/4-inch diskettes are formatted with 8 sectors. The required block size is 512 bytes, double density, in IBM-type format. Users of other operating systems have a choice of formats and block sizes. Blocks may contain 128, 256, 512, or 1024 bytes.

Sector lengths are entered at format time via the following decimal codes:

00	=	128 bytes
01	=	256 bytes
02	=	512 bytes (normal for IRIS)
03	=	1024 bytes

Formatting is done one track at a time. When the appropriate track has been accessed by a Seek command, the Write Track command writes an image of the memory buffer. The memory address word in the IOCB points to the starting location in the memory buffer.

The command format is as follows:

IOCB: Write Track	
Starting memory address	(word address)
Termination word	(set by firmware)

5.6.1 IBM-TYPE FORMATS

IBM-type formats are recommended. For IRIS users, the diskettes should be formatted into IBM-type, double-density format using 512-byte tracks.

The format for a track containing 512-byte sectors is shown in Figure 5-18.

The pattern shown in the Sector Table of Figure 5-18 should be repeated 15 times for double density, 8 times for single density, or until the controller interrupts out (approximately 598 bytes).

PREAMBLE TABLE (1 OF 1)

	<u>Octal Constant (Word)</u>	<u>Octal Count (Words)</u>		
GAP IV (PRE-INDEX GAP)	47116	50	(116-116)	
	0	6	(0-0)	
	173366	1	(366-366)	
	173374	1	(366-374)	374= INDEX MARK
GAP I (POST-INDEX GAP)	47116	31	(116-116)	

SECTOR TABLE (1 OF 17 OCTAL)

	<u>Octal Constant (Word)</u>	<u>Octal Count (Words)</u>		
	0	6	(0-0)	
	172765	1	(365-365)	
	172776	1	(365-376)	376= ADDRESS MARK
	0*	1	TRACK-SIDE	
	2*	1	SECTOR-2**	
	173516	1	(367-116)	367= CRC***
GAP II (ID GAP)	47116	12	(116-116)	
	47000	1	(116-0)	
	0	5	(0-0)	
	365	1	(0-365)	
	172765	1	(365-365)	
	175400	1	(373-0)	373= DATA ADDRESS MARK
	0	377	(0-0)	
	367	1	(0-367)	367= CRC***
GAP III (DATA GAP)	47116	33	(116-116)	

POSTAMBLE TABLE (1 OF 1)

	<u>Octal Constant (Word)</u>	<u>Octal Count (Words)</u>	
GAP IV (PRE-INDEX GAP)	47116	453	(116-116)

- *These numbers are sector, track and side dependent.
- **512-byte flag.
- ***Two CRCs inserted automatically by the 1797 chip.

Figure 5-18. 512-Byte Sector Format

5.6.2 NON-IBM FORMATS

IBM formats may be modified to a limited extent if the following requirements are met:

- Sector size must be 128, 256, 512, or 1024 bytes.
- Gap sizes must meet the minimum requirements shown in Table 5-8.

Note that the Index Mark is not required. For Phase Lock Loop (PLL) lock-up time, motor speed variations, etc., additional bytes must be added to the minimum gap sizes given in Table 5-8.

TABLE 5-8. GAP SIZES

Gap Number	Single Density		Double Density	
	Size (For FM)	Hexa-decimal	Size (For MFM)	Hexa-decimal
I	16 bytes	FF	32 bytes	4E
II	11 bytes	FF	22 bytes	4E
*	6 bytes	00	12 bytes	00
			3 bytes	A1
III	10 bytes	FF	24 bytes	4E
**	4 bytes	00	8 bytes	00
			3 bytes	A1
**	4 bytes	00	8 bytes	00
IV	16 byte	FF	16 bytes	4E

*Byte count must be exact.

**Byte counts are minimum, but exactly 3 bytes of A1 must be written.

1000

1000

1000

Section 6

ST506 (WD1002-5) INTERFACE

6.1 INTRODUCTION

This section describes the interface for Western Digital's WD1002-5 Disc Controller which has been integrated into the POINT 4 MARK 2 Peripheral Interface Board (PIB).

The WD1002-5 Disc Controller can interface up to three ST506-type drives and up to four floppy disc drives. POINT 4's MARK 2 uses one port to interface the ST506 (5-1/4-inch Winchester-type) disc drive and one port for the floppy disc drive.

For a detailed discussion of the WD1002-5 Controller, refer to Western Digital's Winchester/Floppy Disk Controller OEM Manual.

The WD1002-5 Disc Controller has the following features:

- ST506 disc operations
- MFM encode and decode logic
- Side select
- Address mark detection
- 8-bit bidirectional bus
- Control-block-oriented instructions
- Software control of sector addressing
- Programmable sector sizes (128, 256, 512, or 1024 bytes)
- Read/write head position verification
- ST506 disc operation handled via an I/O control block (IOCB)
- I/O instructions using device codes 50 to 57
- Data transfer performed in DMA mode (device code 77)
- Status and error reporting on completion of operation
- CRC (cyclic redundancy check) generation and detection

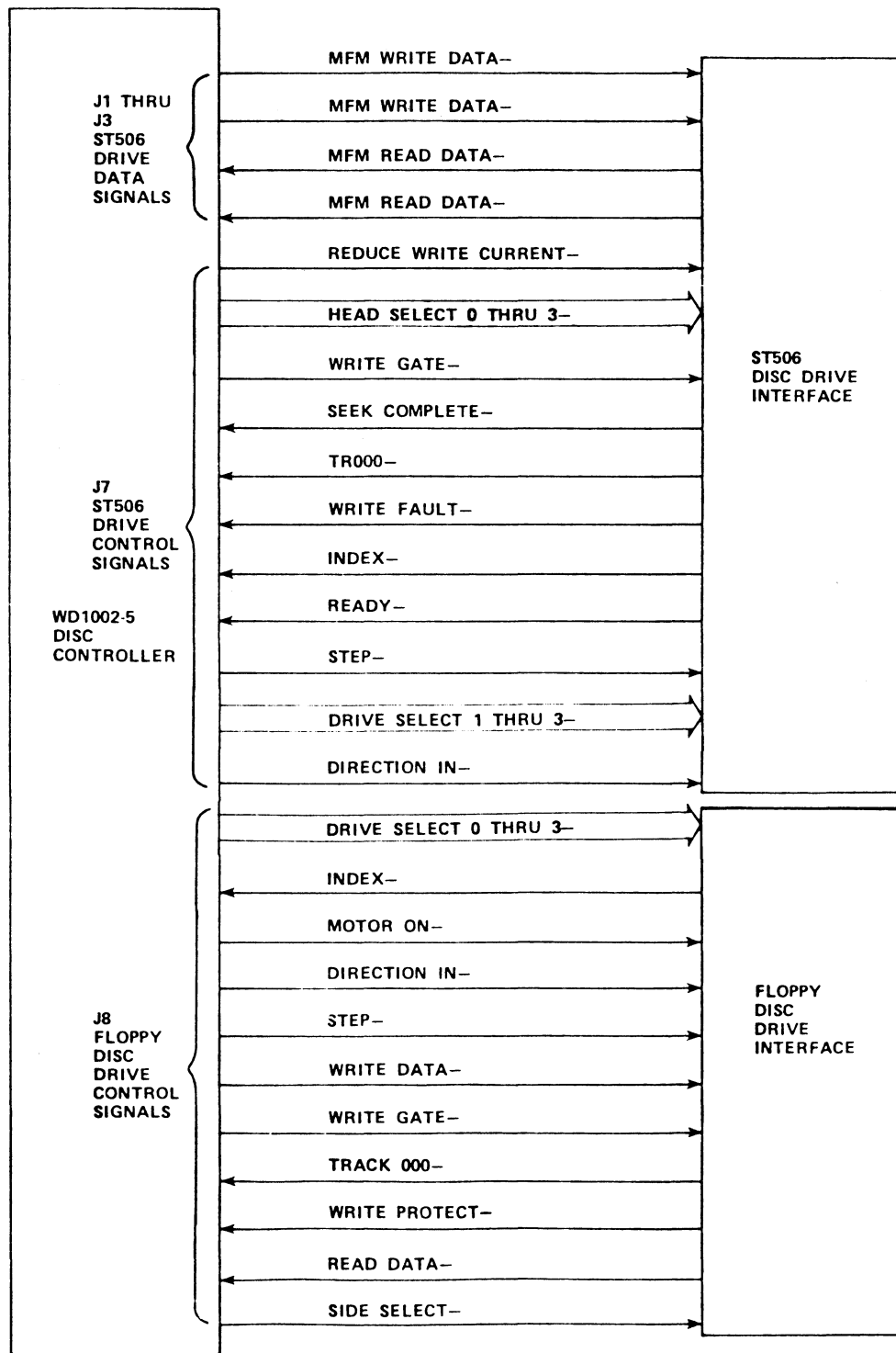
- Error detection and correction (ST506 only)
- Internal diagnostics
- 16 different stepping rates
- Drive port assignments:
 - J7 for ST506 disc drive
 - J8 for floppy disc drive

6.2 MARK 2 INTERFACE

The MARK 2 PIB interfaces the MARK 3 CPU and the WD1002-5 peripherals via seven connectors as follows:

J5	MARK 3 CPU 40-pin connector
J6	4-pin power connector
J7	ST506 disc drive 34-pin connector for control signals
J8	Floppy disc drive 34-pin connector for control signals
J1, J2, J3	ST506 20-pin high-speed data connectors which communicate with each disc drive. They may not be multiplexed. Refer to Western Digital's Winchester Disk/Floppy Controller Manual for more information.

Figure 6-1 illustrates WD1002-5 disc controller interface signals. Table 6-1 describes ST506 drive control signals. Table 6-2 describes ST506 drive data signals. Table 6-3 describes floppy disc drive control signals.



081-94

Figure 6-1. WD1002-5 Disc Controller Interface Signals

TABLE 6-1. ST506 DRIVE CONTROL (J7) SIGNALS

Signal Ground	Signal Pin	Direction	Description
1	2	OUT	Reduce Write Current - When activated with the Write Gate, a lower write current is used to compensate for greater bit-packing density on the inner cylinders. Activated when the cylinder number is greater than or equal to four times the contents of the write precomp register. Output is valid only during write and format commands.
3	4	OUT	Head Select 2 - Selects heads on drive number 3.
5	6	OUT	Write Gate - Enables write to disc.
7	8	IN	Seek Complete - Signals that the head of a selected drive has reached the specified cylinder and has stabilized. The signal is not checked after a seek command and overlapped seeks are allowed.
9	10	IN	Track 000 - Indicates that the read/write heads are positioned on track 0. The signal is sampled before each step pulse is issued.
11	12	IN	Write Fault - Indicates that a fault condition exists on the selected drive. Commands are not executed while this signal is true.
13	14	OUT	Head Select 0 - Selects read/write heads on drive number 1.
15	16		Not connected
17	18	OUT	Head Select 1 - Selects read/write heads on drive number 2.

TABLE 6-1. ST506 DRIVE CONTROL (J7) SIGNALS (Cont)

Signal Ground	Signal Pin	Direction	Description
19	20	IN	Index - Indicates the index point for synchronization during formatting. Also used as a timeout mechanism for retries. Should pulse once every rotation of the disc.
21	22	IN	Ready - Signals that a selected drive is up to speed. No commands are executed unless this signal is true.
23	24	OUT	Step - Pulsed once for every cylinder to be stepped. The direction of the step is determined by Direction In. The step pulse period is determined by the internal stepping rate register during implied seek operations, or explicitly during seek commands. During auto-restore, the step pulse period is determined by the seek complete time from the drive.
25	26	OUT	Drive Select 1 - Selects drive number 1.
27	28	OUT	Drive Select 2 - Selects drive number 2.
29	30	OUT	Drive Select 3 - Selects drive number 3.
31	32		Not connected
33	34	OUT	Direction In - Specifies the direction of the read/write heads when Step is pulsed. When high, the direction is out; when low, the direction is in.

TABLE 6-2. ST506 DRIVE DATA (J1 THRU J3) SIGNALS

Signal Ground	Signal Pin	Direction	Description
2	1		Not connected
4	3		Not connected
8	7		Not connected
	9		Not connected
	10		Not connected
11			Ground
12			Ground
	13	OUT	MFM Write Data
	14	OUT	MFM Write Data - Differential Write Data lines
15			Ground
16			Ground
	17	IN	MFM Read Data
	18	IN	MFM Read Data - Differential Read Data lines
19			Ground
20			Ground

TABLE 6-3. FLOPPY DISC DRIVE CONTROL (J8) SIGNALS

Signal Ground	Signal Pin	Direction	Description
1	2		Not connected
3	4		Not connected
5	6	OUT	Drive Select 0 - Selects heads on drive 0.
7	8	IN	Index - Pulses once every disc rotation to indicate the beginning of a track.
9	10	OUT	Drive Select 1 - Selects heads on drive 1.
11	12	OUT	Drive Select 2 - Selects heads on drive 2.
13	14	OUT	Drive Select 3 - Selects heads on drive 3.
15	16	OUT	Motor On - Controls the dc spindle motor of the floppy drive. If Motor On Mode (MOM) - 0 (selectable jumper option on drive) then a 40-nanosecond delay occurs. Otherwise, a one-second delay occurs after Motor On and before any reading or writing is attempted. If the floppy drive is not accessed in approximately 3 seconds, the motor is turned off by the controller.
17	18	OUT	Direction In - Selects direction of read/write heads. When high, direction is out; when low, direction is in.

TABLE 6-3. FLOPPY DISC DRIVE CONTROL (J8) SIGNALS (Cont)

Signal Ground	Signal Pin	Direction	Description
19	20	OUT	<p>Step - Pulsed once for each cylinder to be stepped. Determines the direction of the step. The internal floppy disc stepping rate register determines the step pulse period during the following operations:</p> <ul style="list-style-type: none"> ● Implied seek ● Auto restore ● Seek and restore commands <p>The stepping-rate period is limited to 8 microseconds minimum during any restore operation.</p>
21	22	OUT	Write Data - Transmits write data. Enabled when Write Gate is enabled.
23	24	OUT	Write Gate - Enables disc write circuitry.
25	26	IN	Track 000 - Indicates that the read/write heads are positioned on the outermost cylinder. This line is sampled before each step is issued.
27	28	IN	Write Protect - Indicates a write-protected disc is in the drive.
29	30	IN	Read Data - Transmits clock and data to controller.
31	32	OUT	Side Select - Selects side of floppy disc to be written or read.

6.3 PROGRAMMED INPUT/OUTPUT

Two modes of operation are standard with the ST506/Floppy Disc Controller: programmed I/O and DMA (Direct Memory Access).

Three command formats are used to communicate with the controller via eight registers in the following format:

DIA ac,xx (Read from ST506/Floppy Controller register)
DOA ac,xx (Write to ST506/Floppy Controller register)
DOBC ac,77 (Initiate data transfer in DMA mode)

where

ac - accumulator
xx - device code

The device codes used by the ST506/Floppy Disc Controller range from 50 to 57:

<u>Device Code</u>	<u>DOA Instruction</u>	<u>DIA Instruction</u>
50	Data Register	Data Register
51	Write Precomp Register	Error Register
52	Sector Count Register	Sector Count Register
53	Sector Number Register	Sector Number Register
54	Cylinder Low Register	Cylinder Low Register
55	Cylinder High Register	Cylinder High Register
56	Size/Drive/Head Register	Size/Drive/Head Register
57	Command Register	Status Register

The first digit (5) in the device code selects the WD1002-5 controller board.

NOTE

For normal read/write operations, the following commands are used:

DIA ac,51 (Get error register if there is an error)
DOA ac,56 (Select drive)
DOBC ac,77 (Start transfer)

All other commands are provided for diagnostic purposes only.

The following subsections provide information on ST506/Floppy Disc programmed I/O and the use of the registers.

6.3.1 DATA REGISTER

The Data Register is an 8-bit holding register that contains the next byte of data to be written to or read from the internal sector buffer. The sector buffer contains data to be read during a Read Sector command. When a Write Sector or Format Track command has been issued, the sector buffer awaits the data to be written. All data transfers are handled by firmware in DMA mode.

The Data Register may not be written to or read from except in the context of a valid command. The syntax for a read or format command is

DOA ac,50

The syntax for a read command is

DIA ac,50

where ac - the accumulator

6.3.2 WRITE PRECOMP REGISTER

The Write Precompensation and the Error Registers (see Section 6.3.3) are physically the same 8-bit register. During a write or format command, the Write Precomp Register contains the cylinder address where the lower write current is to be turned on (see Table 6-1) and write precompensation is to start.

The lower write current is activated when the cylinder being addressed is four times greater or equal to the value contained in this register. For example, if write precompensation is required for cylinder number 128 (80 hex) or higher, the register must be loaded with the value of 32 (20 hex).

Write precompensation is not used for floppy discs.

The register can only be written to; it cannot be read from. The command format is

DOA ac,51

where ac - the accumulator

6.3.3 ERROR REGISTER

The 8-bit Error Register contains fault information generated by the last command executed. The information in the register is valid only if the error bit in the Status Register is set, except when a Test command was issued (see Section 6.3.8.1). If the last command executed was a Test command, the error bit in the Status Register is not set but a binary diagnostic error code is loaded into the Error Register. This code indicates which device on the WD1002-5 board was in error as shown in Table 6-4.

TABLE 6-4. DIAGNOSTIC ERROR CODES

Code	Description
5	WD1015 Error
4	WD1014 Error
3	Sector Buffer Error
2	WD1010 Error
1	WD2797 Error
0	OK - WD1002-5 is functional

The Error Register can be read from; it cannot be written to.
The command format is

DIA ac,51

where ac - the accumulator

The bit configuration of the Error Register is shown in Figure 6-2; Table 6-5 defines each bit in the accumulator.

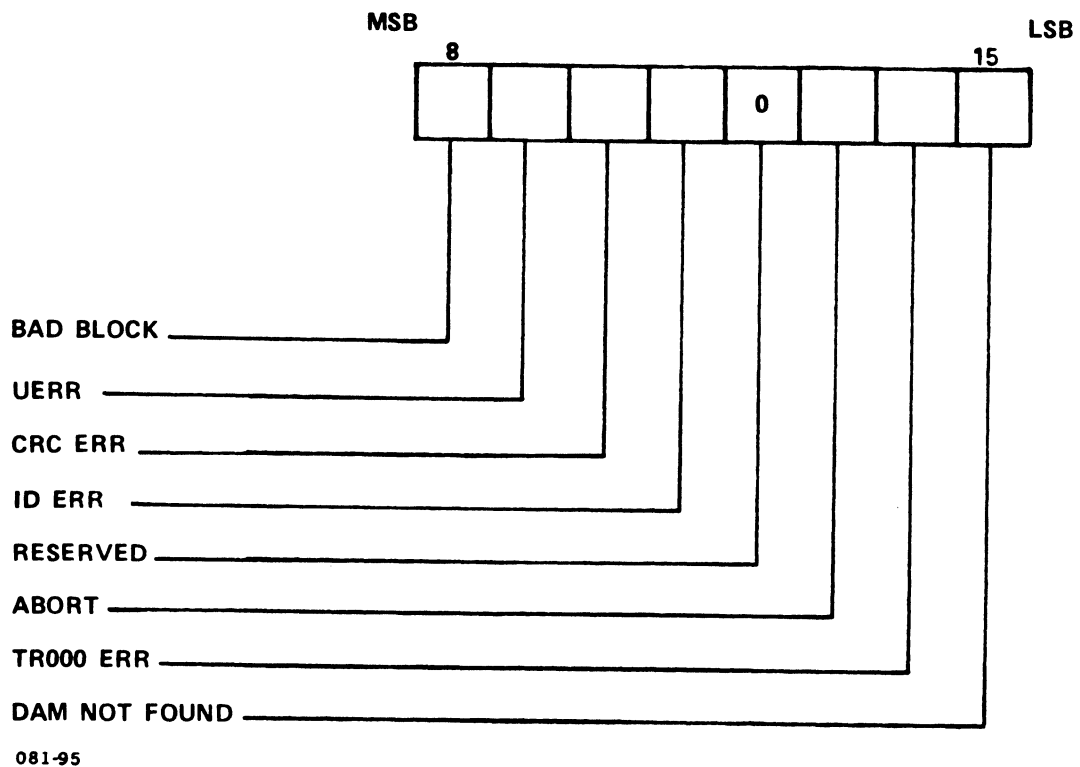


Figure 6-2. Error Register Bit Configuration

TABLE 6-5. ERROR REGISTER BIT DEFINITIONS

Bits	Name	Function
8	BAD BLOCK	Bad Block Detect - When set to 1, indicates that a bad block mark was detected in the specified ID field. If a write sector command was issued, the Write gate may be pulsed but the sector will not be written. If a read sector command was issued, the data field will not be read. The bad block may not be detected if there is a flaw in the ID field.
9	UERR	Uncorrectable Error - When set to 1, indicates that an ECC or CRC error was encountered in a data field during a write sector command and that the error cannot be corrected.
10	CRC ERR	CRC Error - When set to 1 indicates that a CRC error has been detected in the ID field.
11	ID ERROR	ID Not Found - When set to 1, indicates that a specified ID field (cylinder, head, sector number, or sector size) was not found after all retries were executed.
12		Reserved - Must be set to zero.
13	ABORT	Aborted Command - When set to 1, indicates that a valid command cannot be executed because of drive status information. For example, the drive is not ready, seek operation was not issued, or a read fault exists.
14	TR000 ERR	Track 0 Error - When set to 1 during a restore command, indicates that the track 000 line was not asserted by the drive after all the stepping pulses have been issued.
15	DAM NOT FOUND	Data Address Mark - When set to 1, indicates that the address mark was not found within 16 bytes of the ID field.

6.3.4 SECTOR COUNT REGISTER

The 8-bit Sector Count Register is used by the Read Sector, Write Sector, and Format commands to implement multiple sector handling with one command. A value of zero implies a transfer of 256 sectors regardless of size. During read and write multiple sector commands, the sector count is decremented. The sector count is then incremented after each sector transfer to or from the buffer.

During a format command, the register is loaded with the total number of sectors to be formatted. It is then decremented as each sector is formatted until it reaches zero. During format operations, sector numbers are specified using interleaved tables loaded in the sector buffer.

The command format for reading the register is

DIA ac, 52

The command format for writing to the register is

DOA ac, 52

where ac - the accumulator

6.3.5 SECTOR NUMBER REGISTER

The 8-bit Sector Number Register is loaded with the specified sector number before a read or write command.

The command format for reading the register is

DIA ac, 53

For writing to the register the command format is

DOA ac, 53

where ac - the accumulator

6.3.6 CYLINDER HIGH AND LOW REGISTERS

The Cylinder High and Cylinder Low Registers are 8-bit accumulators. Together they provide a 16-bit register to hold the cylinder number specified for a seek, read, or write command. The two least significant bits (LSB) form the two most significant bits (MSB) of the cylinder number (see Figure 6-3).

The command format for reading the contents of the Cylinder High Register is

DIA ac,55

For writing to the Cylinder High Register, the command format is

DOA ac,55

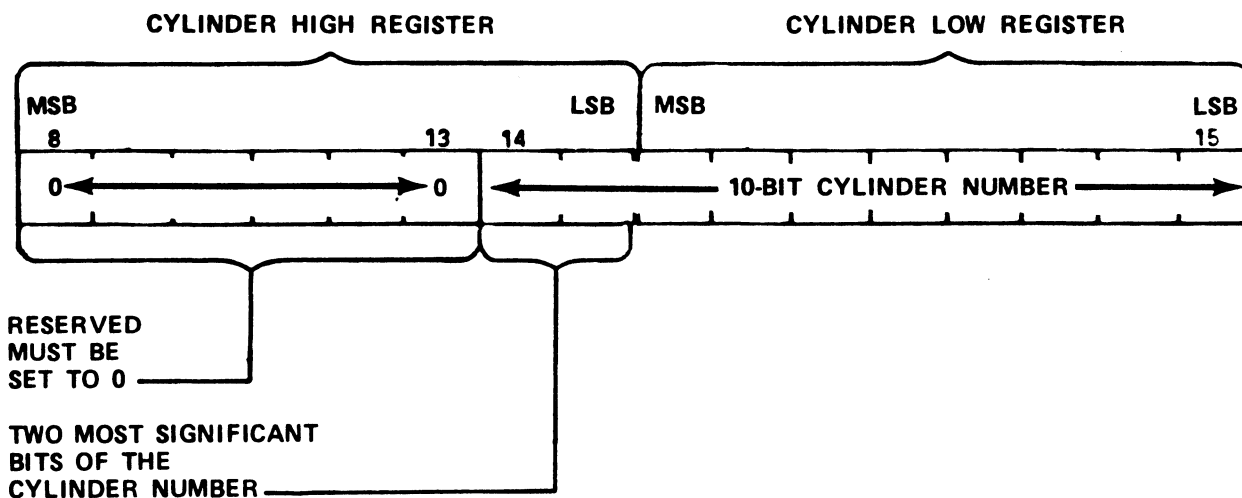
The command format for reading the Cylinder Low Register is

DIA ac,54

For writing to the Cylinder Low Register, the command format is

DOA ac,54

where ac - the accumulator



081-96

Figure 6-3. High-Low Cylinder Registers and Cylinder Number

6.3.7 SELECT/DRIVE/HEAD REGISTER

The 8-bit Select/Drive/Head (SDH) Register contains the ECC/CRC sector size, drive, select, and head select bits. Figure 6-4 shows the bit configuration.

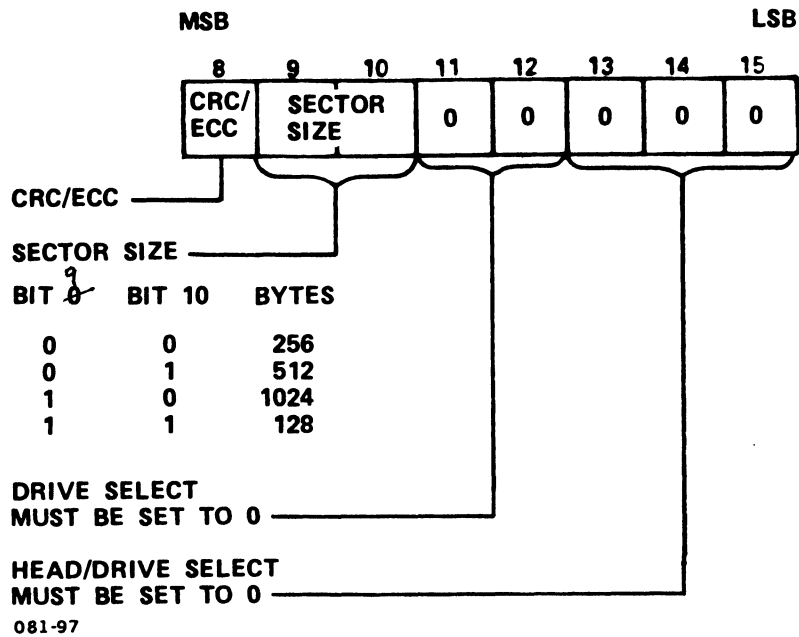


Figure 6-4. SDH Register Bit Configuration

6.3.8 COMMAND REGISTER

The Command and Status Registers are physically the same 8-bit register. As a Command Register, it holds the command currently being executed. A command is input with a DOA instruction. When execution of the command is completed, or if a halt occurs because of an error condition, status information may be retrieved with a DIA instruction (see Section 6.3.9).

Three instructions are available to software:

1. Test
2. Restore
3. Seek

The bit configuration for each command is discussed in Sections 6.3.8.1 through 6.3.8.3. The Command Register should not be loaded when the device is busy.

The command format is

DOA ac,57

where ac - the accumulator

6.3.8.1 Test Command Bit Configuration

The Test command is used to isolate faults in the WD 1002-5 board logic by running internal diagnostics. Faults are reported in the Error Register (see Section 6.3.3). The Command Register bit configuration for a Test command is shown in Figure 6-5.

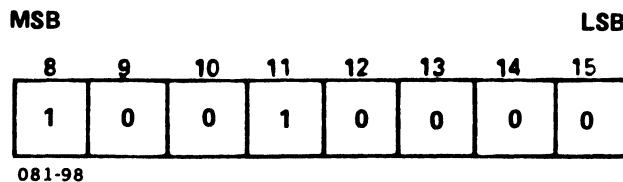


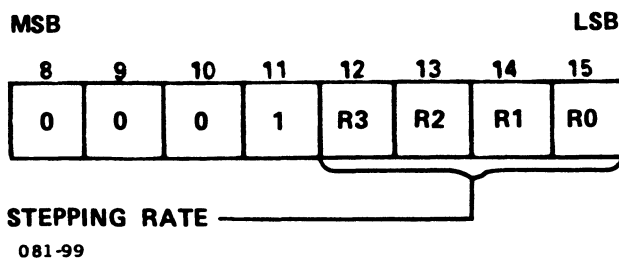
Figure 6-5. Test Command Bit Configuration

6.3.8.2 Restore Command Bit Configuration

The Restore command is used to calibrate the position of the read/write head by stepping the head outward until the TR000 line is true.

When a Restore command is received, the Busy bit in the Status Register is set. The Cylinder High and Cylinder Low Registers are cleared. For ST506 operation, the actual stepping rate is determined by the Seek Complete period.

The Restore command bit configuration is shown in Figure 6-6.



BITS (12-15) R3-R0	ST506
0000	APPROX. 35 MICROSECONDS
0001	0.5 MILLISECONDS
0010	1.0
0011	1.5
0100	2.0
0101	2.5
0110	3.0
0111	3.5
1000	4.0
1001	4.5
1010	5.0
1011	5.5
1100	6.0
1101	6.5
1110	7.0
1111	7.5 MILLISECONDS

↓

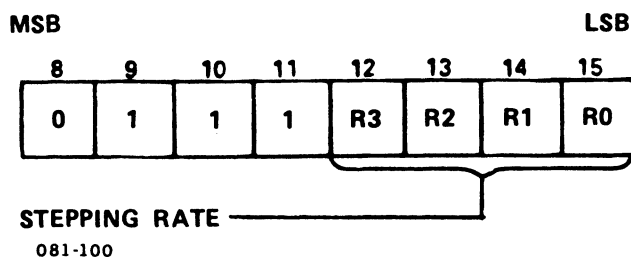
081-99/081-100

Figure 6-6. Restore Command Bit Configuration

6.3.8.3 Seek Command Bit Configuration

The Seek command positions the read/write head at a specified cylinder. It is primarily used to start two or more concurrent seeks on drives that support buffered stepping. The Seek Complete Line is not sampled after a Seek command to make multiple seek operations possible.

The bit configuration for a Seek command is shown in Figure 6-7.



BITS (12-15) R3-R0	ST506
0000	APPROX. 35 MICROSECONDS
0001	0.5 MILLISECONDS
0010	1.0
0011	1.5
0100	2.0
0101	2.5
0110	3.0
0111	3.5
1000	4.0
1001	4.5
1010	5.0
1011	5.5
1100	6.0
1101	6.5
1110	7.0
1111	7.5 MILLISECONDS

081-99/081-100

Figure 6-7. Seek Command Bit Configuration

6.3.9 STATUS REGISTER

The 8-bit Status Register contains information on the status of the drives after a command has been executed. It uses the same address as the Command Register (see Section 6.3.8). If the Busy bit is set, no other bits in the register are valid. The INTRQ line is reset when the Status Register is accessed. Status information is retrieved with a DIA instruction in the following format:

DIA ac,57

where ac - the accumulator

The bit configuration for the Status Register is shown in Figure 6-8. Table 6-6 defines the function of each bit.

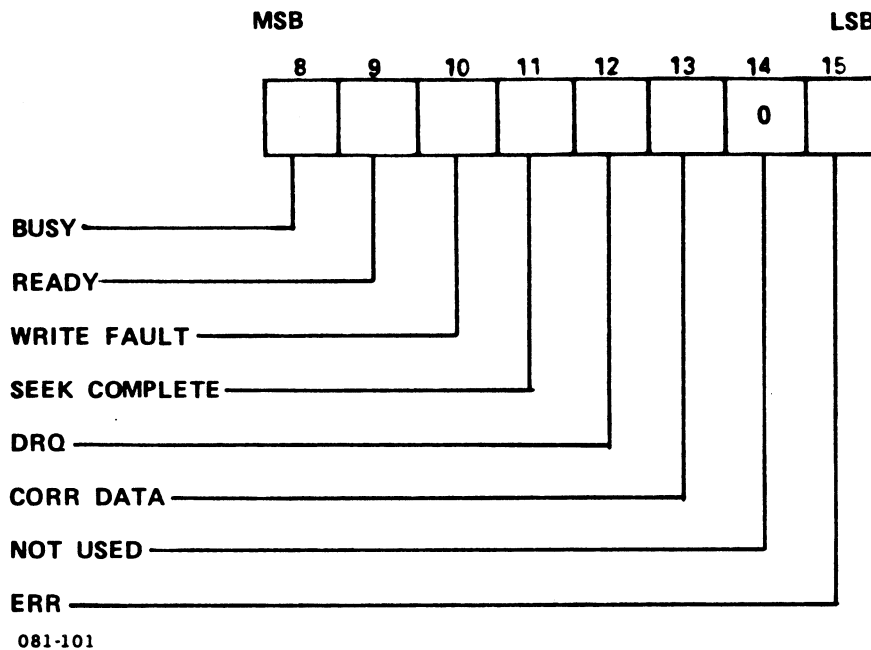


Figure 6-8. Status Register Bit Configuration

TABLE 6-6. STATUS REGISTER BIT FUNCTIONS

Bits	Name	Function
8	BUSY	Busy - When set to 1, a command is being executed. No other bits in this or any other register are valid when this bit is set.
9	READY	Drive Ready - When set to 1, indicates that the specified drive is ready. No command is executed unless this bit is set.
12	WRITE FAULT	Write Fault or Write Protect - When set to 1, no command is executed. If a write-protected disc is sensed, the write command is aborted.
11	SEEK COMPLETE	Seek Complete - When set to 1, indicates that a seek operation is completed. For floppy operations, the bit is set when the SDH Register is loaded.
12	DRQ	Data Request - When set to 1, indicates that the Sector Buffer is ready to accept data or contains data to be read. This bit is reset when the Sector Buffer has been fully read from or written to.
13	CORR DATA	Corrected Data - When set to 1, indicates that an error correction has been successfully completed on one or more data fields when in multiple mode or on the data field just read from the ST506 disc. If an uncorrectable error occurred, the command is aborted and the appropriate bit is set in the Error Register.
14		Not used.
15	ERR	Error Condition - When set to 1, indicates that one or more bits are set in the Error Register. Reset when a new command is issued.

6.4 INPUT/OUTPUT CONTROL BLOCK (IOCB)

The input/output control block (IOCB) contains commands to be performed via direct memory access (DMA). The IOCB for the ST506/Floppy Disc Controller consists of five words located anywhere in main memory. It is used to indicate the operational mode. The five words are set up by the software before an operation begins (see also Section 6.3). The format of the ST506 Disc IOCB is shown in Figure 6-9.

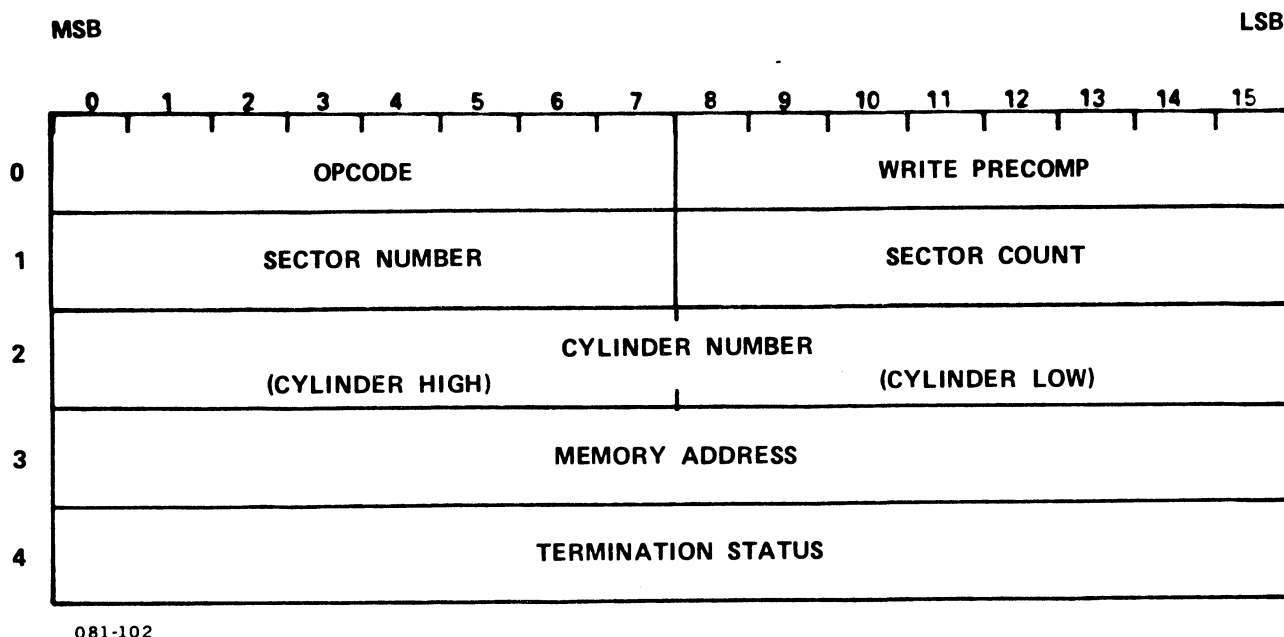


Figure 6-9. ST506 Disc IOCB Format

6.4.1 OPCODE/WRITE PRECOMP WORD (WORD 0)

Word zero of the IOCB is divided into two bytes. The first byte (bits 0-7) contains the opcode for five possible data transfer commands:

- Read Sector
- Readlong
- Write Sector
- Writelong
- Format Track

These commands use an implicit stepping rate for the read/write head positioning which is determined by the stepping rate specified for the last Restore or Seek command executed.

The second byte (bits 8-15) contains the contents of the Write Precomp Register which defines the cylinder address where the lower write current is to be turned on (see Section 6.3.2).

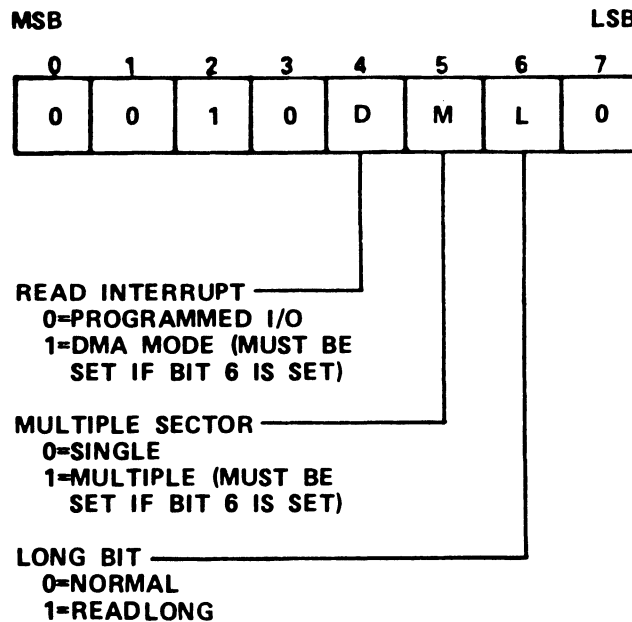
6.4.1.1 Read Sector Command

The Read Sector command is used to read a sector of data from disc. The Status and Error Registers are updated for every block of data transferred.

Provided the ECC is enabled, ECC bytes are recomputed. After the buffer is full, the recorded ECC bytes are compared to the check bytes and syndrome bytes are generated. If the syndrome is nonzero, errors have occurred.

Error correction is invoked if two consecutive syndromes match; otherwise up to eight retries are attempted. If the data is correctable, the correction is made and the data is read into memory. If eight retries are unsuccessful, an error status is sent to the CPU, Multiple Sector Read commands are modified to single sector commands and are issued a corresponding number of times.

The Read Sector command bit configuration is shown in Figure 6-10.



081-103

Figure 6-10. Read Sector/Readlong Command Bit Configuration

6.4.1.2 Readlong Command

The Readlong command is similar to the Read Sector command except that the generation of the syndrome bytes is inhibited. The four recorded check bytes are copied from the disc and read into memory. The Readlong command can only be used in DMA mode (bit 4=1) which transfers an entire buffer of data at a time. Refer to Figure 6-10 for the appropriate bit settings.

6.4.1.3 Write Sector Command

The Write Sector command is used to write a sector of data from memory to disc. Upon receipt of the command, the DRQ bit is set in the Status Register until the entire sector length has been written into the buffer.

The bit configuration for a Write Sector command is shown in Figure 6-11.

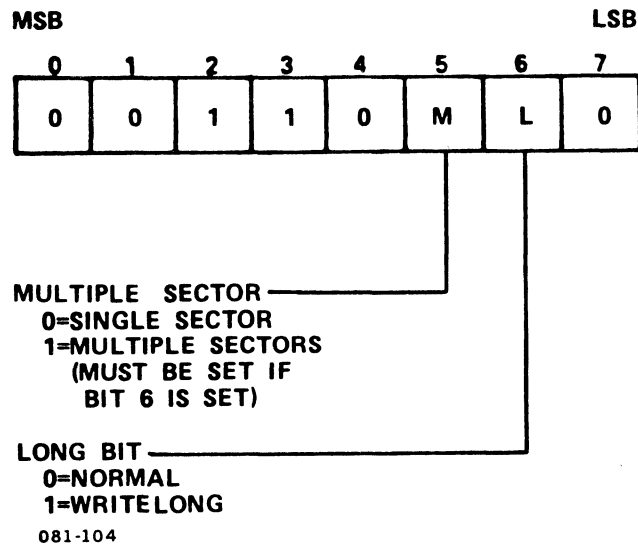


Figure 6-11. Write Sector/Writelong Command Bit Configuration

6.4.1.4 Writelong Command

The Writelong command functions like the Write Sector command except that ECC operations are inhibited. The four check bytes are passed unaltered to the buffer and are subsequently written to disc at the end of the data.

Refer to Figure 6-11 for the appropriate Writelong command bit configuration.

6.4.1.5 Format Track Command

The Format Track command is used for initializing the ID and data fields on a particular disc. The DRQ bit in the Status Register is set until the interleave table is written to the buffer. The number of bytes transferred to the buffer must correspond to the current sector size.

When the buffer has been completely filled, the specified number of sectors are written to disc and the DRQ bit is reset. The data field is written with 00 for hard disc or E5 (Hex) for floppies. ECC or CRC bytes are automatically computed and written to disc.

Once the index is found, a number of ID and data fields are written to disc. As each sector is written, the Sector Count Register is decremented. Consequently, it must be updated before each format operation.

The Format Track command bit configuration is shown in Figure 6-12.

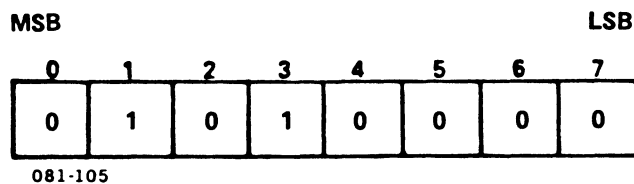


Figure 6-12. Format Track Command Bit Configuration

6.4.4 MEMORY ADDRESS WORD (WORD 3)

Word 3 is used to specify the word address of the memory buffer. The format of the memory address word is shown in Figure 6-15.

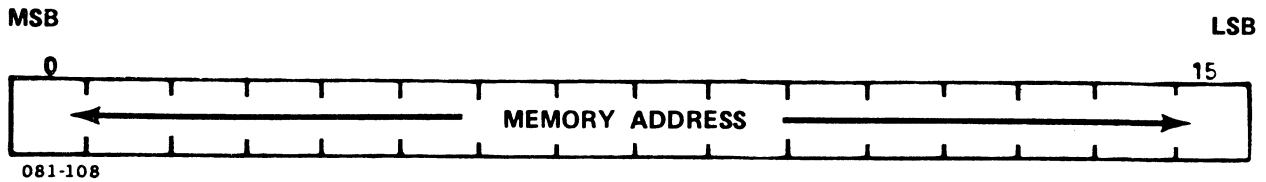


Figure 6-15. Memory Address Bit Configuration (Word 3 of IOCB)

6.4.5 TERMINATION STATUS WORD (WORD 4)

Upon completion of an operation, the controller writes the termination status into word 4 of the IOCB. The result of the operation is represented in the lower byte of word 4 and the most significant bit is set when an operation has been completed (see Figure 6-16).

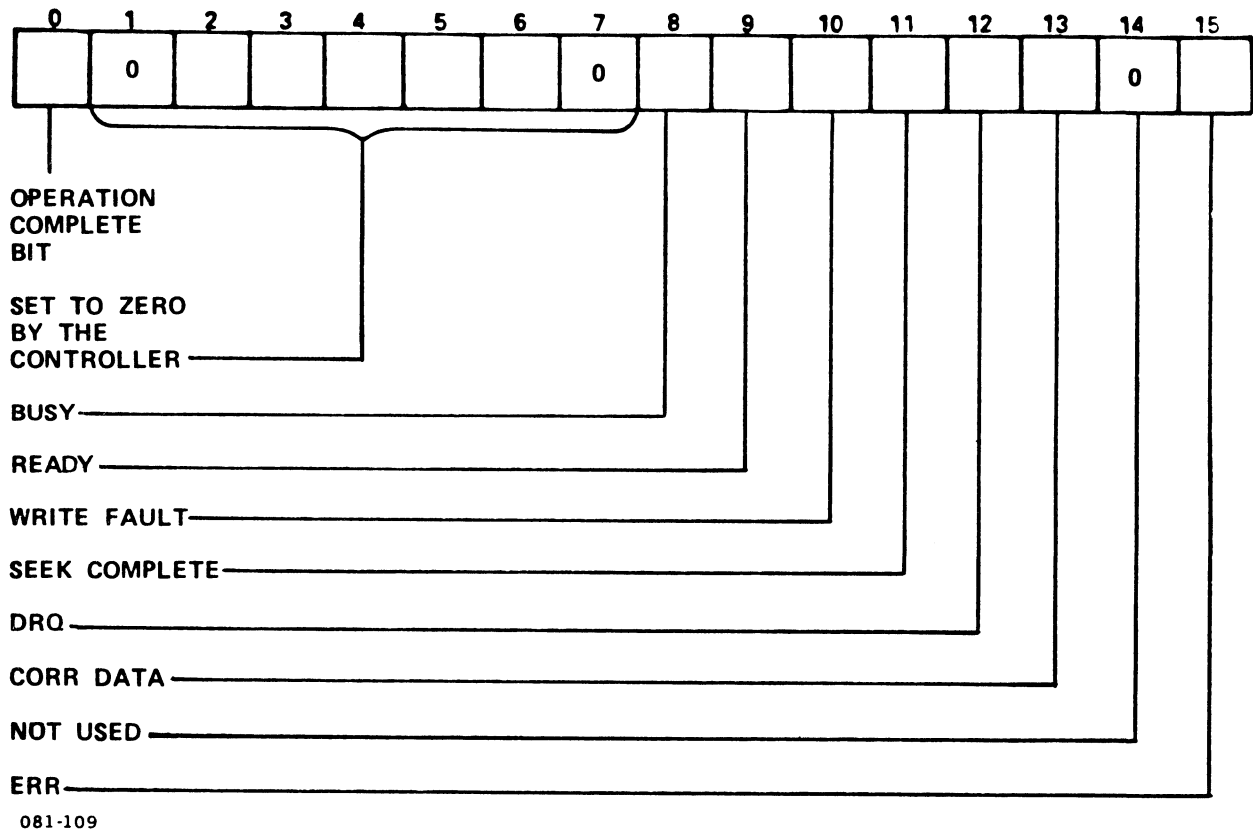


Figure 6-16. Termination Status Bit Configuration (Word 4 of IOCB)

6.4.2 SECTOR NUMBER AND COUNT WORD (WORD 1)

Word 1 of the IOCB contains the sector number and sector count. The first byte (bits 0-7) is loaded with the number of the sector being transferred. The second byte (bits 8-15) contains the sector count for a format operation or zeros for single block transfers.

Figure 6-13 shows the bit configuration for word 1 of the IOCB.

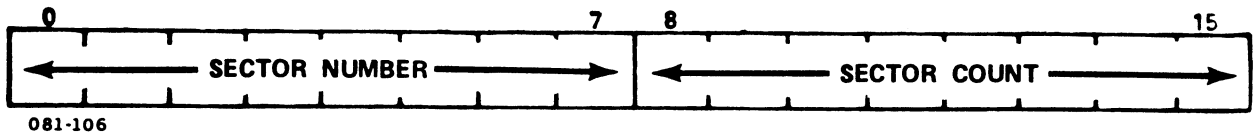


Figure 6-13. Sector Number and Count Bit Configuration (Word 1 of IOCB)

6.4.3 CYLINDER NUMBER WORD (WORD 2)

Word 2 of the IOCB contains the cylinder number as represented by the Cylinder High and Cylinder Low Registers.

Figure 6-14 shows the bit configuration for word 2 of the IOCB.

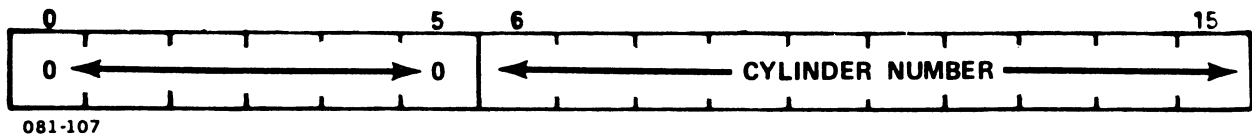


Figure 6-14. Cylinder Number Bit Configuration (Word 2 of IOCB)

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:

<u>Figure</u>	<u>Operation</u>
B-1	Command Transfer Sequence
B-2	Read Status Sequence
B-3	Write Sequence
B-4	Read Sequence
B-5	Write Transfer Sequence
B-6	Read Transfer Sequence
B-7	Off-line Sequence

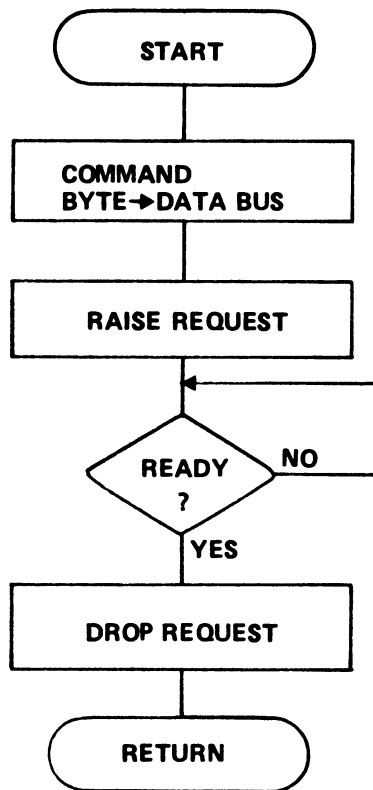


Figure B-1. Command Transfer Sequence Flowchart

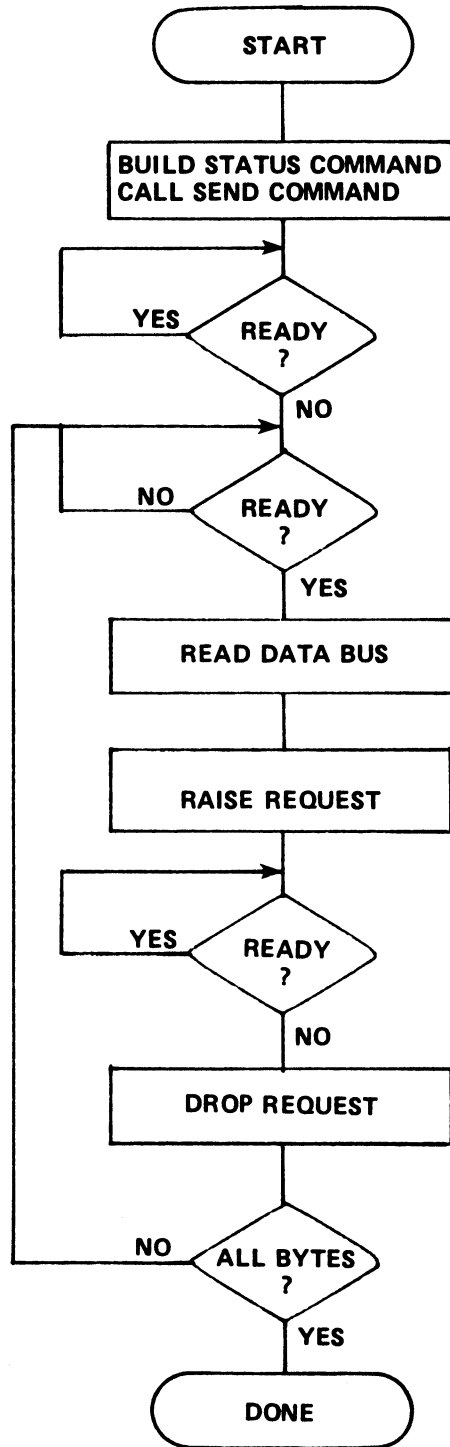


Figure B-2. Read Status Sequence Flowchart

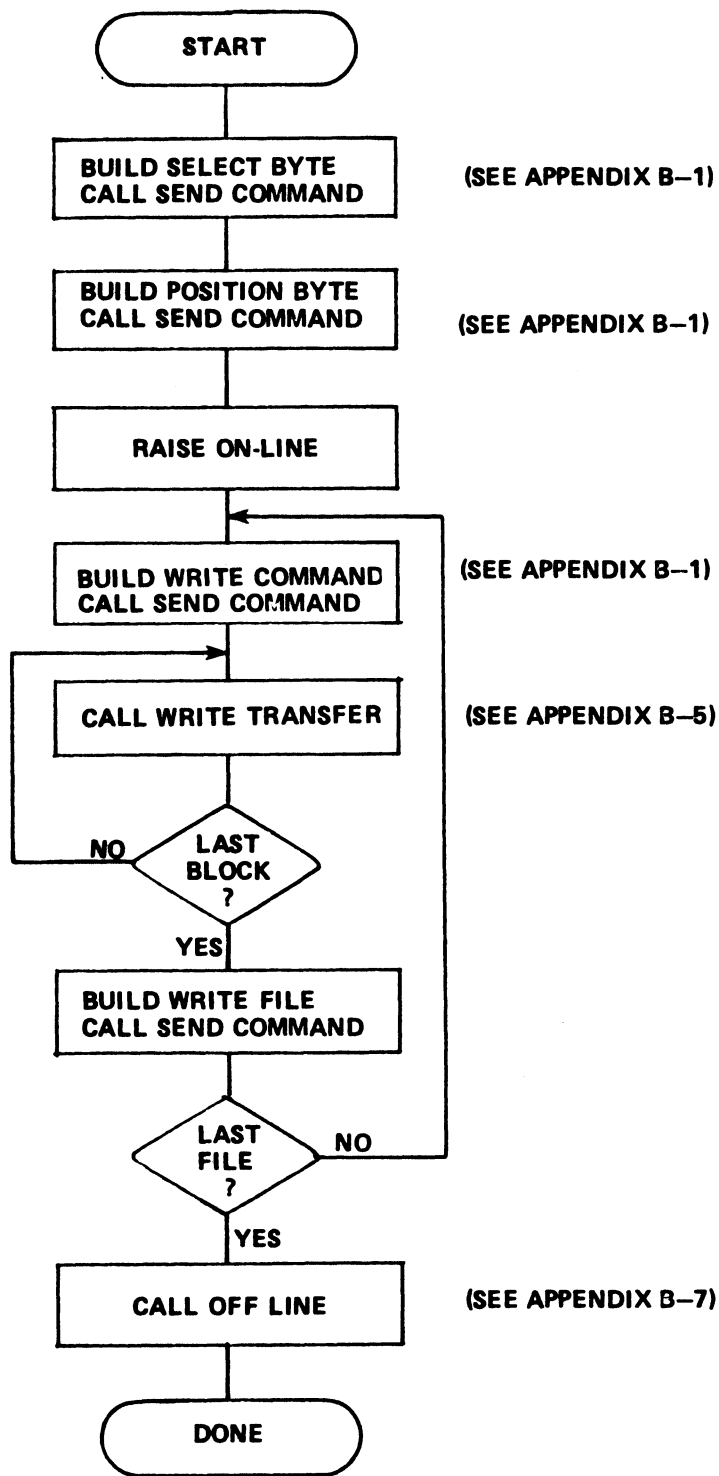


Figure B-3. Write Sequence Flowchart

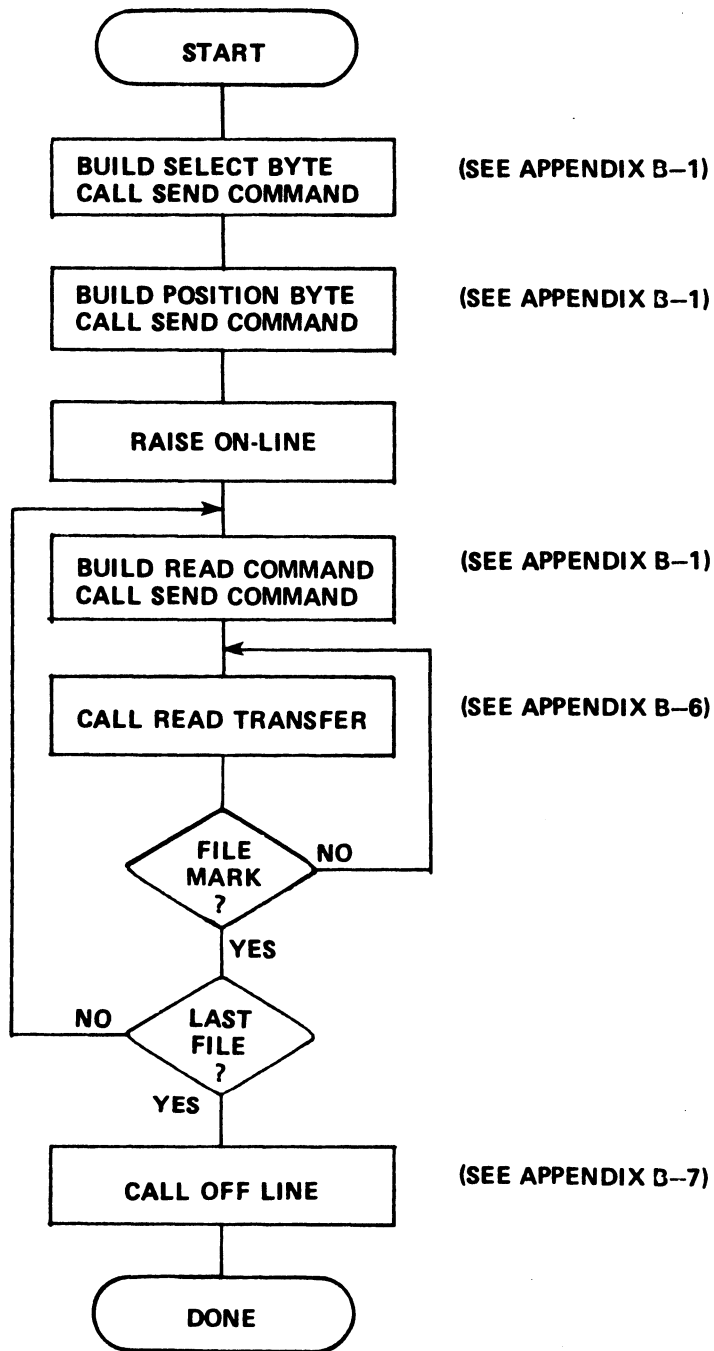


Figure B-4. Read Sequence Flowchart

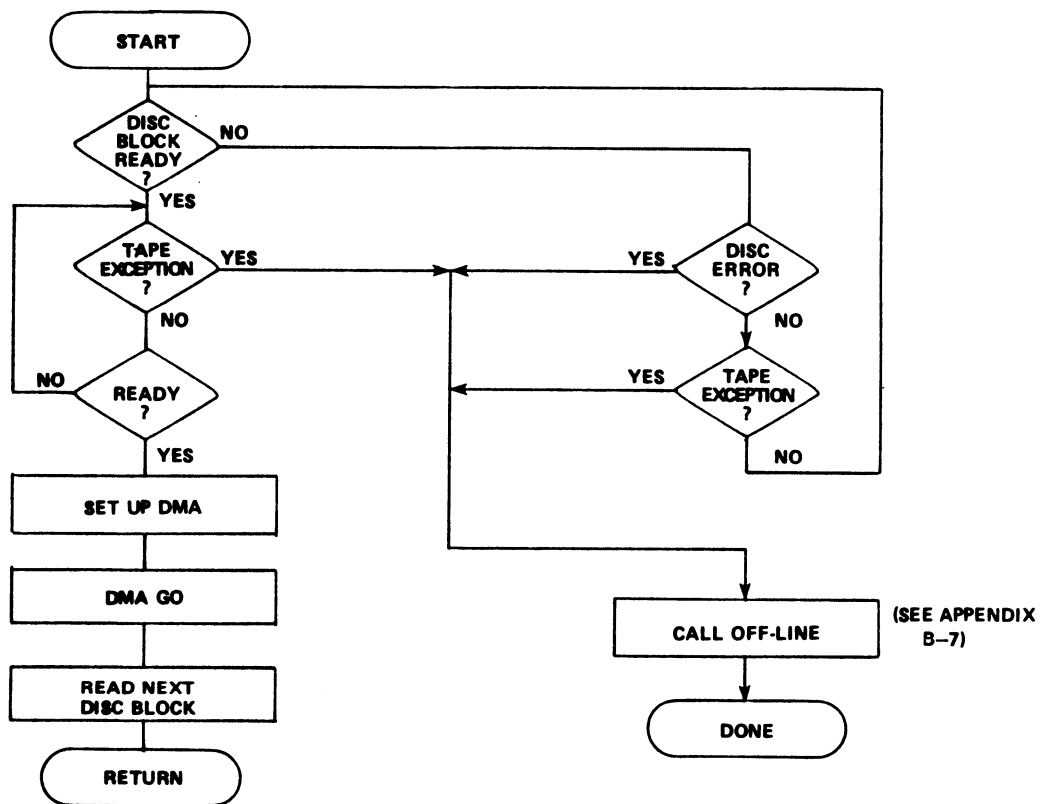


Figure B-5. Write Transfer Sequence Flowchart

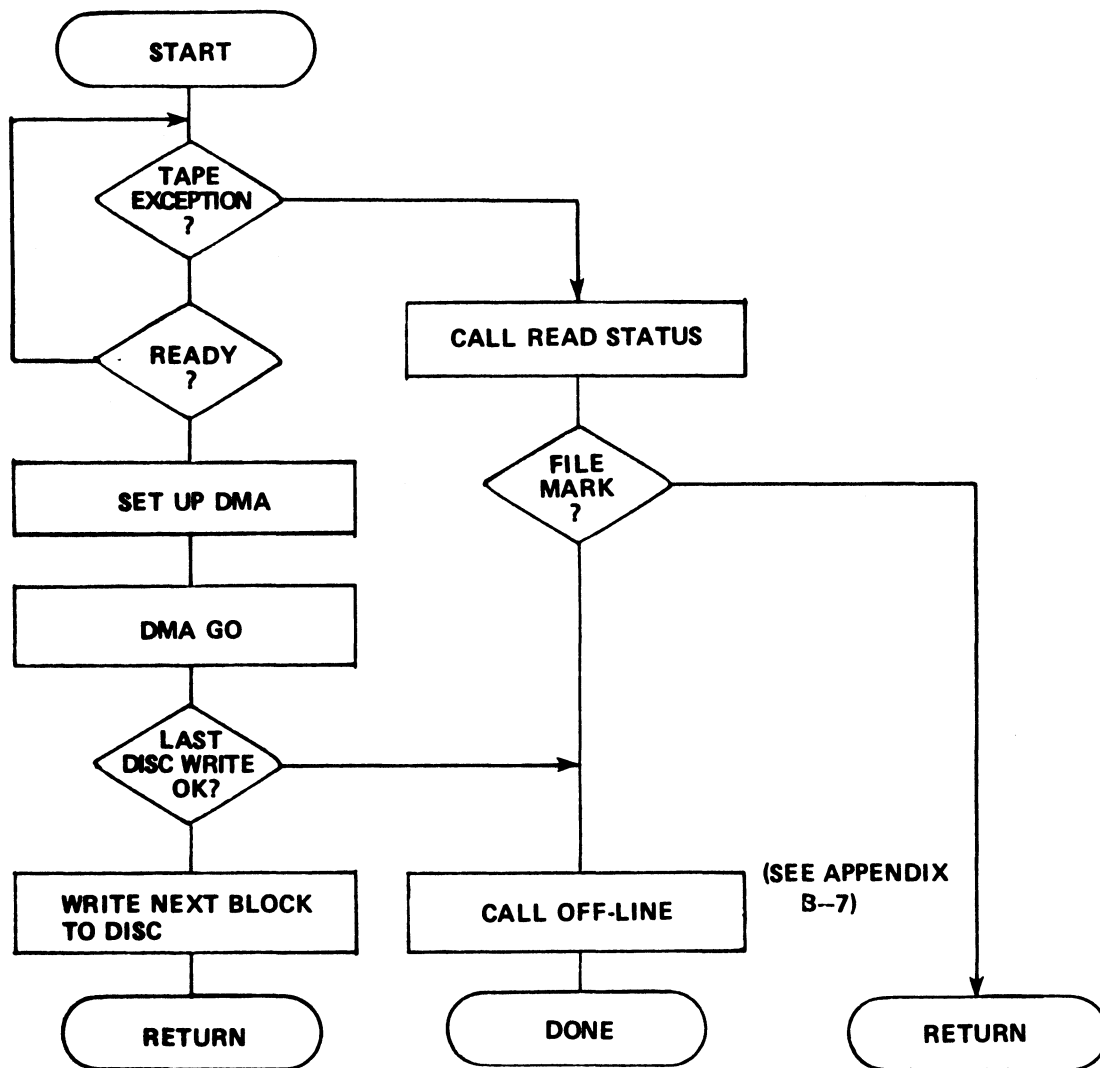


Figure B-6. Read Transfer Sequence Flowchart

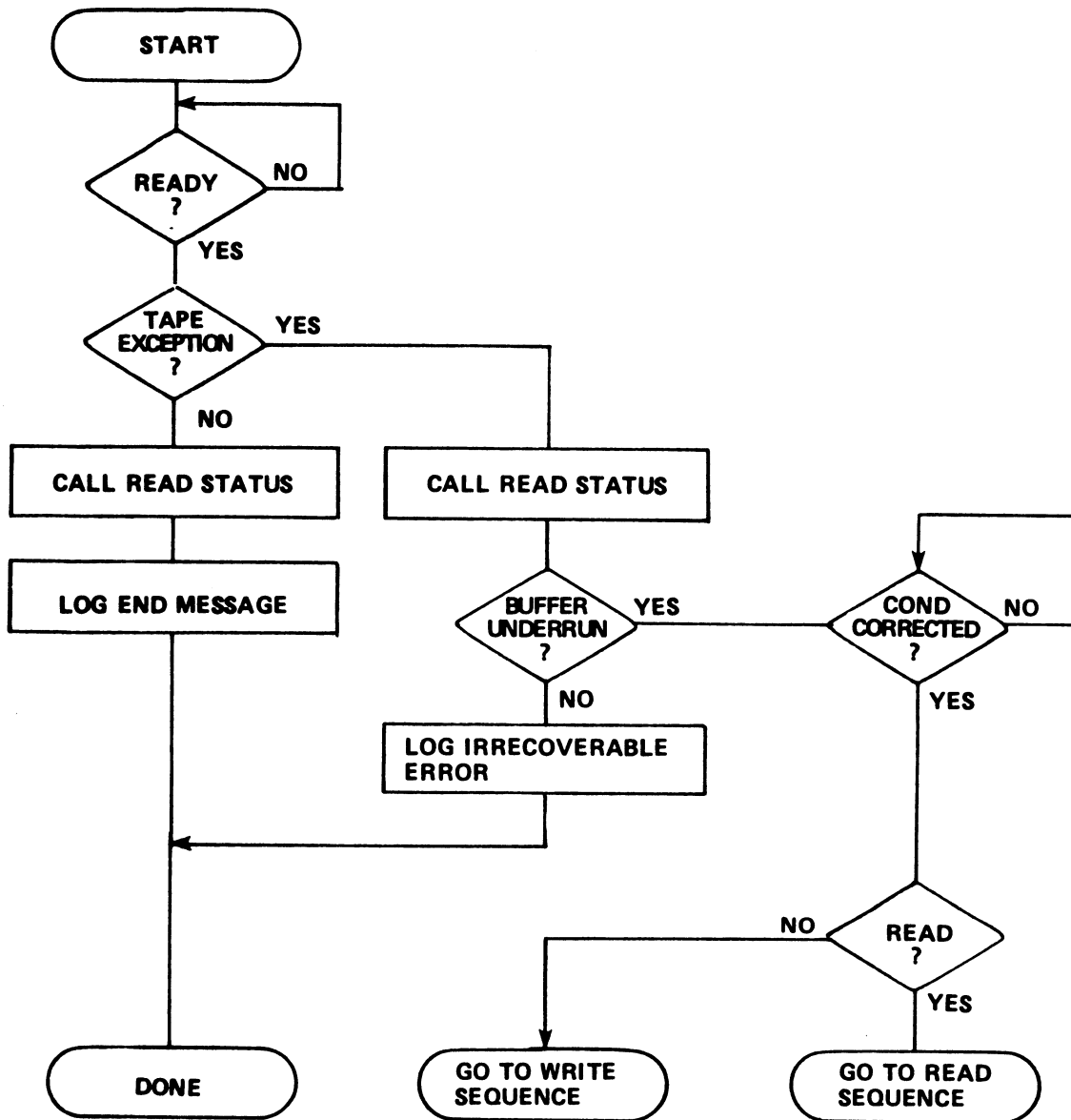


Figure B-7. Off-line Sequence Flowchart

Appendix C

FLOPPY DISC HANDLING FLOWCHARTS

This appendix contains programming flowcharts for typical floppy disc handling operations. The flowcharts cover the floppy disc command sequences discussed in Section 5. They also include internal sequences not covered in the text. The text divides the commands into two types: head movement and data transfer commands. The flowcharts, which provide additional detail, divide the commands into three types. The three types cover the following commands:

Type I = Head Movement commands

- Restore
- Seek
- Step
- Step-in
- Step-out

Type II = Data Transfer commands

- Read Sector
- Write Sector

Type III = Data Transfer/Format commands

- Read Track/Address
- Write Track (Format)

Operations covered in the flowcharts are:

<u>Operation</u>	<u>Figure</u>
Head Movement (Type I)	C-1
Read/Write Sector (Type II)	C-2
Write Track (Type III)	C-3
Read Track/Read Address (Type III)	C-4

Abbreviations used in the flowcharts are defined below.

CR	Command Register
CRC	Cyclic Redundancy Check
DAL	Data Access Line
DAM	Data Address Mark
DDEN	Double Density
DR	Data Register
DRQ	Data Request
DSR	Data Shift Register
HLD	Head Load
HLT	Head-Load Timing
IDAM	ID Address Mark
INTRQ	Interrupt Request
SR	Sector Register
SSO	Side Select Output
STR	Status Register
TG	Track Greater Than
TR	Track Register
WG	Write Gate

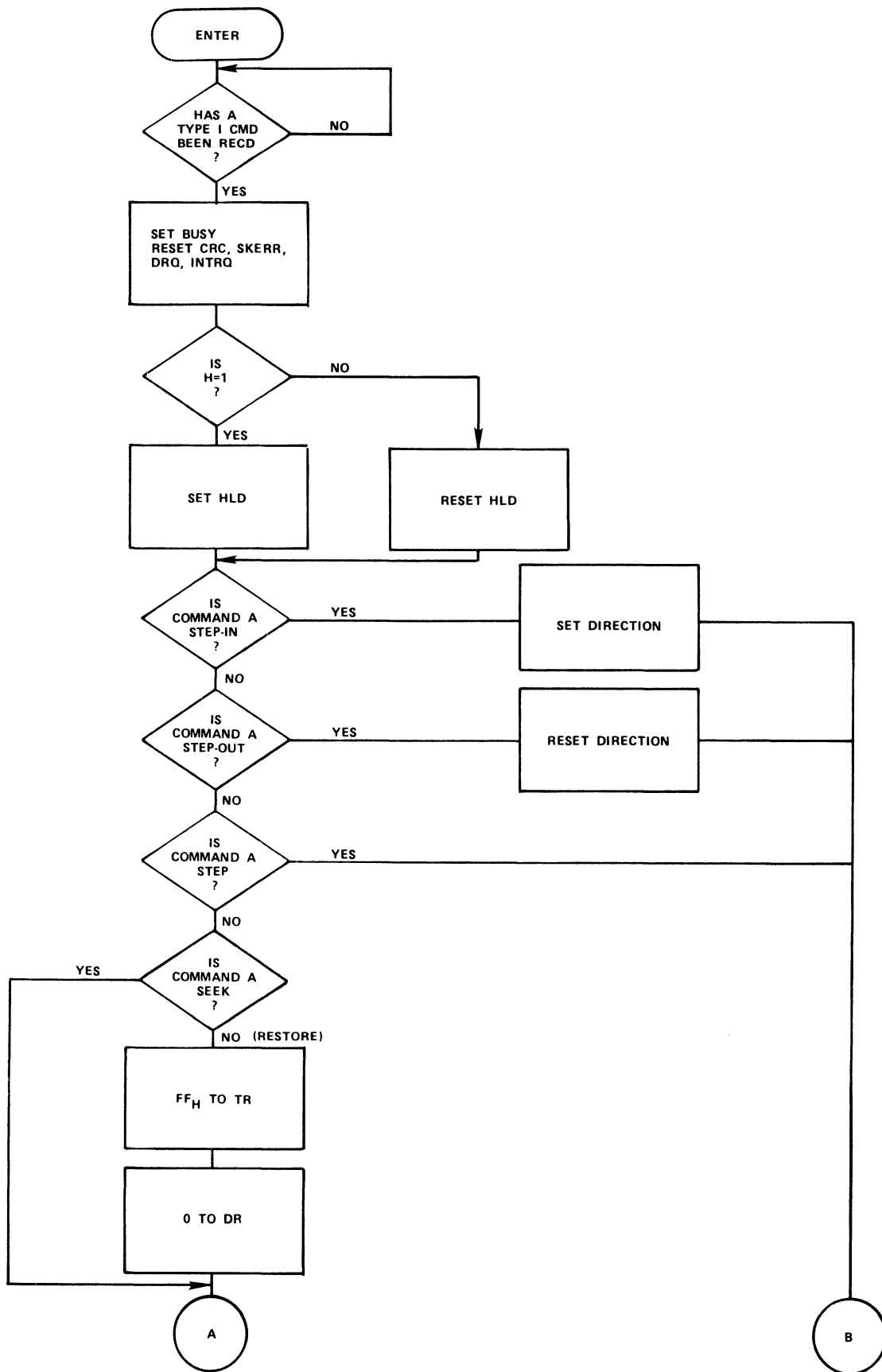


Figure C-1. Head Movement (Type I) Flowchart (1 of 3)

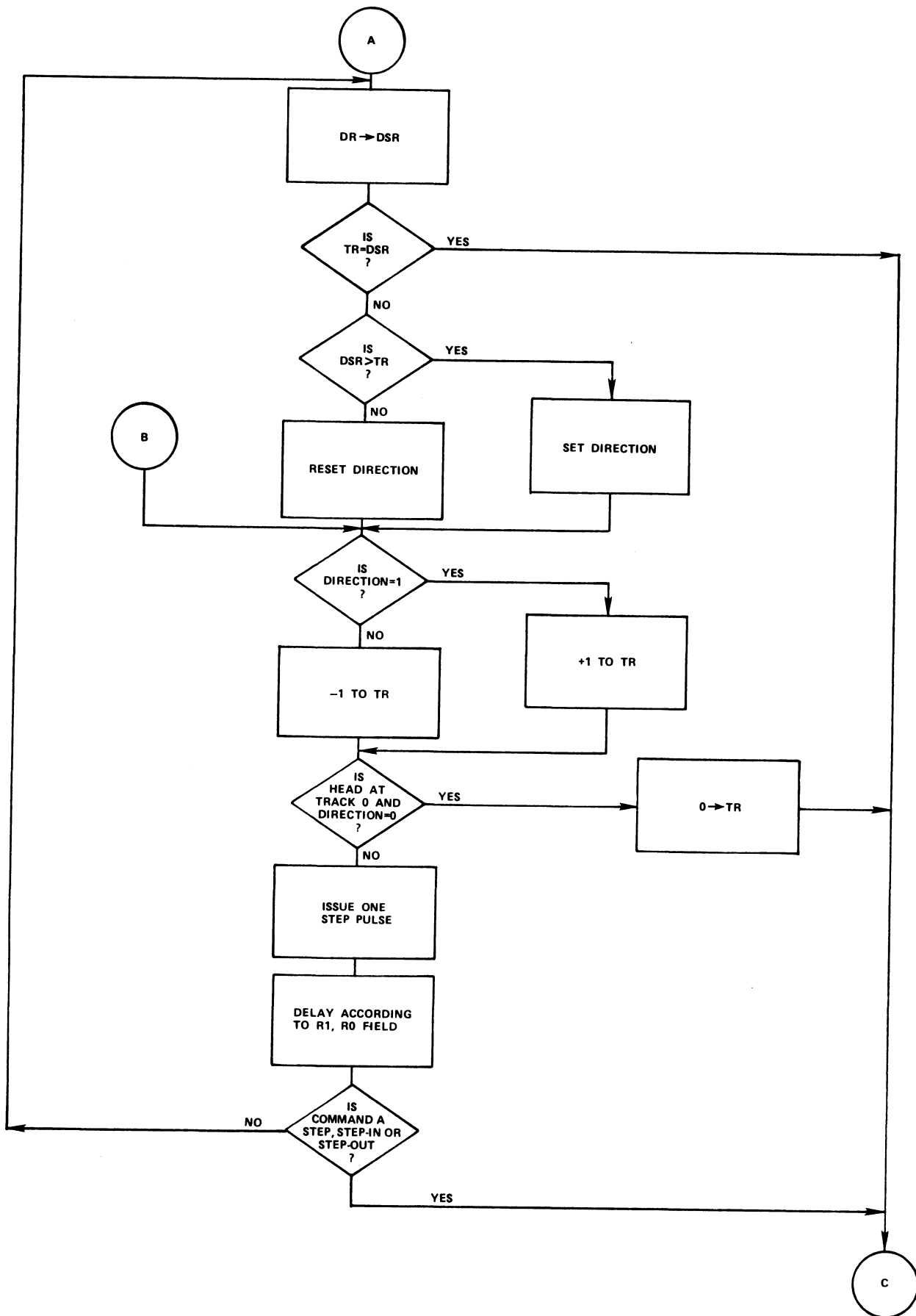


Figure C-1. Head Movement (Type I) Flowchart (2 of 3)

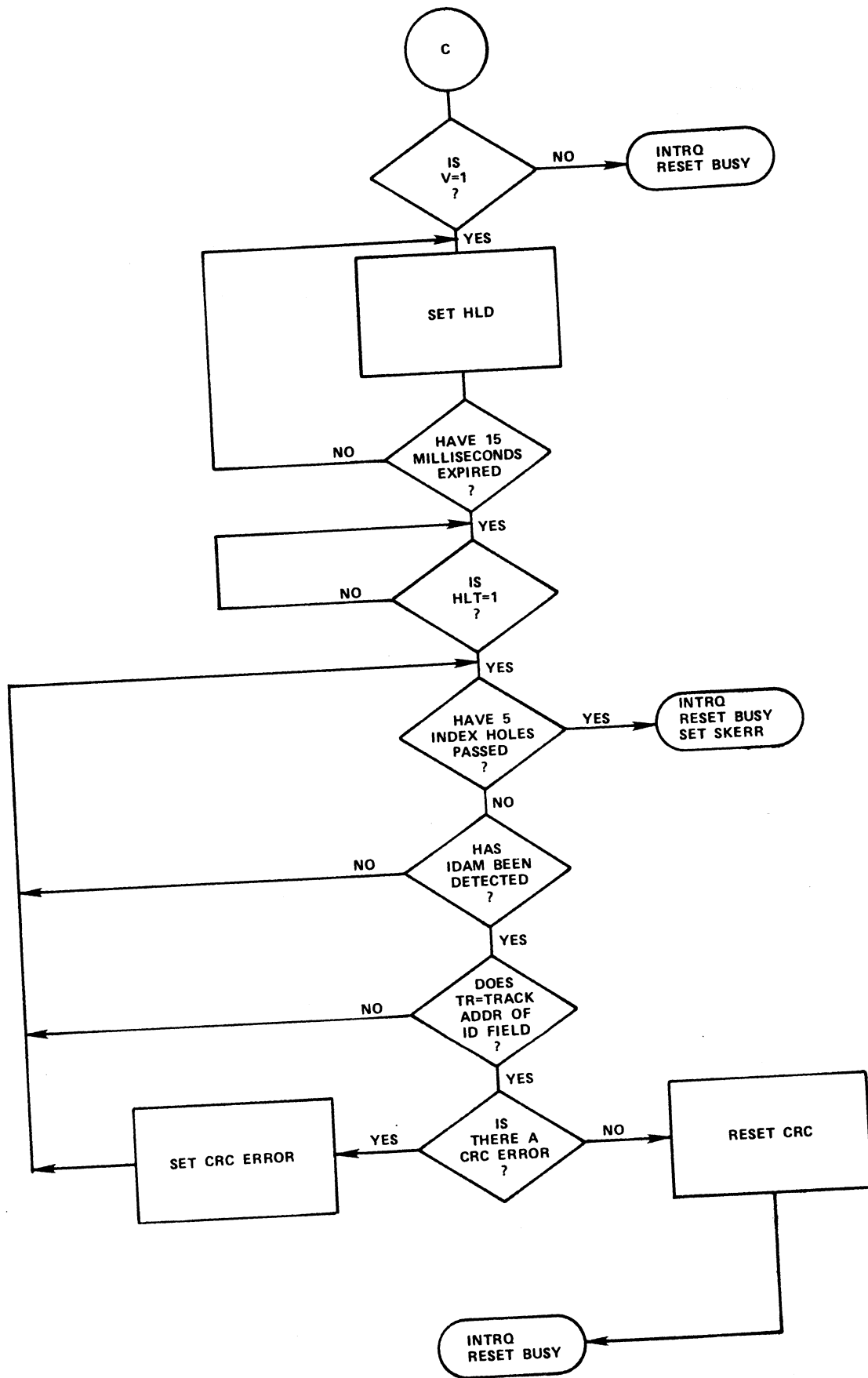


Figure C-1. Head Movement (Type I) Flowchart (3 of 3)

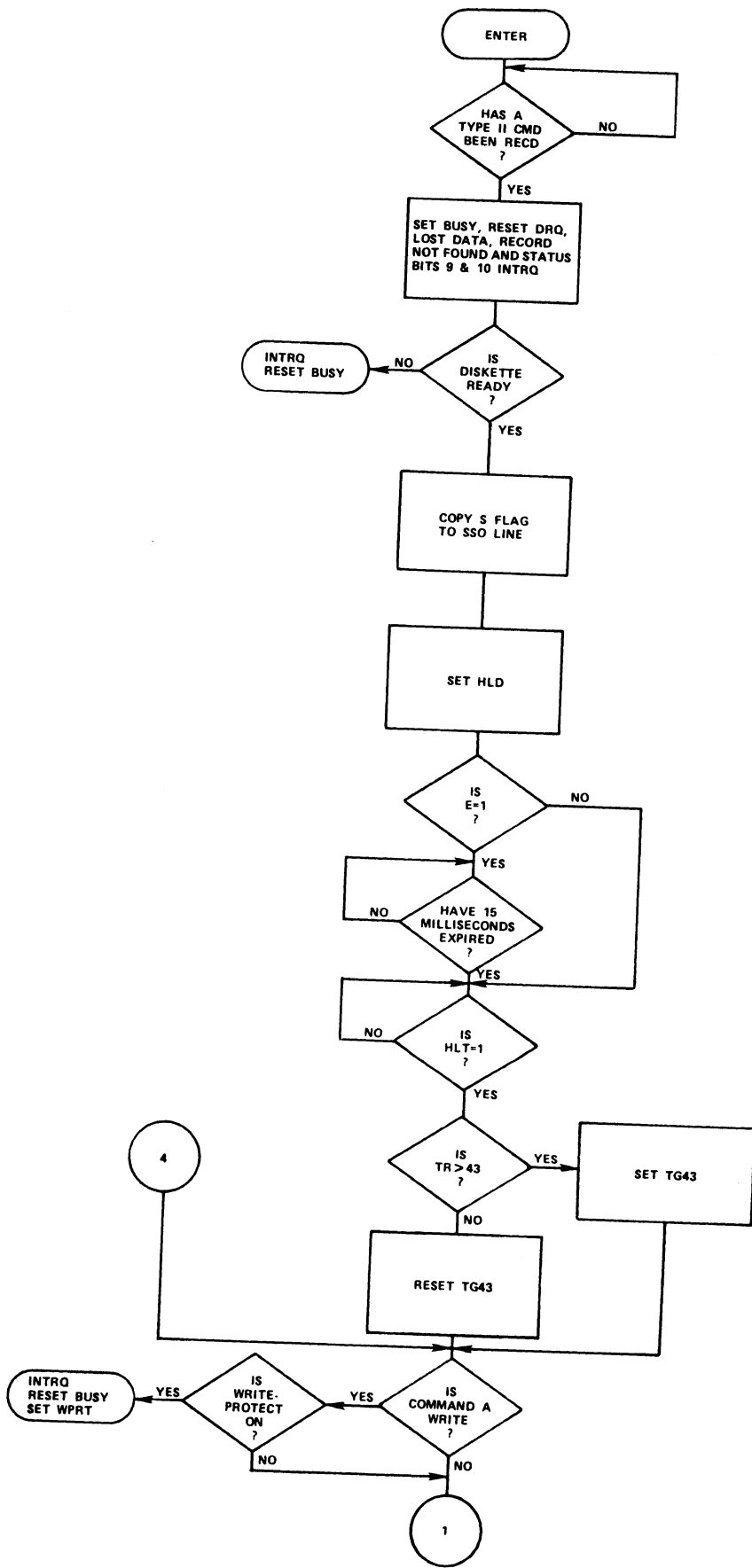


Figure C-2. Read/Write Sector (Type II) Flowchart (1 of 3)

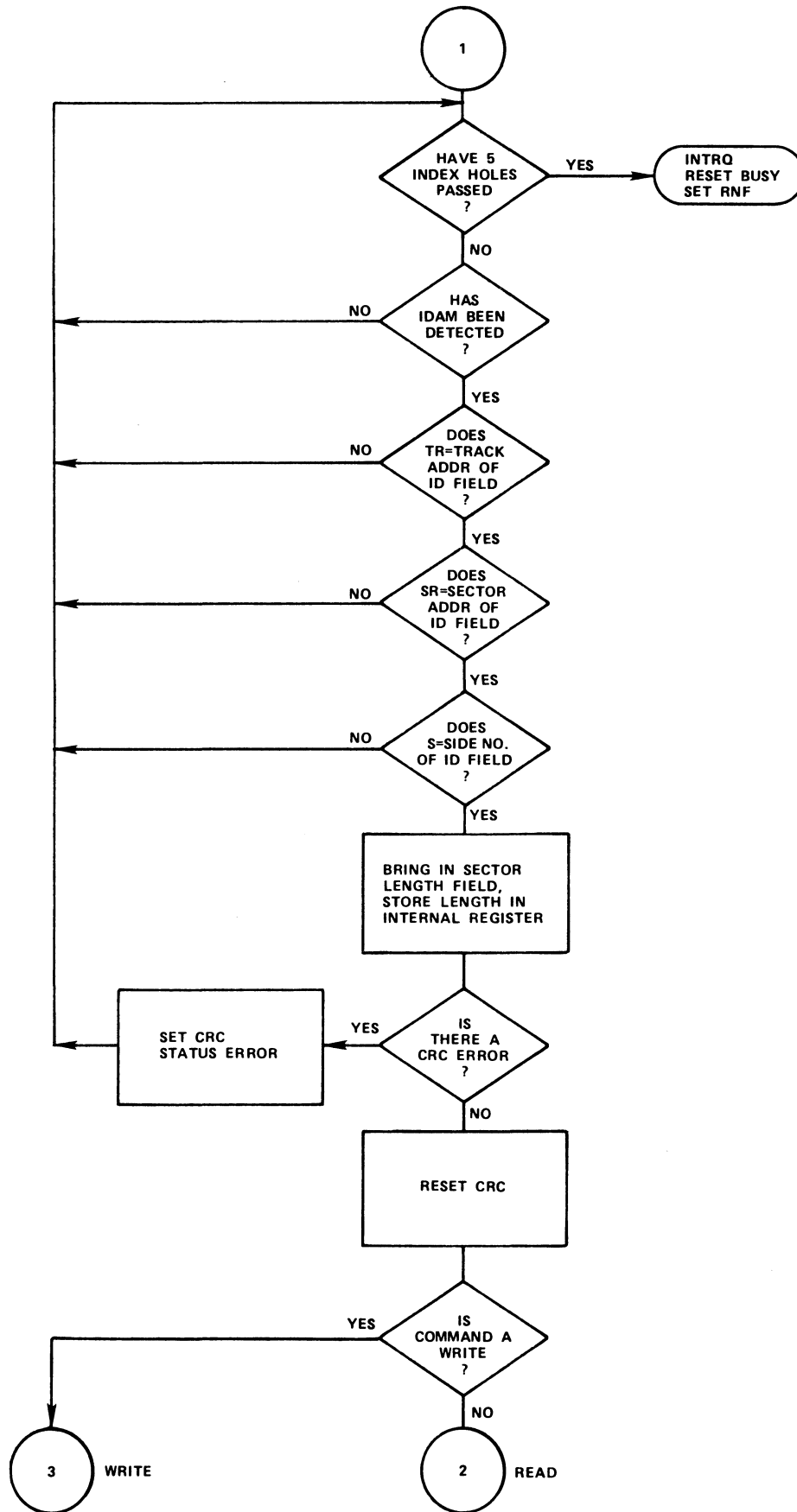


Figure C-2. Read/Write Sector (Type II) Flowchart (2 of 3)

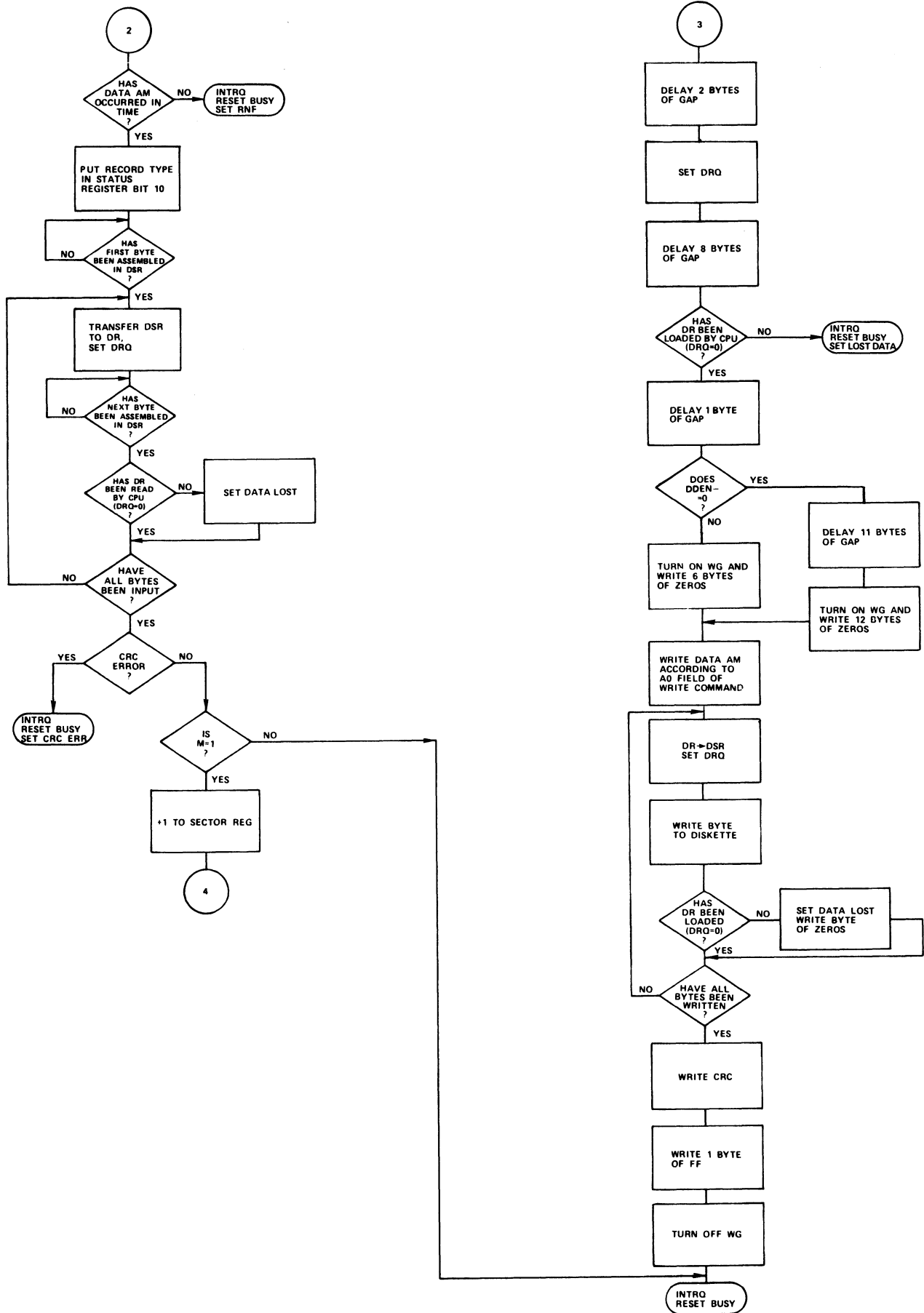


Figure C-2. Read/Write Sector (Type II) Flowchart (3 of 3)

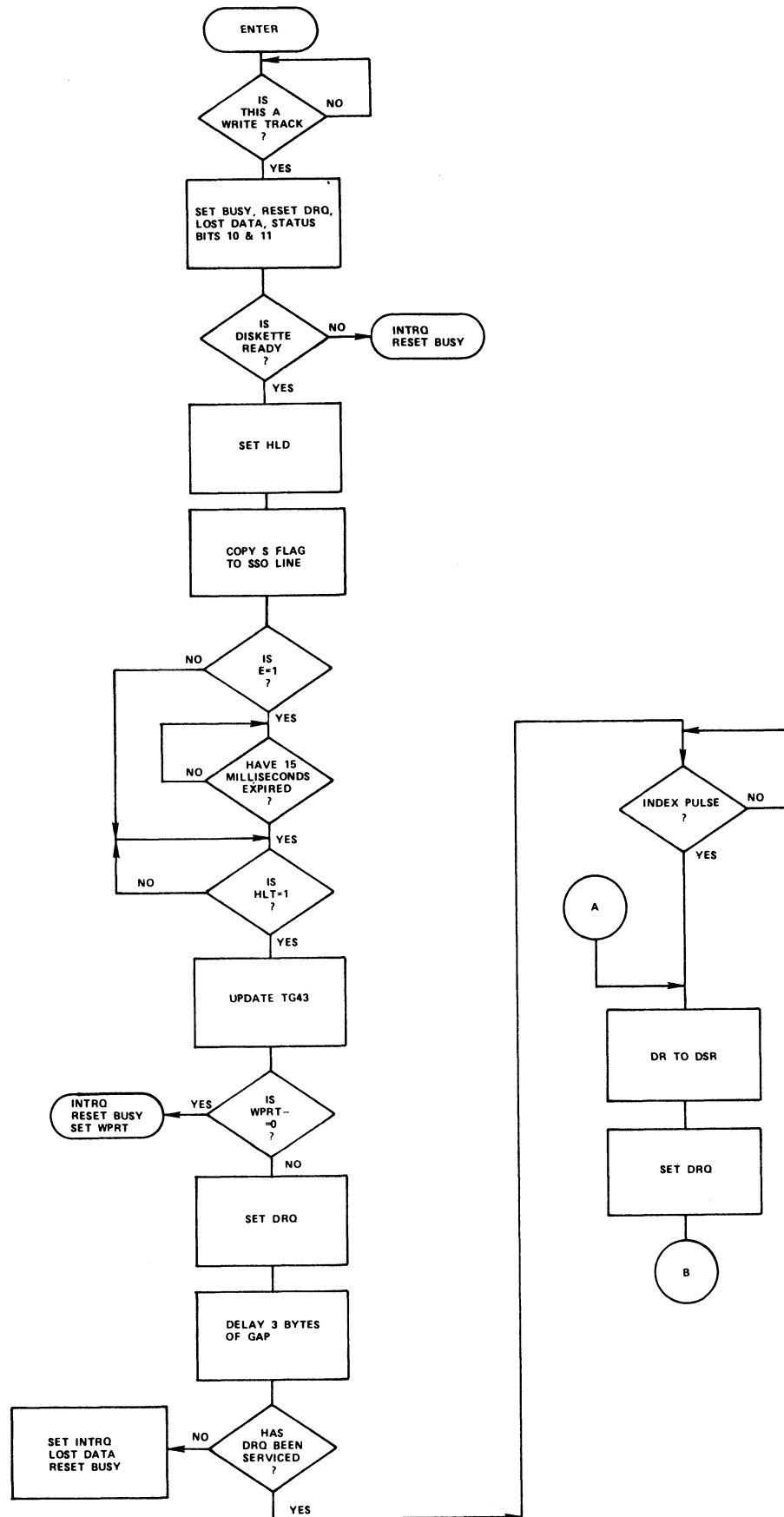


Figure C-3. Write Track (Type III) Flowchart (1 of 2)

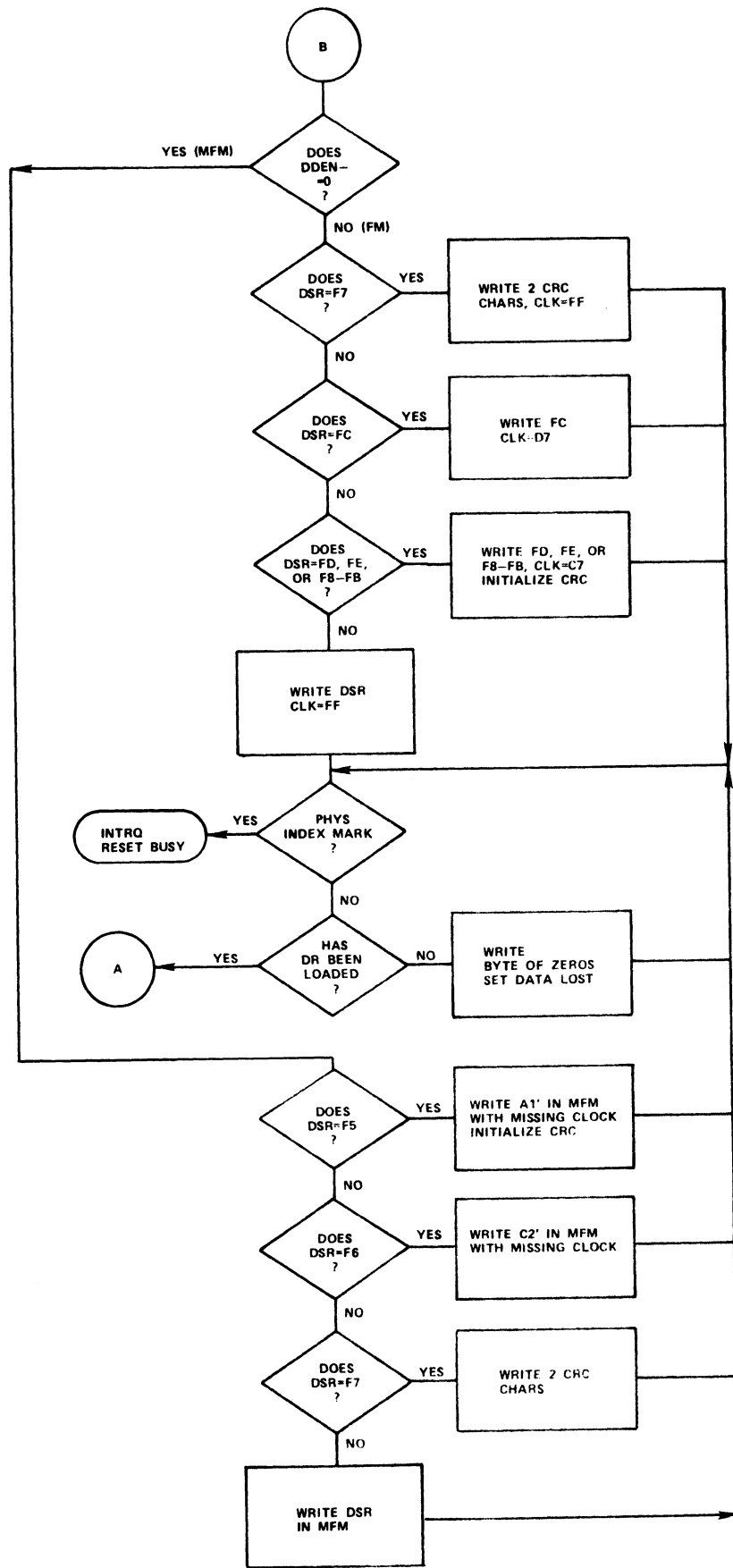


Figure C-3. Write Track (Type III) Flowchart (2 of 2)

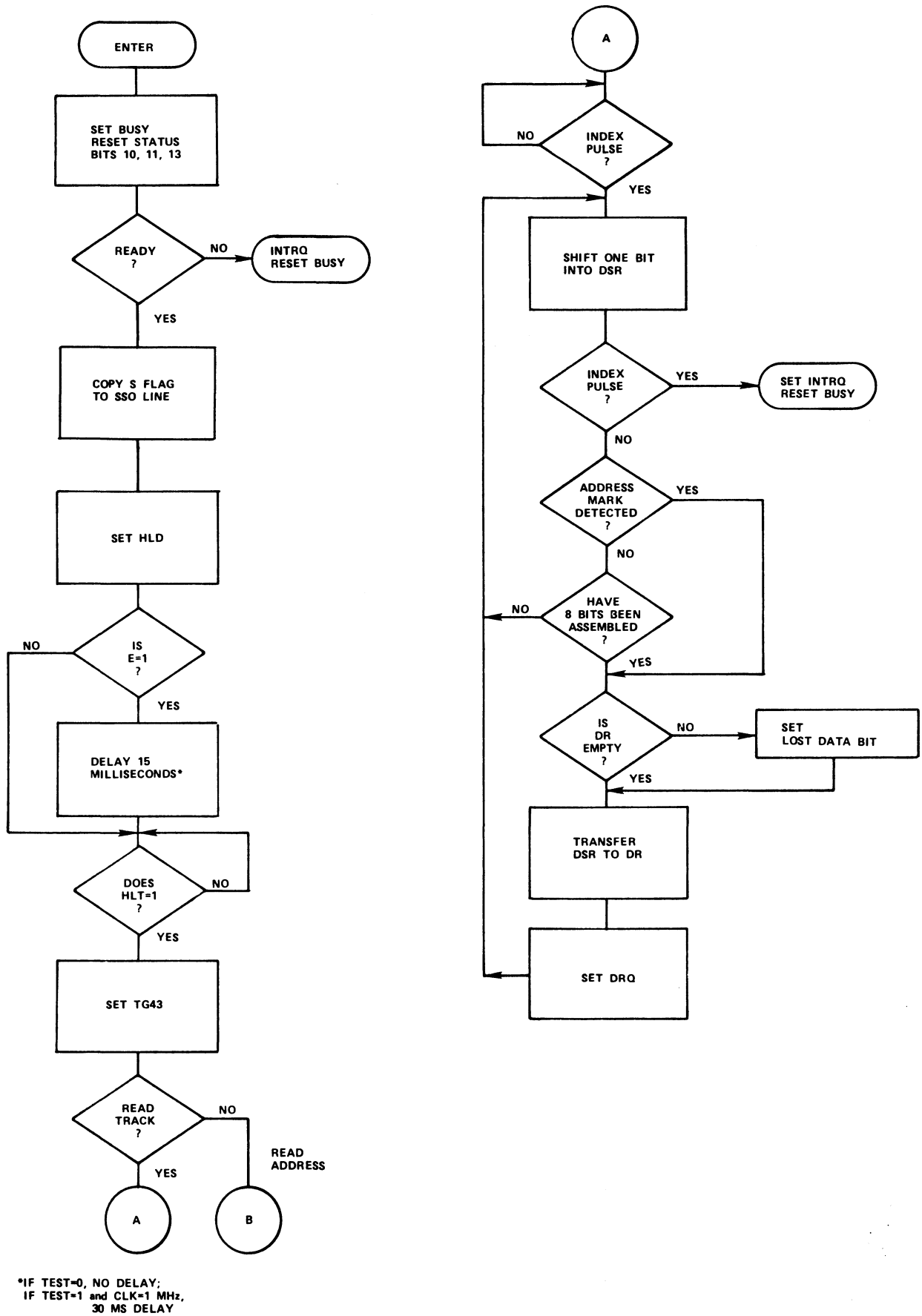


Figure C-4. Read Track/Read Address (Type III) Flowchart (1 of 2)

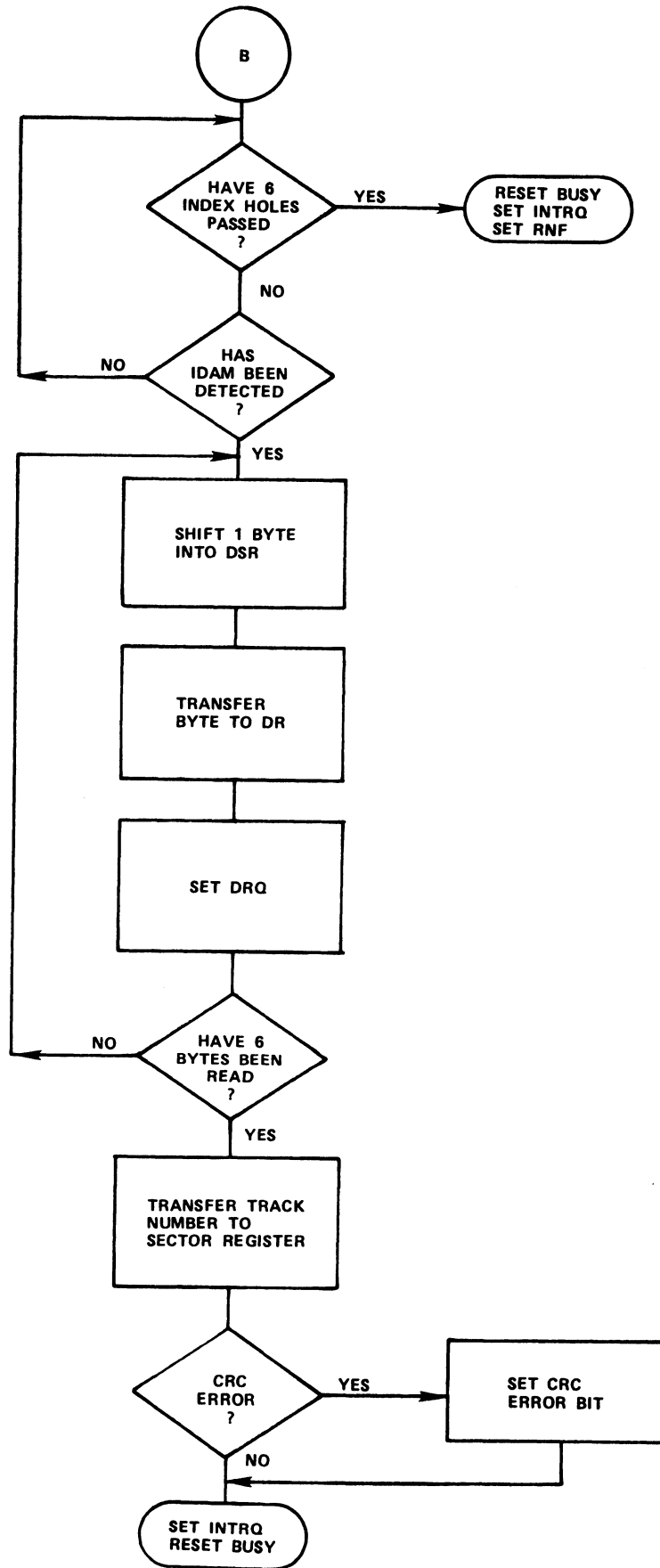


Figure C-4. Read Track/Read Address (Type III) Flowchart (2 of 2)

COMMENT SHEET

MANUAL TITLE POINT 4 MARK 2/3 Peripherals Interface Manual

PUBLICATION NO. HM-081-0027 REVISION D

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.



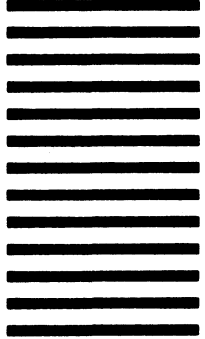
NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1458 TUSTIN, CA

POSTAGE WILL BE PAID BY ADDRESSEE

POINT 4 Data Corporation
PUBLICATIONS DEPARTMENT
15442 Del Amo Avenue
Tustin, CA 92680



CUT ON THIS LINE





15442 Del Amo Avenue
Tustin, CA 92680
(714) 259-0777