PROGRAMMER'S REFERENCE SERIES: MODELS 6236/6237 AND 6239/6240 DISK SUBSYSTEMS

Warning:

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instruction manual, may cause interference to radio communications. It has been tested and found to comply with the limits for Class A computing devices pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.



NOTICE

Data General Corporation (DGC) has prepared this document for use by DGC personnel, customers and prospective customers. The information contained herein shall not be reproduced in whole or in part without DGC's written approval.

DGC reserves the right to make changes in specifications and other information contained in this document without prior notice, and the reader should in all cases consult DGC to determine whether any such changes have been made.

THE TERMS AND CONDITIONS GOVERNING THE SALE OF DGC HARDWARE PRODUCTS AND THE LICENSING OF DGC SOFTWARE CONSIST SOLELY OF THOSE SET FORTH IN THE WRITTEN CONTRACTS BETWEEN DGC AND ITS CUSTOMERS. NO REPRESENTATION OR OTHER AFFIRMATION OF FACT CONTAINED IN THIS DOCUMENT INCLUDING BUT NOT LIMITED TO STATEMENTS REGARDING CAPACITY, RESPONSE-TIME PERFORMANCE, SUITABILITY FOR USE OR PERFORMANCE OF PRODUCTS DESCRIBED HEREIN SHALL BE DEEMED TO BE A WARRANTY BY DGC FOR ANY PURPOSE, OR GIVE RISE TO ANY LIABILITY OF DGC WHATSOEVER.

IN NO EVENT SHALL DGC BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATED TO THIS DOCUMENT OR THE INFORMATION CONTAINED IN IT, EVEN IF DGC HAS BEEN ADVISED, KNEW OR SHOULD HAVE KNOWN OF THE POSSIBILITY OF SUCH DAMAGES.

DASHER, microNOVA, TRENDVIEW, microECLIPSE, and ECLIPSE are U.S. registered trademarks of Data General Corporation.

SECOND EDITION

(Second Printing June 1984)

FIRST EDITION

(First Printing July 1983)

Ordering No. 014-701001 Copyright® Data General Corporation, 1984 All Rights Reserved Printed in the United States of America Rev. 01, June 1984

PREFACE

This is the programmer's reference manual for the Model 6236/6237 and the Model 6239/6240 Data General disk subsystems. It describes the subsystems from an assembly language programmer's viewpoint and presents the instruction set and typical programming techniques.

If you need to know how to operate the controls of the disk drives, read the Operator's Guide manuals:

- Part number 014-701001 for the Model 6236/6237 drive.
- Part number 014-000798 for the Model 6239/6240 drive.

For information about programming the ECLIPSE® computer, read the User's Manual: Programmer's Reference for ECLIPSE® Line Computers, part number 015-000024.

Contacting Data General

- To order any Data General manual, notify your sales representative and supply the manual title and order number.
- If you have software problems, please notify your local Data General systems engineer.
- If you have hardware problems, please notify the Field Engineering Dispatch Center.

CONTENTS

	age		Page
CHAPTER 1		eving Interrupt Information	
INTRODUCTION		ynchronous Interrupts	
	Syr	nchronous Interrupts	3-17
DISK SUBSYSTEM HARDWARE	-1 DIA	A	3-18
Disk Drives	-2 DIE	3	3-19
Controller	-3 DIC	Z	3-20
DISK SUBSYSTEM SOFTWARE	-4 PIO CO	MMANDS	3-21
Commands and Data Structures	-6 Stand	lard Return Format	3-22
Commands	-6 Contr	rol and Initialization Commands	3-22
Control Blocks	-6 Get	t Controller Information (033)	3-23
Information Blocks		t Interface Information (031)	3-24
Programming the Disk Subsystem	_	t Mapping Information (027)	3-24
	Get	t Unit Information (035)	3-25
CHAPTER 2	Set	Controller Information (032)	
HOST/CONTROLLER INTERFACE		Interface Information (030)	
11051/CONTROLLER INTERF/IGE		Mapping Information (026)	
CONTROLLER REGISTERS	Cat	Unit Information (034)	
	-1 TT	it Status (131)	
Command Registers		ided Status Commands	
Status Registers	-4	t Extended Status 0 (040)	
BUSY AND DONE FLAGS	٠,	t Extended Status 1 (041)	
Busy Flag	-0	t Extended Status 2 (042)	
Done Flag	-0	t Extended Status 3 (043)	
Checking Busy and Done Settings		Control Commands	
PULSES	-4	ncel List (123)	
CONTROL BLOCKS	-3	t List Status (133)	
Specifying Controller Operations	-J Dog	start (116)	
CB Format	-0	rt List (100)	
INFORMATION BLOCKS	-12	rt List-High Priority (103)	
User Options	-1 <i>2</i>	espass (132)	
Interface Information Block	-14	Independent Commands	
Controller Information Block	-14 D!-	gnostic Mode (024)	
Retry Strategy	-13	gen (025)	
Unit Information Block	-10 Dog	set (777)	
Extended Status	-40		
ADDRESSING AND	D	While Reset Commands	
MAPPING CONVENTIONS	-20 D	gin (002)	
Addressing Conventions	-Z1	ogram Load (000)	
Mapping Conventions	-44	MMANDS	
Address Transfers	-43	ral Commands	
INTERRUPTS	-47	Operation (000)	
		calibrate Disk (400)	
CHAPTER 3		Commands	
COMMANDS AND INSTRUCTIONS			
		ad/Verify (201)	
INSTRUCTIONS	- I	nd/Verify Single Word (205)	
Instruction Syntax and Internal	Kea	ad Raw Data (210)	
Representation	-1	ad Headers (220)	
Sending a Command	-5 Kea	ad with Modified Bit Map (242)	
DOA	.9 Write	Commands	
DOB	-10 Wr	ite (100)	
	10 Wr	ite/Verify (101)	3-47

Page			Page
3-48	COM	MAND SET	A-1
3-48	Ins	tructions	A-1
3-49	PIC	O Commands	A-4
	CB	Commands	A-4
	Co	mmand Registers	A-5
	Sta	tus Registers: Asynchronous	
4-1		-	A-7
4-1	Sta	tus Registers: Synchronous	
4-2			
4-2			
4-4			
4-4		-	
4-5			
4-6			
4-6			
4-7		-	
4-7			A-16
4-7			
4-9			
4-11			
4-13			
4-13			
4-15	MAP	PING OPTIONS	A-19
4-15	1. Sewesterman		
4-19	The State of the S		
4-19	OCI	'AL AND HEXADECIMAL	3. L
4-21	CON	IVERSION	B-1
4-21			
4-21	APP	ENDIX C	
4-22	ASC	II AND EBCDIC CHARACTER	
	COL	DES	C-1
0000			
A-1			
FIC:	IDEC		
HIGU	IKES		
	2-4	Interface Information Block	2-13
1-1	2-5		2-15
	2-6		2-16
1-5	2-7		
1-6	2-8		
=	2-9	Transfer Addresses	
. 1-7			
1-7	3-1	Assembler Instructions and Controll	er
•			3-2
2-2	3-2	9	3-5
2-7	.	Table Comming a Communa	<i>-</i> 0
	3-48 3-48 3-49 4-1 4-1 4-2 4-2 4-4 4-5 4-6 4-7 4-7 4-9 4-11 4-13 4-15 4-19 4-21 4-21 4-21 4-22 A-1 FIGU 1-1 1-5 1-6 .1-7 1-7 2-2 2-3	3-48 COM 3-48 Ins 3-49 PIC CB REGI Coi Sta 4-1 Int 4-1 Sta 4-2 Int 4-2 CON' 4-4 INFO 4-4 Use 4-5 Ext 4-6 STAT 4-6 Coi 4-7 Sta 4-7 PIC 4-7 If 4-9 Use 4-11 CON 4-13 Int 4-13 Int 4-13 Int 4-14 Int 4-15 Int 4-15 Int 4-16 Int 4-17 Int 4-18 Int 4-19 In	3-48

3-4	Sample Code: Retrieving Interrupt Information	3_12	A-2	Status Registers: Asynchronous Interrupts A-7
3-5	Action: Retrieving Interrupt		A-3	Status Registers: Synchronous
3-6	Information		A-4	Interrupts A-8 Control Block A-11
3-0	Standard Return Format	3-22	A-4 A-5	Controller Information Block A-12
4.4	Error Classifications	4.2	A-6	Interface Information Block A-12
4-1	Error Classifications	4-4	A-7	Unit Information Block A-13
A-1	Command Registers	A-5	11-7	Cint information block A-15
		TABL	.ES	
		Page		Page
1-1	Model 6236/6237 Performance	0	4-10	Program Load Command: Status
	Specifications	1-2		Register B 4-20
1-2	Model 6239 Performance		4-11	Get and Set Command: Status
	Specifications	1-3		Register B 4-21
	•		4-12	Get List Status Command: Status
2-1	Program-Accessible Controller			Register A 4-21
	Registers	2-1	4-13	Reset Command: Status Register B 4-22
2-2	Test Conditions	2-4		Ç
2-3	Pulses	2-4	A-1	Controller/Drive Specifications A-1
2-4	Control Block Contents	2-8	A-2	Model 6239/6240 Controller/Drive
2-5	Host/Controller Interface			Specifications A-2
	Addresses		A-3	PIO Commands A-4
2-6	Mapping Options		A-4	CB Commands A-4
2-7	Determining an Interrupt Type	2-25	A-5	Accumulator Contents A-5
			A-6	Status Register Contents:
3-1	Accumulator Contents	3-8	_	Asynchronous Interrupts A-7
3-2	Status Register Contents:		A-7	Status Register Contents:
	Asynchronous Interrupts	3-15		Synchronous Interrupts A-9
3-3	Status Register Contents:	0.4=	A-8	CB Status Word A-14
3-4	Synchronous Interrupts PIO Control and Initialization	3-17	Λ-9	CB Error Status Word A-14
3-4	Commands	2 22	A-10 A-11	CB Unit Status Word A-15 Command Status: Execution State . A-15
3-5	PIO Extended Status Commands		A-11 A-12	Command Status: Execution State . A-15 Command Status: Start List A-15
3-6	PIO List Control Commands		A-12 A-13	Command Completion Status A-16
3-7	PIO State Independent Commands	3-37	A-14	Asynchronous Interrupts A-16
3-8	PIO Valid While Reset Commands	3-40	A-15	Unit Status Word A-17
3-9	General CB Commands		A-16	Begin: Status Register B After an
3-10	CB Read Commands		11 10	Error A-17
3-11	CB Write Commands		A-17	Program Load: Status Register B
				After an Error A-17
4-1	Controller States Terminology	4-1	A-18	Get and Set Command Status A-18
4-2	Error Reporting		A-19	Get List Status Command Status A-18
4-3	CB Status Word	4-8	A-20	Reset Command Status A-18
4-4	CB Error Status Word		A-20	Mapping Options A-19
4-5	CB Unit Status Word			
4-6	Command Status		B-1	Octal Conversion Chart B-1
4-7	$ \ \ Command Completion Status \ldots .$			
4-8	Asynchronous Interrupts	4-16	C-1	ASCII Character Codes C-1
4-9	Begin Command: Status Register B	4-20	C-2	EBCDIC Character Codes C-4

CHAPTER 1 INTRODUCTION

This chapter gives an overview of the Model 6236/6237 and the Model 6239/6240 Data General Disk Subsystems. It briefly describes the hardware and software that make up the subsystems. Although there are some hardware differences between the Model 6236/6237 and the Model 6239/6240 disk subsystems, the programming of the two subsystems is the same.

Throughout this manual the term "Model 6236/6237" is used whenever a statement applies to both the Model 6236 and the Model 6237 Disk Subsystems. Likewise, "Model 6239/6240" refers to both the Model 6239 and the Model 6240 Disk Subsystems. Wherever differences exist, the subsystems will be identified by model number. Where a statement pertains to both the Model 6236/6237 and the Model 6239/6240 disk subsystems, the subsystems will be referred to collectively as the "Disk Subsystem".

The rest of this manual details the instruction set and typical programming techniques for the Disk Subsystem.

DISK SUBSYSTEM HARDWARE

A Model 6236 subsystem has a high-performance rack-mounted, moving-head disk drive. A Model 6237 subsystem has three 6236 drives in a meter-high cabinet. The Model 6239 subsystem consists of a self-contained, random access, moving-head, nonremovable media disk drive, and a controller PCB.

Figure 1-1 shows the components of a typical Disk Subsystem and its host interface.

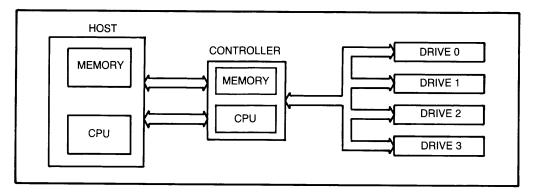


Figure 1-1. Overview: Disk Subsystem Interface

The Model 6240 disk drive consists of three 6239 disk drives installed in a standalone, meter-high cabinet.

The Disk Subsystem controller contains all the logic necessary to transfer data between four disk drives and main memory, via a Burst Multiplexer Channel Interface (BMCI).

Disk Drives

The Disk Subsystem drive logic is intelligent, and performs basic high-level drive functions such as track-following servo, error correcting codes, and advanced data encoding. It consists of a Data General microECLIPSE® processor, a read/write processor, and 20 kilobytes (Kbytes) of local memory. The micro-ECLIPSE processor provides an intelligent interface between a drive and a controller. Consequently, the drive and controller can handle error correction without interrupting the host. The read/write processor handles read and write operations.

Table 1-1 lists performance specifications for the Model 6236/6237 subsystem.

Table 1-1. Model 6236/6237 Performance Specifications

Description	Specification
Formatted capacity	354 Mbytes
Number of sectors/track	56
Data bytes/sector	512
Platter diameter	14 in.
Tracks per inch	714
Cylinder capacity	458.72 Kbytes
Number of cylinders (total)	789
User	786
Controller reserved	1
Internal diagnostic	1
Field service diagnostic	1
Access time	
Single track	5 ms
Average (1/3 stroke)	20 ms
Maximum (full stroke)	35 ms

Table 1-2 lists performance specifications for the Model 6239 subsystem. The Model 6240 disk drive subsystem consists of a controller and 3 independent Model 6239 disk drives. Thus, the Model 6240 records data on 24 disk surfaces as opposed to the 8 surfaces available on the Model 6239.

Table 1-2. Model 6239 Performance Specifications

Description	Specification
Formatted capacity	592 Mbytes
Number of sectors/track	75
Data bytes/sector	512
Platter diameter	14 in.
Tracks per inch	800
Cylinder capacity	606.21 Kbytes
Number of cylinders (total)	981
User	978
Controller reserved	1
Internal diagnostic	1
Field service diagnostic	1
Access time	
Single track	5 ms
Average (1/3 stroke)	21 ms
Maximum (full stroke)	38 ms

The head/disk assembly is sealed at the factory and will not need on-site service.

Controller

The Disk Subsystem controller is an intelligent, high-speed disk controller. It provides an interface between a drive and its host computer.

The controller board is a 15-inch multilayer board that plugs into an ECLIPSE ® or ECLIPSE MV computer backpanel. It contains its own CPU and memory. This microprocessor provides 6 program-accessible registers, 16 Kbytes of RAM, and 8 Kbytes of ROM.

The microprocessor provides an intelligent interface between the controller and its host computer. This interface lets the controller offload many functions previously performed by the device driver software. The controller is capable of performing tasks such as automatic retry on errors, 17-bit error correction, and self-diagnosis. In addition, it can skip defective sectors automatically. It incorporates a unique method for disk backup using modified sector flags.

The controller communicates with the host over two buses:

- A Burst Multiplexer Channel (BMC) for the controller's high disk data transfer rate of 1.7 megabytes (Mbytes) per second on the Model 6236/6237 and 2.2 Mbytes per second on the Model 6239/6240.
- A bus for programmed I/O (PIO) transfers.

The Disk Subsystem can support one or two controllers. Two controllers provide resource sharing through dual-porting protocols (see Chapter 2).

Each controller can support up to 4 drives. The controller interfaces with its disk drives through a 50-pin bulkhead connector. The data interface between the controller and its drives is an 8-bit bidirectional command/data bus. Data transfers are controlled by a 2901-based processor.

DISK SUBSYSTEM SOFTWARE

Programming techniques differ for intelligent and nonintelligent disk subsystems. The Model 6236/6237 and Model 6239/6240 are intelligent disk subsystems. This section describes some of the differences and similarities between an intelligent and a nonintelligent disk subsystem.

In nonintelligent disk subsystems, all drive and controller tasks are initiated by device driver programs. Since the controller and drives in the Model 6236/6237 and the Model 6239/6240 have their own processors, they can perform many basic tasks without host intervention. This frees the host for other work. Device driver programs can concentrate on error handling and can run more efficiently.

With the Model 6236/6237 and Model 6239/6240 Disk Subsystem, you still code assembler I/O instructions that tell the controller to perform drive operations. However, once you pass an initial command to the controller, it can fetch other commands and data by itself, without interrupting the host. The controller interrupts the host only if the host requests interrupts or if an error occurs.

The Model 6236/6237 and Model 6239/6240 Disk Subsystem has a set of program-accessible registers and flags, just like nonintelligent disk subsystems. In addition to these, the Disk Subsystem uses two special data structures for transferring commands, data, and status information. Control blocks specify drive operations and data transfers. Information blocks specify drive and controller options and, when an error occurs, status information.

Device driver programs for nonintelligent disk subsystems include code for transferring data and interpreting interrupts. Similarly, a device driver for a Model 6236/6237 or a Model 6239/6240 Disk Subsystem includes code for creating and manipulating control and information blocks.

Figure 1-2 shows the relationship between these data structures and the Disk Subsystem hardware.

NOTE: Flags (Busy and Done) and pulses (C, P, and S) are loaded into the same registers.

Typical program execution, using components shown in Figure 1-2, is as follows:

- Using standard Data General I/O assembler instructions, the device driver program loads controller commands and pulses into the host accumulators (AC0, AC1, AC2, and AC3).
- A Start Pulse sends the contents of the host accumulators to the controller's registers over the PIO.
- The controller's CPU executes the specified command.
- If the command specifies a control block address, the controller fetches the control block from host memory over the BMC. It then performs the drive operation specified in the control block.

If the command specifies an information block address, the controller transfers a copy of the block to or from host memory over the BMC.

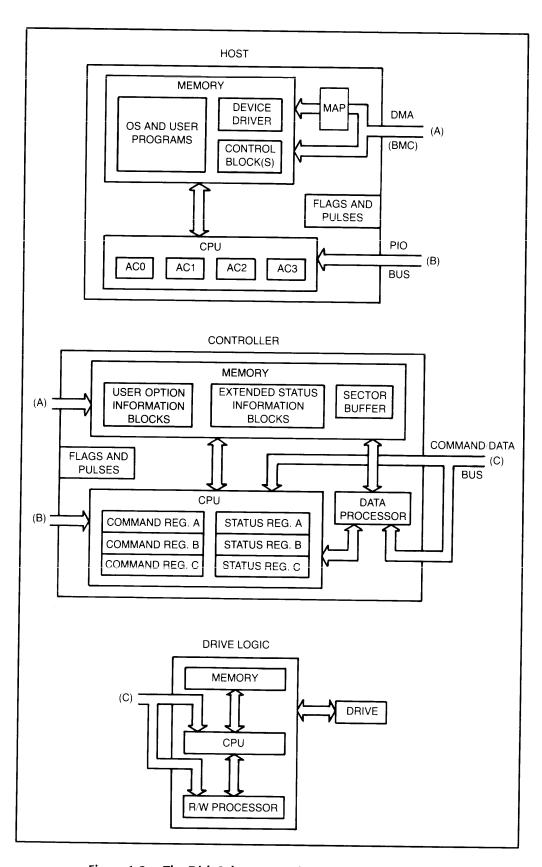


Figure 1-2. The Disk Subsystem and Its Software Components

Commands and Data Structures

This section tells more about the disk subsystem commands and data structures.

Commands

Disk Subsystem commands control the interactions of the host, the controller, and the disk. We categorize these commands as follows:

- Instructions The instruction set consists of Data General assembler I/O instructions, which transfer information between the host and the controller.
- PIO Commands The programmed input/output (PIO) command set consists of controller commands. They manipulate the controller, control blocks, and information blocks. You send PIO commands to the controller with assembler I/O instructions.
- CB Commands The control block (CB) command set consists of disk commands. To perform a disk operation, you first load a control block with a disk command operation code. Then, you execute the control block by issuing a PIO command via an instruction.

Control Blocks

A control block (CB) is a user-defined block stored in host memory. It gives the controller all the information it needs to perform a drive operation (such as operation code or how much data to transfer). When an operation is completed, the controller writes return and error information into the control block. You can read this information and take any necessary action.

Figure 1-3 shows the basic format of a control block.

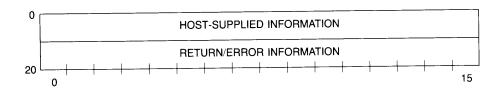


Figure 1-3. Control Block

You can link a list of control blocks into a CB list. This lets you tell the controller where the first control block is. It can then fetch all control blocks in the list without interrupting the host. The controller will interrupt the host only when the list is complete or if an error occurs.

Information Blocks

There are two kinds of information blocks: user option and extended status. They are stored in controller memory and can be retrieved by the host via I/O instructions.

User option blocks define options that you can specify. Currently, you can define three kinds of options: controller, interface, and unit. Controller options specify the number of controller retries for soft errors. Interface options specify universal information the host needs to know about a device. Unit options specify universal and device-specific information about a device.

Figure 1-4 illustrates the basic format of a user option information block.

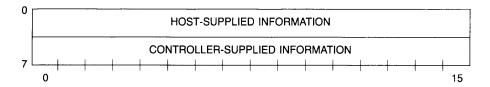


Figure 1-4. User Option Information Block

If an error occurs during a controller or drive operation, the controller writes status information into its status registers and the control and extended status blocks. You can usually resolve the error with the information in the status registers and control block. However, if the error is severe, you will want to store a copy of the extended status block. Data General personnel will use this block to analyze the problem. Figure 1-5 illustrates the basic format of an extended status block.

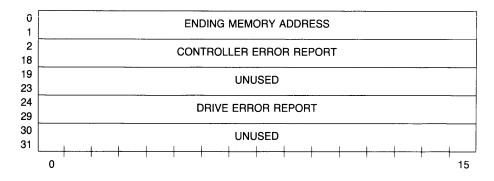


Figure 1-5. Extended Status Information Block

Programming the Disk Subsystem

Programming the Disk Subsystem involves writing software that manipulates the subsystem hardware. In this manual, we present Disk Subsystem programming concepts as follows:

- Host/Controller Interface (Chapter 2)
- Commands and Instructions (Chapter 3)
- Status Interface (Chapter 4)

Chapter 2 describes normal controller/drive operations: sending instructions to controller registers, issuing pulses, manipulating flags, and creating and defining data structures. Chapter 3 details the commands you use for all subsystem host, controller, and drive operations. If an error occurs, or if you need more information about command status, Chapter 4 describes controller and drive status information.

CHAPTER 2 HOST/CONTROLLER INTERFACE

This chapter details the components of the host/controller interface:

- Controller registers
- Busy and Done flags
- Pulses
- Control and information blocks
- Addressing
- Interrupts

Registers, flags, and pulses are common to all disk subsystems. Control and information blocks are concepts introduced with recent Data General subsystems. The controller uses special addressing conventions and interrupt schemes for handling these blocks.

CONTROLLER REGISTERS

The six program-accessible controller registers are listed in Table 2-1.

Table 2-1. Program-Accessible Controller Registers

Assembly Language Register Name	Controller Register Name	Number of Bits
Data In A (DIA)	Status Register A	16
Data In B (DIB)	Status Register B	16
Data In C (DIC)	Status Register C	16
Data Out A (DOA)	Command Register A	16
Data Out B (DOB)	Command Register B	16
Data Out C (DOC)	Command Register C	16

Use assembler I/O instructions to transfer information between these registers and host accumulators. The I/O instructions have the same names as the assembly language registers: DIA, DIB, DIC, DOA, DOB, and DOC. The DOA, DOB, and DOC instructions let you send PIO commands to the controller's command registers. The DIA, DIB, and DIC instructions let you retrieve interrupt information from the controller's status registers.

Command Registers

The command registers are used to issue PIO commands to the controller.

Figure 2-1 shows the command registers. Registers A and B contain the command's arguments. If these arguments are addresses, register A contains the high-order word and register B contains the low. Register C contains the PIO command code and the return request bit setting. This setting indicates whether the controller will generate a synchronous interrupt when it completes a command.

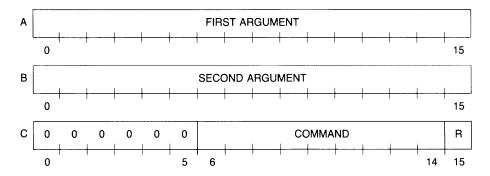


Figure 2-1. Command Registers

You may load these registers only if the Busy flag is clear. Attempting to load these registers when Busy is set will have unpredictable effects. Also, you may load the registers in any order. The controller does not begin PIO command execution until you issue an S pulse.

Status Registers

The controller writes interrupt return information into its status registers. (The two types of interrupts, asynchronous and synchronous, are discussed later in this chapter.)

Figure 2-2 shows the status registers. Registers A and B contain interrupt information. Register C contains these codes:

- Status bits that indicate the controller state
- Command completion status bits that indicate the type of interrupt
- An interrupt code if an asynchronous interrupt occurred
- The command code and return request bit setting of the most recent controller command if a synchronous interrupt occurred

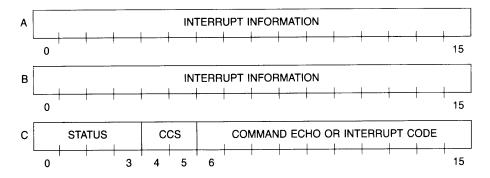


Figure 2-2. Status Registers

Bits 0-3 of status register C are valid only when the Busy flag is clear. The remaining bits of this register and all of status registers A and B are valid only when the Done flag is set.

BUSY AND DONE FLAGS

The Busy and Done flags serve two purposes:

- They indicate whether the controller is performing an operation or has completed one.
- They let you check a flag setting before issuing a command.

Busy Flag

The Busy flag indicates that the controller is performing an operation. This flag is set when the controller receives an S pulse from the host. It remains set until the controller finishes executing the PIO command started by the S pulse.

The host cannot issue a DOA, DOB, or DOC instruction while the Busy flag is set. The controller will accept new PIO commands only when its Busy flag is clear, in which case bits 0-3 of status register C are valid. These bits indicate the completion status of the previous PIO command.

Done Flag

The Done flag indicates that the controller has finished one operation and is waiting to begin another. This flag is set if the controller generates an interrupt. If it is set, the controller's status registers contain valid return information.

The Done flag is never set when the Busy flag is set. If an S pulse is issued while the Done flag is set, the controller clears the Done flag when it sets the Busy flag.

Checking Busy and Done Settings

You cannot issue certain instructions or pulses when one or the other flag is set. The SKP I/O instruction tests a flag setting before you perform an operation. If the test is true, the next assembler instruction is skipped.

For the SKP instruction, bits 8 and 9 are referred to as the t field. The t field bit settings, mnemonics, and test conditions it selects are listed in Table 2-2.

Table 2-2. Test Conditions

Bit Setting	Mnemonic	Next Instruction Skipped If:
00	BN	Busy flag is 1 (Nonzero)
01	BZ	Busy flag is 0 (Zero)
10	DN	Done flag is 1 (Nonzero)
11	DZ	Done flag is 0 (Zero)

To test for one of these conditions, append the appropriate mnemonic to the SKP instruction. For example, to test whether the Busy flag is set, issue the SKPBN instruction.

The next section tells you how to set the Busy and Done flags using pulses.

PULSES

You manipulate the controller's Busy and Done flags with bits 8 and 9 of an I/O assembler instruction. You use these bits to specify control functions called pulses. In most cases, you issue a pulse by appending its mnemonic to an I/O instruction name. For example, to issue an S pulse for the DOC instruction, code DOCS for the instruction name. Table 2-3 lists pulses and the I/O instructions you use to issue them.

Table 2-3. Pulses

Bit Setting	Pulse	I/O Instruction	Control Function
00	No Pulse	Reserved	No control.
01	S Pulse	DOAS, DOBS, DOCS, DIAS, DIBS, or DICS	Starts a PIO command (sets the Busy flag; clears the Done flag).
10	C Pulse	NIOC	Services an interrupt (clears the Done flag and pending interrupt).
11	P Pulse	NIOP	Sets the controller's Interrupt on Not Busy flag if the Done flag is clear.

NOTES:

1. S Pulse

- You must load the command registers with the proper command code and arguments before you issue this pulse.
- You only need to issue this pulse for one instruction. That is, after loading two of the three command registers, issue the S pulse when you load the remaining register.
- Do not load the command registers or issue this pulse if the Busy flag is set. If you do, you may overwrite information the controller is using for the current operation.

2. C Pulse

Do not load any command registers before issuing this pulse.

3. P Pulse

- If the Done flag is not set, this pulse sets the Interrupt on Not Busy flag.
- If the Interrupt on Not Busy flag is set and no other interrupts are pending, the controller will send an asynchronous interrupt to the host when the Busy flag clears. It then sets the Done flag and clears the Interrupt on Not Busy flag.
- You should not load any command registers before issuing this command.
- Use this pulse when you want the controller to notify the host when it can issue the next PIO command (when the Busy flag is clear).

CONTROL BLOCKS

Specifying Controller Operations

You use a control block (CB) to specify controller operations. It consists of transfer and return information.

To perform a certain controller operation (read or write), you must first define a control block for the operation in host memory. Then, transfer the CB address to the controller.

To perform multiple operations, you can create a linked list of CBs. Then, transfer the address of the first CB in the list to begin execution. (A CB list is a group of CBs connected by their link words. There is no upper limit to the number of CBs in a list; there may be as few as one.)

You pass a control block address to the controller as a PIO command argument or in the link words of another CB. This address is a double-word memory address.

You can specify a physical or logical CB address:

- If bit 0 of the address is clear, the controller interprets the address as a physical memory address.
- If bit 0 is set, the controller interprets the remaining 31 bits as a premapped logical address.

If you specify a logical address, it must be mapped by the host. Also, you must always reference a CB address the same way. That is, you cannot supply a CB address once as a logical address, and again as a physical address.

There are constraints on the location of a CB in main memory:

- If you pass a physical address to the controller (e.g., 0B0), you must ensure that the CB does not span two discontinuous physical memory pages.
- If you pass a logical address, the CB may reside anywhere.

See "Addressing Conventions" later in this chapter for more information.

CB Format

Figure 2-3 shows the format of a control block. A control block contains two types of information:

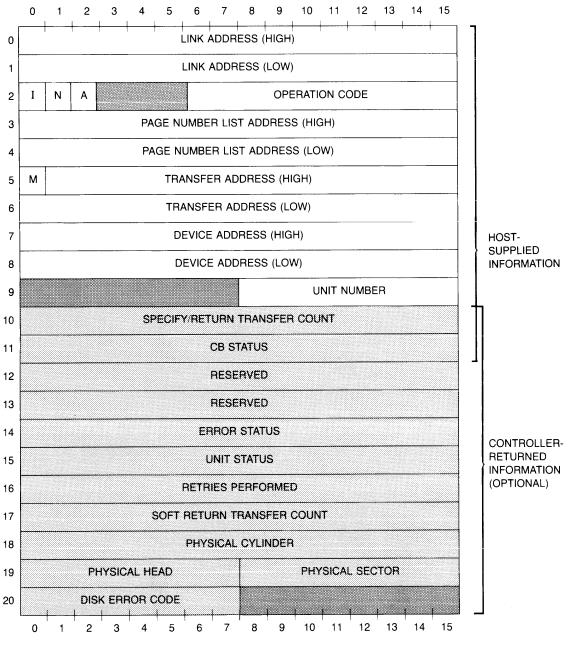
- Host-supplied information: words 0-11 (nonshaded)
 - You code values for these words. They contain control information, such as the address of the next CB in a list and the drive operation code.
- Controller return information: words 10-20 (shaded)

Table 2-4 describes the function of each word in a control block in more detail. You can request that the controller return from 0 to 11 status words when it completes a CB. You use the CB extended status field in the interface information block to specify the number of return words.

If you specify 0, the controller will return no status information, regardless of the result of the drive operation. If you specify from 2 to 11 and no errors occur, the controller will return words 10 and 11 only. If any error occurs, the controller will return the number of requested words.

We recommend that you always specify at least two return words in case an error does occur.

(The interface information block is detailed later in this chapter.)



LEGEND:

- ☐ HOST-SUPPLIED INFORMATION
- **UNUSED**
- CONTROLLER-RETURNED INFORMATION

Figure 2-3. Control Block

Table 2-4. Control Block Contents

Word	Bits	Name	Contents or Function
0 and 1	0-15	Link Address	Address of the next CB in the list (see Note 1).
2	0	Interrupt Bit	If set: The controller generates an unconditional interrupt when it completes a CB (see Note 2).
			Not set: The controller generates an interrupt only if the link address is 0, or if an error occurs.
	1	No Retries Bit	If set: Every error appears hard. No retries occur, regardless of the controller information block values.
			Not set: Retries occur according to controller information block values.
	2	Atomic Bit	If set: Provided the next CB in the list is for the same unit, the controller executes that CB regardless of options.
			Not set: CB execution order is affected by the optimization bit in the Unit Information Block.
	3-5	_	Unused.
	6-15	Operation Code	Controller operation to be performed:
			Octal Code Meaning 000 No Operation 100 Write 101 Write/Verify 104 Write Single Word 105 Write/Verify Single Word 142 Write with Modified Bit Map 200 Read 201 Read/Verify 205 Read/Verify Single Word 210 Read Raw Data 220 Read Headers 242 Read with Modified Bit Map 400 Recalibrate Disk These operations, called CB commands, are detailed in Chapter 4.
3 and 4	0-15	Page Number List Address	Address of the page number list in host memory (see Note 3).
5	0	Mapping Bit	If set: The transfer address (words 5 and 6) is a logical premapped address.
			Not set: The transfer address is a physical address.
5 and 6	1-15 and 0-15	Transfer Address	Starting address of the data transfer (see Note 4).
7 and 8	0-15	Device Address	Logical sector address of the device that is to receive the data transfer (see Note 5).

Table 2-4. Control Block Contents (continued)

Word	Bits	Name	Contents or Function
9	0-7		Unused.
	8-15	Linit	
	0-15	Unit Number	The number of the unit you want to perform the operation on.
10	0-15	Specify/ Return	The transfer count before and after the CB is executed:
		Transfer Count	Specify Transfer Count: The number of data sectors you want to transfer.
			Return Transfer Count: The number of data sectors transferred (see Note 6).
11	0-15	CB Status	Specifies the CB status before and after the operation. This word must be 0 before the CB is executed. After execution, this word contains the following status information:
			Bit Meaning If Set:
			Any CB hard execution error Interpretation error
			Soft errors in execution occurred; controller
			recovered 3 CB termination by Cancel List command
			4 ECC correction needed 5 ECC correction failed
			6-14 Unused 15 CB Done bit
12 and 13	0-15	_	Reserved.
14	0-15	Error Status	Bit Meaning If Set:
		Otatus	0 Interrupt Timeout
			1 Drive Interface Fault 2 2901 Timeout
	•		3 Buffer Overflow (Data Late)
			Controller Detected Checksum Error Drive Error
			6 BMC Timeout
			7 Ending Memory Address Error
			8 Data Checksum Error 9-10 Reserved
			11 Verify Error
			12 BMC Error 13 Data Parity Error
			14 ECC Detected Error
			15 Header Noncompare
			These bits indicate the last error encountered for any CB and are valid whenever bit 0 or 2 of the CB status word is set (see Note 7).
15	0-15	Unit Status	Bit Meaning If Set:
			0 Command Failed
			1 Power Failed 2 Ready

Table 2-4. Control Block Contents (continued)

Word	Bits	Name	Contents or Function
			3 Busy 4-5 Port Reserve Bits 6-7 Unit Number 8 Logic Fault 9 Power Fault 10 Servo Data Fault 11 Positioner Fault 12 Read/Write Fault 13 Bus Fault 14-15 Reserved These bits apply to the unit specified in word 14; its number is echoed in bits 6-7.
16	0-15	Retries Performed	Total number of retries the controller performed (see Note 8).
17	0-15	Soft Return Transfer Count	The number of data sectors transferred before the final CB error (see Note 9).
18	0-15	Physical Cylinder	Physical cylinder address where the error occurred.
19	0-7	Physical Head	Physical head address where the error occurred.
	8-15	Physical Sector	Physical sector address where the error occurred.
20	0-7	Disk Error Code	Represents one of these drive errors (see Note 10): Bus Fault Logic Power Checkpoint Positioner Read/Write
	8-15	_	Unused

NOTES:

1. The high-order bit (word 0; bit 0) of the link address indicates whether this address is logical or physical:

If set: Logical address premapped by the host.

Not set: Physical address.

2. Set the interrupt bit if you want to know when each CB in a list completes. When the controller finishes executing a CB, it generates an asynchronous interrupt and then continues executing the list.

The controller will not generate the interrupt in two situations:

- When the last CB in a list is completed.
- When an error occurs that requires the controller to generate an interrupt.

3. The page number list contains physical page numbers where logical transfer addresses are mapped. The first page number in the list must map the first logical page of the transfer. The list must contain a page number for every page in the transfer.

The high-order bit (word 3; bit 0) indicates whether this address is logical or physical:

If set: Logical address premapped by the host.

Not set: Physical address. (The page number list must be stored contiguously in host memory.)

4. The transfer address interpretation (physical or logical) is based on the mapping bit (word 5; bit 0).

If you are using a page number list to map the transfer, the M bit must be 0 and this field will contain the logical address of the buffer.

- 5. The controller interprets the device address as a logical block number on the disk. A logical block is a sector. The controller converts the logical address to a physical one.
- 6. The controller writes the return transfer count to indicate the number of sectors actually transferred. The controller writes this word before it writes the status word.

If a hard or soft error occurs (with interface information block S bit set), adding the contents of this field and the device address field gives you the error's logical address.

- 7. Chapter 4 describes the error status word in detail. Briefly, if a hard execution error occurs (CB status word bit 0=1), these bits represent hard errors. If a soft error occurs (CB completed successfully; CB status word bit 2=1), these bits represent soft errors. The retries performed field (word 16) contains the number of attempts made to recover from soft errors.
- 8. If the number of retries performed is not 0, the error status word (word 14) indicates what kind of soft or hard error occurred.
- 9. The soft return transfer count applies to soft errors only. If a hard error occurs, this field contains the same value as the return transfer count.
- 10. Each error code represents the 8 high-order bits of the error code generated by the drive in error. If the drive cannot recover from one of these errors, it converts the error code to a hexadecimal code and displays that code on its front panel. Data General personnel use these codes to determine what is wrong with a drive.

INFORMATION BLOCKS

Nonintelligent controller programs use PIO (Programmed I/O) to exchange both commands and status information. Such controllers must start each data transfer with a separate command. With this Disk Subsystem, however, the host uses PIO to define and transfer only mapping information.

The controller handles all other control status transfers via direct memory access (that is, across the BMC). The transfer agents are user-defined control blocks and predefined information blocks. Control blocks were described above. The following sections describe the two current types of information blocks: user options and error status. Both types of blocks are stored in controller memory.

User Options

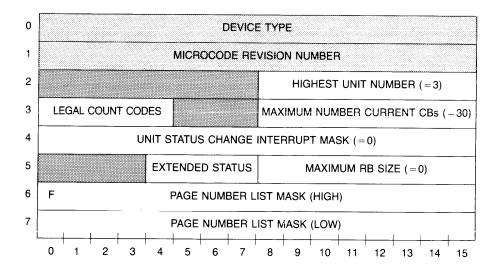
User options are options that you can specify or change. Currently, user option blocks are interface, controller, and unit blocks.

Certain commands let you retrieve and reassign values for each block. A Get command lets you retrieve the current information block. You can then examine or change the block's values by using an offset from the block's address in host memory. A Set command lets you set any new values you assign.

Interface Information Block

This block contains universal information that the host needs to know about a device. The information is the same for any device attached to this controller. The controller microcode sets this information at power-up. To access this information, you use the Get (031) and Set (030) Interface Information commands detailed in Chapter 3.

You cannot change some interface information block fields. If you attempt to do so, the controller will use the microcode values anyway. Alterable fields are not shaded in Figure 2-4.



LEGEND:

- ☐ HOST-SUPPLIED INFORMATION
- UNUSED
- CONTROLLER-SUPPLIED INFORMATION

Figure 2-4. Interface Information Block

Word	Bits	Name	Contents or Function
0	0-15	Device Type	Contains (101) for a Model 6238-B Controller, or (100) for a Model 6238 Controller: The device type code. You cannot change this value.
1	0-15	Microcode Revision Number	The microcode revision number for the controller. You cannot change this value.
2	0-7	_	Unused.
	8-15	Highest Unit Number	Contains 3: The highest unit number. You cannot change this value. (1 to 4 drives can be attached to a controller. Each drive is assigned a unit number from 0 to 3).
3	0-4	Legal Count Codes	The legal count codes for a Set command. This controller counts sectors only. Therefore, the only legal sector count value is 01000 ₂ .
	5-7	_	Unused.
	8-15	Max Number Current CBs	Contains 30: The maximum number of current CBs for the controller. You cannot change this value.
4	0-15	Unit Status Change Interrupt	Masks word 15 of the CB (Unit Status word) and causes an interrupt if any bits change state.
		Mask	Default: Always 0; not supported on this controller.

Word	Bits	Name	Contents or Function
5	0-3	_	Unused.
	4-7	CB Extended Status	The number of extended CB words: 0 to 11 (see Note 1). Default: 11
	8-15	Max RB Size	Contains 0: The maximum return block size. Since this controller does not use return blocks, this value is always zero.
6	0	Page Number List Format	If set: Page number list entries are two words long. Not set: Page number list entries are one word long.
		ļ	Default: 0 at power-up or Reset.
	1-15	Page Number List Mask	The high-order word of the Page Number List Mask (see Note 2).
		(high)	Default: 0 at power-up or Reset.
7	0-15	Page Number List Mask (low)	The low-order word of the Page Number List Mask (see Note 2).
		(1044)	Default: 0 at power-up or Reset.

NOTES:

1. You can specify that the controller return from 0 to 11 words of status information (CB words 10-20). If you specify 0, the controller does not write any CB return information after it completes a CB, regardless of the result of the operation. If you specify from 2 to 11 words and no errors occur, the controller will write CB return words 10 and 11 only. If an error occurs, the controller will write all requested CB return words.

We recommend that you always specify at least two return words in case an error does occur.

2. Words 6 and 7 of the interface information block tell the controller how to interpret entries for the page number list. The F bit (page number list format bit) indicates whether the entries are single (=0) or double (=1) word entries. The remainder of word 6 and all of word 7 form a 31-bit mask. This mask is logically ANDed with the page number list entry to produce a physical page number. If the F bit is not set, then only the low-order portion of word 7 is used. The host can use the extra bits since they will be masked out by the controller.

Controller Information Block

This block contains device-specific information about the controller. The controller microcode sets this information at power-up. To access this information, you use the Get (033) and Set (032) Controller Information commands detailed in Chapter 3.

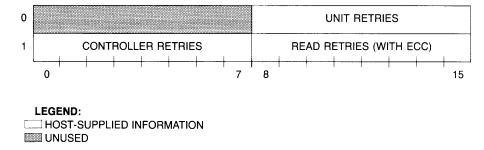


Figure 2-5. Controller Information Block

Word	Bits	Name	Contents or Function
0	0-7		Unused.
	8-15	Unit Retries	The maximum number of times the controller will try to execute a CB for a drive that is reporting an error.
			Default: 0 at power-up.
1	0-7	Controller Retries	The maximum number of times the controller will try to execute a CB with controller errors occurring (Timeout, ECC).
			Default: 0 at power-up.
2	8-15	Read Retries (with ECC)	The maximum number of correction retries the controller will attempt.
		(**************************************	Default: 0 at power-up.

Retry Strategy

The retry fields apply to CB execution errors only. They specify the number of unit, controller, and read retries.

Unit retries are independent of controller and read retries. During a unit retry, the controller issues a soft reset to the drive. Since many disk errors are mechanical (such as excessive spindle speed or temperature), they may not respond to a soft reset and the retry count will be quickly exceeded.

If a soft reset fails or if the controller times out a disk operation, the controller issues one hard reset to the drive. A hard reset resets the drive logic. If the hard reset fails, the controller declares an error: it sets bit 5 (Drive error) in the CB error status word and writes the CB unit status word.

Controller retries and read retries are related. Error correction code (ECC) errors are the most common type of errors. The controller retry count indicates the maximum number of rereads without using ECC. A read retry is attempted if the controller retry count is exceeded, and the controller senses an ECC error.

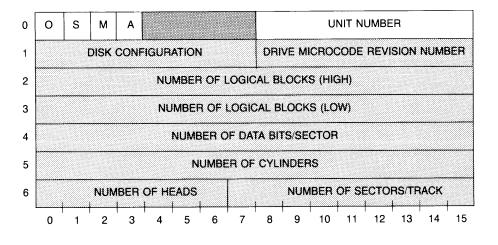
If the controller retry count is exceeded and the error is not an ECC error, a hard CB execution error occurs. The controller indicates the type of error in the CB error status word.

The controller writes the number of attempted retries into the CB retries performed field.

Unit Information Block

There is one unit information block for each unit attached to a controller. The unit information block contains everything needed to describe a unit, both universal information (word 0) and device-specific information (words 2-6). The controller microcode sets this information at power-up. To access this information, you use the Get (035) and Set (034) Unit Information commands detailed in Chapter 3.

You cannot change some unit information block fields. If you attempt to do so, the controller will use the microcode values anyway. Alterable fields are not shaded in Figure 2-6.



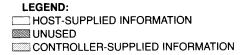


Figure 2-6. Unit Information Block

Word	Bits	Name	Contents or Function
0	0	Optimization	If set: No restriction on when the controller executes current CBs.
			Not set: The controller must follow certain rules for selecting the next CB to run (see the "Optimization" section below).
			Default: 0 at power-up, IORST, or Reset.
	1	Interrupt on	If set: The controller will interrupt if a soft CB error occurs.
		Soft CB Errors	Not set: The controller will not interrupt if a soft CB error occurs.
			Default: 0 at power-up, IORST, or Reset.
	2	Modified Bit	If set: All write operations set the modified bit for a sector.
			Not set: Write operations clear the modified bit for a sector.
			Default: 0 at power-up, IORST, or Reset.
			Note that you can issue the CB Read with Modified Bit Map command to back up the modified sectors. To clear the modified sector bits, read the modified sectors, clear the unit information block modified bit, and rewrite the sectors.
	3	Alternate Ports	Controls dual-ported operations. Dual-ported operation strategy involves optimization and alternate ports bit settings (see below).
			Default: 0 at power-up, IORST, or Reset.
	4-7	_	Unused.
	8-15	Unit Number	The unit you want information about. Possible values are 0-3.
1	0-7	Disk Configuration	Bit Meaning If Set: 0 Moving Heads 1 Fixed Heads 2 Fixed/Removable Media 3 Dual-Ported 4-7 Unspecified You cannot change any of these values. Bits 0-3 are set to 1001 for this controller.
	8-15	Drive Microcode Revision	The current revision number of the drive's microcode.
2 and 3	0-15	Number of Logical Blocks	Contains 1,157,952 on the 6239/6240 or 691,680 on the 6236/6237: The high- and low-order words for the number of disk sectors you can access. You cannot change this value.
4	0-15	Data Bytes/Sector	Contains 512: The number of data bytes per sector. You cannot change this value.

Word	Bits	Name	Contents or Function
5	0-15	Cylinders	Contains 978 on the 6239 6240 or 786 on the 6236 6237: The number of cylinders per disk. You cannot change this value.
6	0-7	Heads	Contains 16: The number of heads per disk. You cannot change this value.
	8-15	Sectors/ Track	Contains 75 on the 6239/6240 or 56 on the 6236/6237: The number of sectors per track. You cannot change this value.

Optimization

CB optimization improves overall performance by restricting current CB execution. You choose execution order by setting or clearing the optimization bit.

If optimization is enabled (optimization bit set), the controller can execute current CBs at any time. For example, if five lists contain CBs for the same unit, the controller does not have to complete one list before starting the next. Instead, it may skip from list to list and execute CBs for the same unit.

If optimization is disabled, the controller selects the next CB or CB list as follows:

- 1. When the controller completes a CB for a given unit, it must do one of two things:
 - Execute the next CB if it specifies the same unit.
 - Execute the next list in the queue for the same unit.
- 2. The controller queues a list behind all other lists for a unit in two cases:
 - When a Start List command starts a new list.
 - When a new unit is specified in a currently executing list.

NOTE: The controller preserves the execution order of CBs in a list, but not that of CBs across different lists.

Dual-port operation

There are three dual-port operations:

- 1. Reserve: holds a disk unit for exclusive use by one controller. If a controller reserves one port, a second controller cannot use the other port.
- 2. Release: frees a disk unit from exclusive use by one controller. A second controller can now use the other port; the first controller has relinquished control over the disk unit.

3. Trespass: overrides a controller's exclusive use of a disk unit. A second controller, via the free port, overrides the controller that is currently reserving the drive through the other port.

Reserve and release operations are performed internally by the controller. The CB unit status word indicates which port, if any, has the drive reserved. The unit information block optimization and alternate ports bit settings determine how the controller reserves or releases a drive:

1. O = x and A = 0

With this setting, the controller reserves a drive before executing a CB. It keeps the drive reserved as long as current CBs specify that drive. The controller keeps a list of current CBs for each drive. When the list is empty, the controller releases the reserved drive.

Specify this setting when you are using only one port.

This is the default setting at power-up or Reset.

2. O = 0 and A = 1

With this setting, the controller reserves a drive before executing a CB and releases it after executing the CB.

Specify this setting to ensure that a second controller can gain access to a disk. You can use this setting in any dual-ported system.

3. O = 1 and A = 1

With this setting, the controller reserves a drive before executing the CB. After executing the CB, it checks the CB optimization queue for that drive. If the queue is empty, the controller releases the drive. If the queue is not empty, the controller executes the next CB in the queue. The controller does not release the drive until the queue is empty.

Specify this setting to improve overall system throughput. It will take longer for each controller to gain access to a drive, but a controller is assured access.

NOTE: The CB optimization queue is guaranteed to empty because CBs are enqueued by starting unit addresses. If the controller fetches a CB with a lower starting address, it enqueues the CB in a second queue for that drive. Thus, the first queue will empty.

To perform the trespass operation, you issue the PIO Trespass command described in Chapter 3 . You will not need to issue this command often, since the optimization strategy usually frees a disk. However, you could use the Trespass command to get immediate access to a drive.

Extended Status

The controller maintains extended status information in its memory. This status is stored in the extended status information block, which consists of these 32 words:

Words	Block Name	Contents
0-1	Ending Memory Address	Control information and the BMC address.
2-18	Controller Error Report	A dump of the controller's 2901 registers. This dump contains disk information generated before and after an error.
19-23	-	Unused; all zeros.
24-29	Drive Error Report	An error code and drive status information.
30-31	_	Unused; all zeros.

The controller maintains one extended status block for each drive attached to it. Extended status is available after the first CB execution error. If you set the S bit in the unit information block (word 0; bit 1), the controller creates an extended status block for each error, including those that may occur on a retry. If you do not set the S bit, the extended status block reflects the last error that was generated.

When an error occurs, the controller:

- Logs the error in an error log on cylinder 0 of the drive.
- Freezes the disk subsystem.
- Writes status information into its status registers.
- Writes status information into the control and extended status information blocks.

Usually, you can resolve the error with the information provided in the status registers and control block. However, if the error is severe or recurring, you will want to store a copy of the drive's extended status information block for Data General personnel. You use the PIO extended status commands described in Chapter 4 to retrieve the appropriate block. After you store an extended status information block, you must issue a PIO Restart command to begin execution again.

ADDRESSING AND MAPPING CONVENTIONS

Information and data transfers for this subsystem occur via programmed I/O (PIO) and direct memory access (DMA). PIO transfers include mapping options and information and control block addresses. Once the controller receives this initial information, it can fetch additional information and data via DMA.

The sections below describe addresses, mapping options, and how this information is transferred.

Addressing Conventions

Table 2-5 lists the addresses you use for the host/controller interface.

Table 2-5. Host/Controller Interface Addresses

Address	Transferred By:	Points To:
Start List	PIO Start List command	The CB or CB list you want to execute.
Link	Control block; words 0 and 1	The next CB in a list (an address that links one CB to another).
Page Number List	Control block; words 3 and 4	The page number list in host memory. (This list contains physical page numbers where logical transfer addresses are mapped.)
Transfer	Control block; words 5 and 6	Where a data transfer should begin.
Information Block	PIO Get or Set command	A controller, extended status, interface, or unit information block.

Note that these addresses are 32-bit double-word memory addresses. They may be physical or logical addresses:

- If bit 0 is set, the remaining 31 bits represent a logical premapped address.
- If bit 0 is not set, the address is physical.

The controller will generate one of these errors if an address is outside of its fetching range:

- A PIO command execution error for start list or information block addresses.
- A CB interpretation error for any other address.

A mapped address is outside the controller's fetching range if bits 1-11 of the highorder word are not zero. An unmapped address is outside the range if bits 1-10 are not zero unless the Upstream Load mode is selected (see Table 2-6 for mapping options).

The assembly language programmer's reference manual for your system should describe how to create these addresses. Note that you should not use a physical address to address a block of memory that spans discontinuous pages.

The following section describes how to specify whether your addresses are logical or physical.

Mapping Conventions

Mapping options define whether address transfers are physical or logical. Before you can issue any address-specifying command, you must issue the Set Mapping command. This command, detailed in Chapter 3, lets you set the current mapping mode.

Table 2-6 lists mapping options and indicates which are valid with this controller.

Table 2-6. Mapping Options

Option		Meaning
DMA over the Data Channel		Direct memory access (DMA) occurs over the data channel.
		Default: Always 0; option not supported.
2.	DMA over the BMC	Direct memory access occurs over the BMC (see Note 1).
		Default: Always 1; only option supported.
3.	Mapping with Map Slot Load Interrupts	Mapping information must be provided by the host.
	·	Default: Always 0; not supported.
4.	Mapping with Internal BMC	Physical remapping occurs over the BMC (see Note 2).
	Physical Remapping	Default: 1 on power-up, Reset, or IORST.
5.	Mapping with Upstream Loading	Mapping information is loaded by the controller and occurs over the BMC (see Note 2). (Valid for 32-bit machines only.)
		Default: 1 on power-up, Reset, or IORST.
6.	Upstream Loading with Host Page Tables	This option is valid for 32-bit machines only. Mapping information is loaded from host page tables (see Note 2).
	Tables	Default: 1 on power-up, Reset, or IORST.
7.	First Map Slot Assigned	The first map slot assigned by the operating system (see Note 3).
	, 100.griou	Default: 0 on power-up, Reset, or IORST.
8.	Number of Map Slots Assigned	The number of map slots assigned by the operating system (see Note 3).
		Default: 0 on power-up, Reset, or IORST.

NOTES:

- 1. Option 2 specifies the data transfer mechanism, the BMC. Since all Disk Subsystem data transfers are made over the BMC, you never change this option.
- 2. Options 4, 5, and 6 let you specify logical transfer addresses (all three are set initially). Select only one (or none) of these options. If you set more than one bit, the controller will generate an error. If you specify none of the options, all transfer addresses must be physical or premapped.

3. Options 7 and 8 let you specify the first map slot and the number of slots. The controller will start with the first slot specified, and use the next sequential number of slots specified.

Address Transfers

The controller outputs these addresses on the BMC address bus. This bus contains a map bit and 21 address bits.

If the map bit equals 0, the address is a 21-bit physical address that looks like this:



Figure 2-7. Physical BMC Address

Bits	Name	Contents or Function
0	Map Bit	0 to indicate a physical address.
1-3	BK	Bank selection bits.
4-6	XCA	Extended channel address bits.
7-21	CA	Channel address bits.

If the map bit equals 1, the address is 20-bit logical address that looks like this:

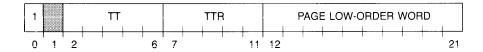


Figure 2-8. Logical BMC Address

Bits	Name	Contents or Function
0 1 2-6 7-11 12-21	Map Bit TT TTR Page Low Order Word	to indicate a logical address. Unused. Specifies one of 32 translation tables. Specifies one of 32 slots (registers) within a translation table. Specifies the low order word within a page.

As stated earlier, the high-order address bit determines whether an address is physical or logical. This determination affects how the address is transferred over the BMC.

Start list, link, page number list, and information block addresses are transferred the same way. If the address is physical, the BMC address points to a physical address in host memory. If the address is logical, it points to a segment in memory identified by the map bit, the map slot addressed, and the page offset.

Transfer addresses can also be logical or physical. If the address is logical, it is transferred like any other logical controller address. If it is physical, it can be further classified. Figure 2-9 shows classifications for transfer addresses.

INTERRUPTS

There are two kinds of interrupts: asynchronous and synchronous. An asynchronous interrupt occurs when the controller completes a CB or CB list, or when an error occurs during CB execution. A synchronous interrupt occurs after a PIO command executes.

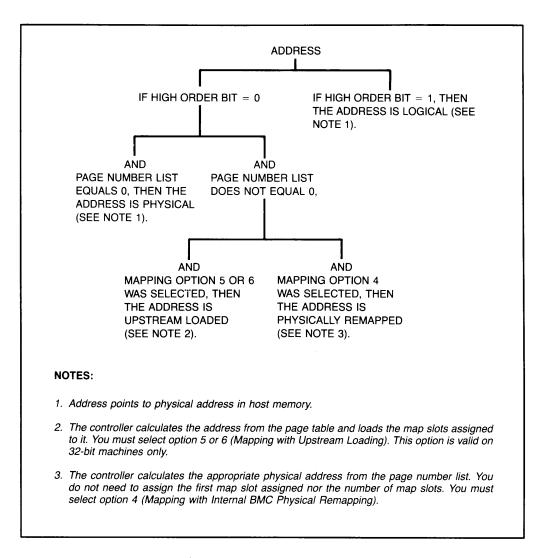


Figure 2-9. Transfer Addresses

Both types of interrupts are generated when the controller loads interrupt information into its status registers and sets its Done flag. However, how and when interrupts are sent to the host depends on the host's interrupt mask and the interrupt's priority:

- A synchronous interrupt is generated if two conditions are met:
 - The host has not masked out interrupts from the controller.
 - The command's return request bit is set.
- An asynchronous interrupt is generated if three conditions are met:
 - The host has not masked out interrupts from the controller.
 - There are no pending synchronous interrupts.
 - The controller's Busy and Done flags are clear.

The host interrupt mask bit is bit 7.

Synchronous interrupts have priority over asynchronous interrupts. If a synchronous interrupt occurs, synchronous return information will replace asynchronous return information in the status registers. The controller will reissue the asynchronous interrupt when you clear the Done flag (issue a C pulse). The host must service a pending asynchronous interrupt before the controller will issue the next asynchronous interrupt.

Your response to an interrupt depends upon the type of interrupt you are confronted with. Examine bits 4-5 of status register C to determine the type of interrupt (see Table 2-7). After handling the interrupt, you should issue a C pulse to service it. The controller will present the next pending interrupt, if any, by setting the Done flag again.

Table 2-7. Determining an Interrupt Type

Bit Setting	Meaning
00	Asynchronous interrupt: bits 10-15 contain an interrupt code that describes the reason for the interrupt.
01	PIO illegal command error: invalid command code.
10	PIO command execution error: unsuccessful command execution.
11	PIO command completed: successful command execution.

Chapter 3 describes how to retrieve interrupt information from the status registers.

CHAPTER 3 COMMANDS AND INSTRUCTIONS

The Disk Subsystem command set is divided into these categories:

- Instructions The instruction set consists of Data General assembler I/O instructions which transfer information between the controller and the host.
- PIO Commands The programmed input/output (PIO) command set consists of controller commands. They manipulate the controller, control blocks, and information blocks. You send PIO commands to the controller with assembler I/O instructions.
- CB Commands The control block (CB) command set consists of disk commands. To perform a disk operation, load a control block with a disk command operation code. Then, execute the control block by issuing a PIO command via an instruction.

This chapter details these instructions and commands.

INSTRUCTIONS

The instruction set consists of six standard Data General assembler I/O instructions: DIA, DIB, DIC, DOA, DOB, and DOC. Use these instructions to transfer information between the host and the controller. Transfer information is loaded into the controller's registers: command information is loaded into the command registers, and return information is loaded into the status registers. Figure 3-1 shows the relationship between assembler instructions and the controller's registers.

You use the DOA, DOB, and DOC instructions to send a PIO command to the controller. Command register C gets the command code and return request bit setting; command registers A and B get the command's arguments.

Use the DIA, DIB, and DIC instructions to retrieve status information from the controller. The kind of status information returned depends on whether the interrupt is asynchronous or synchronous.

Instruction Syntax and Internal Representation

Use the following syntax to code an instruction:

instruction_name[f] ac,DSKP

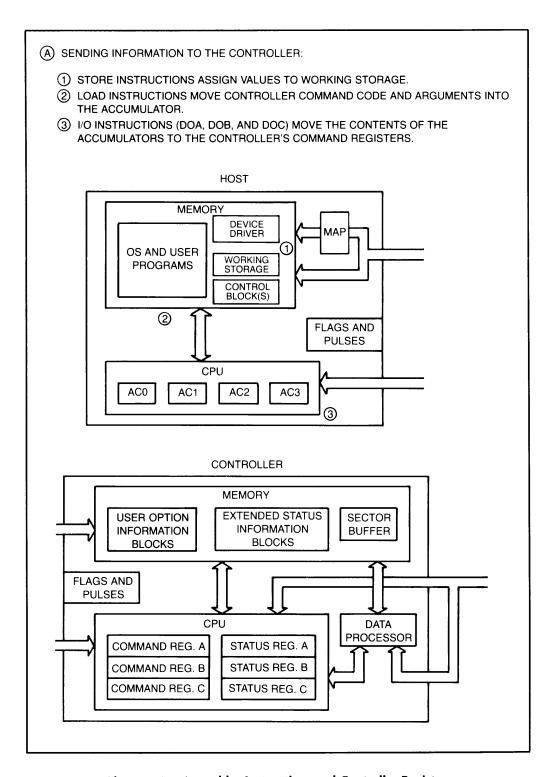


Figure 3-1. Assembler Instructions and Controller Registers

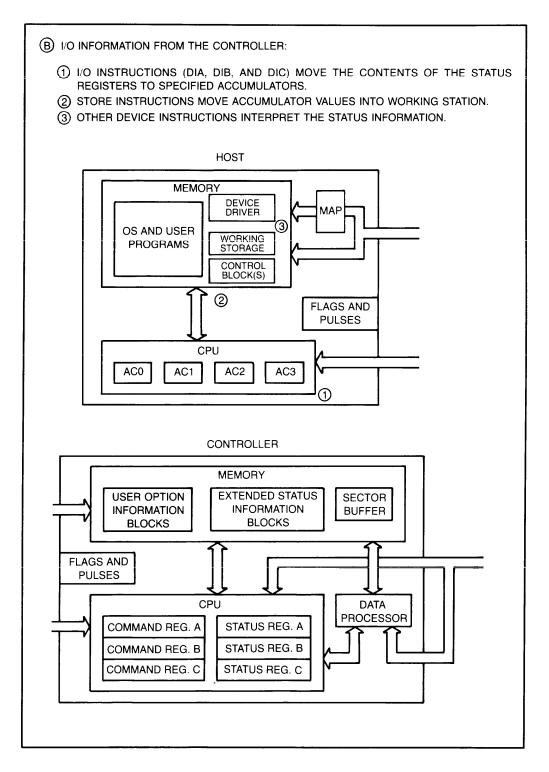


Figure 3-1. Assembler Instructions and Controller Registers (continued)

where:

instruction_name is the I/O instruction you want to issue:

DIA DIB DIC DOA DOB DOC

f is the pulse you want to issue (optional):

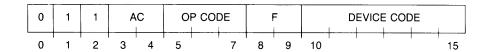
$$C = C$$
 pulse $P = P$ pulse $S = S$ pulse

ac is the host accumulator that contains the transfer information:

$$0 = AC0$$
 $1 = AC1$ $2 = AC2$ $3 = AC3$

DSKP is the assembler mnemonic for the device code.

The machine language syntax is represented internally as follows:



Bits	Name		Contents or Function		
0-2	_	Identifies an I/O type instruction.			
3-4	Accumulator	Specifies one of the host's four accumulators:			
		Bit Setting	Accumulator		
		00	AC0		
		01	AC1		
		10	AC2		
		11	AC3		
5-7	Operation Code	Specifies whi	ch I/O instruction to perform:		
		Op Code	Instruction		
		001	DIA		
		011	DIB		
		101	DIC		
		010	DOA		
		100	DOB		
	:	110	DOC		
8-9	Flag Control Function	Specifies a p	ulse setting:		
		Bit Setting	Name		
		00	No Pulse (no effect on flags)		
		01	S Pulse		
		10	C Pulse		
		11	P Pulse		
10-15	Device Code		device that is to respond to the instruction. The primary for the Disk Subsystem drive is (24).		

The following sections describe sending a command to the controller, and retrieving asynchronous and synchronous interrupt information.

Sending a Command

Use the DOA, DOB, and DOC instructions to send a PIO command to the controller. These instructions transfer information from accumulators in the host to the command registers in the controller.

To send a PIO command to the controller, you must first load PIO command information into three host accumulators with an LDA instruction. You then transfer this information to the controller with the DOA, DOB, and DOC instructions:

- DOC transfers the command code and the return request bit (R) setting to command register C.
- DOA and DOB transfer the command's arguments, if any, to command registers A and B.

This transfer information is detailed later in this chapter. For now, note the following:

- The command code is an octal number that represents the command you want to transfer. For example, the command code for the Begin command is (002).
- The return request bit indicates whether you want the controller to interrupt the host after it completes a command (i.e., issue a synchronous interrupt).
- Command arguments are command dependent. If the argument is a double-word memory address, the DOA accumulator should contain the high-order word, and the DOB accumulator should contain the low-order word. Also, the high-order bit (bit 0 of DOA accumulator) should be 0 if you are referring to a physical memory address. It should be 1 if you are referring to a logical memory address.

Figure 3-2 shows sample code for sending a command to the controller. Figure 3-3 shows how this command information is transferred to the controller.

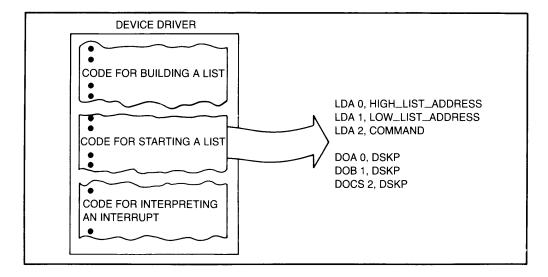


Figure 3-2. Sample Code: Sending a Command

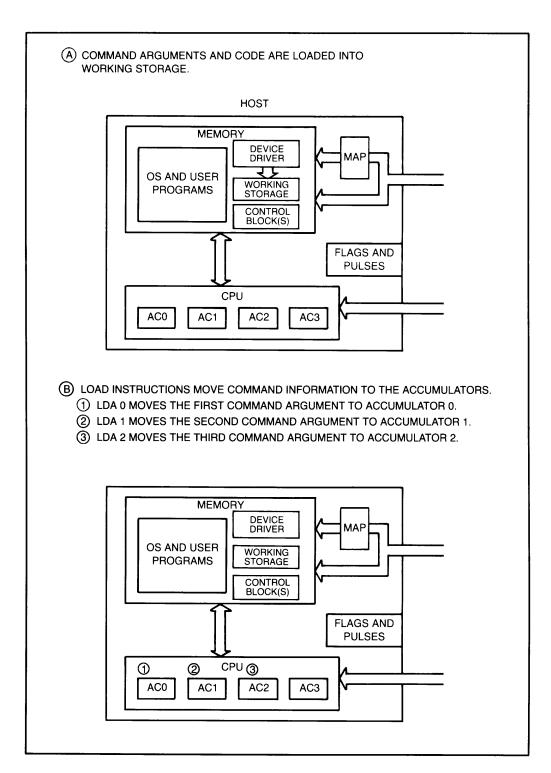


Figure 3-3. Action: Sending a Command

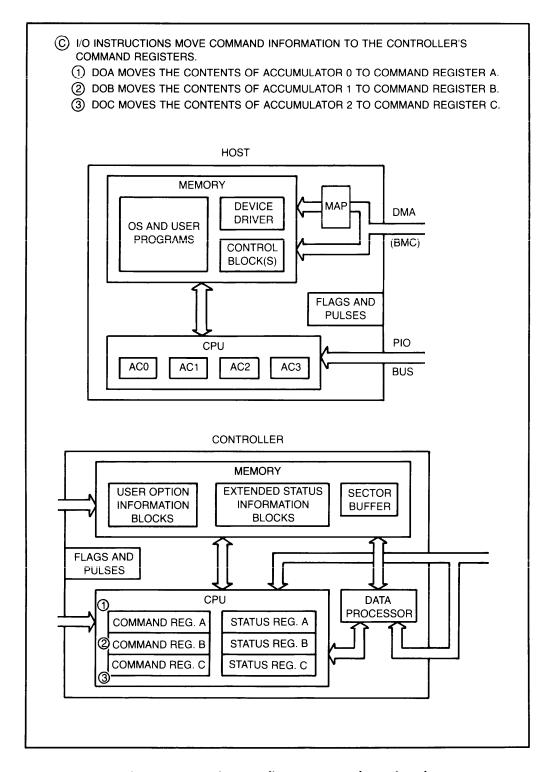


Figure 3-3. Action: Sending a Command (continued)

Table 3-1 shows what the accumulators should contain to issue a specific PIO command. The sections following Table 3-1 detail the individual command registers and instructions. Later sections detail PIO commands.

Table 3-1. Accumulator Contents

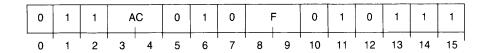
If you want to issue DOA accumulator this command: should contain:		DOB accumulator should contain:	DOC accumulator should contain:
Program Load (000)	Unit number	Unused	0000000000000000R
Begin (002) Unit number		Unused	000000000000010R
		Enter: (125252) Exit: 0	00000000010100R
Sysgen (025)	Unit Number	Unused	000000000010101R
Set Mapping Information (026)	Mapping options	Mapping options	00000000010110R
Get Mapping Information (027)	Unused	Unused	000000000010111R
Set Interface Information (030)	High-order address word of the interface information block	Low-order address word of the interface information block	000000000011000R
Get Interface Information (031)	High-order address word of the interface information block	Low-order address word of the interface information block	000000000011001R
Set Controller High-order addre word of the control information block		Low order address word of the controller information block	000000000011010R
Get Controller Information (033) High-order address word of the controller information block		Low order address word of the controller information block	000000000011011R
Set Unit Information (034) High-order address word of the unit information block		Low order address word of the unit infor- mation block	00000000011100R
Get Unit Information (035) High-order address word of the unit information block		Low order address word of the unit infor- mation block	00000000011101R
Get Extended Status 0 (040) High-order address word of the extended status information block for drive 0		Low-order address word of the extended status information block for drive 0	00000000100000R
Get Extended Status 1 (041)	l a		000000000100001R
Get Extended Status 2 (042) High-order address word of the extended status information block for drive 2		Low-order address word of the extended status information block for drive 2	00000000100010R

Table 3-1. Accumulator Contents (continued)

If you want to issue this command:	DOA accumulator should contain:	DOB accumulator should contain:	DOC accumulator should contain:
Get Extended Status 3 (043)			000000000100011R
Start List (100)	High-order address word of the CB list to execute	Low-order address word of the CB list to execute	000000001000000R
Start List (High Priority) (103)	High-order address word of the CB list to execute	Low-order address word of the CB list to execute	000000001000011R
Restart (116)	Unit number	Code Word	000000001001110R
Cancel List (123)	High-order address word of CB list to cancel	Low-order address word of CB list to cancel	000000001010011R
Unit Status (131)	Unit number	Unused	000000001011001R
Trespass (132)	Unit number	Unused	000000001011010R
Get List Status (133) High-order address word of the first CB in a list		Low-order address word of the first CB in a list	000000001011011R
Reset (777)	Unused	Unused	000000111111111R

DOA

DOA[f] ac,DSKP

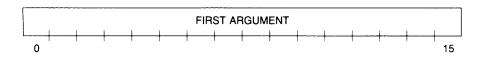


Use: Sends the first argument for a command to the controller. Before issuing this instruction, you must load the specified accumulator with the first command argument.

Effect: Loads the contents of the specified accumulator (ac) into the controller's command register A.

After the data transfer, the host sets the Busy and Done flags according to the function specified by f. The contents of the specified accumulator remain unchanged.

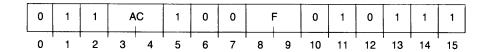
After the instruction executes, both the accumulator and command register A will contain:



Bits	Name	Contents or Function	
0-15	First Argument	Depends on the command being issued to the controller (see Table 3-1).	

DOB

DOB[f] ac,DSKP

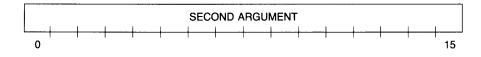


Use: Send the second argument for a command to the controller. Before issuing this instruction, you must load the specified accumulator with the second command argument.

Effect: Loads the contents of the specified accumulator (ac) into the controller's command register B.

After the data transfer, the host sets the Busy and Done flags according to the function specified by f. The contents of the specified accumulator remain unchanged.

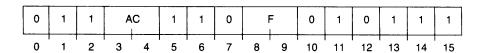
After the instruction executes, both the accumulator and command register B will contain the following:



I	Bits	Name	Contents or Function	
	0-15	Second Argument	Depends on the command being issued to the controller (see Table 3-1).	

DOC

DOC[f] ac,DSKP



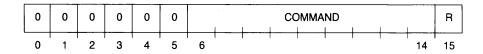
Use:

Send a command to the controller. Before issuing this instruction, load the specified accumulator with the command information: the octal code for the command and return request bit setting.

Effect: Loads the contents of the specified accumulator (ac) into the controller's command register C.

After the data transfer, the host sets the Busy and Done flags according to the function specified by f. The contents of the specified accumulator remain unchanged.

After the instruction executes, both the accumulator and command register C will contain the following:



Bits	Name	Contents or Function	
0-5	_	These bits must all be zero.	
6-14	Command	The octal con	nmand code to be transmitted to the controller:
		Octal Code	Command Name
		000	Program Load
		002	Begin
		025	Sysgen
		024	Diagnostic Mode (Enter/Exit)
		026	Set Mapping Information
		027	Get Mapping Information
		030	Set Interface Information
		031	Get Interface Information
		032	Set Controller Information
		033	Get Controller Information
		034	Set Unit Information
		035	Get Unit Information
		040	Get Extended Status 0
		041	Get Extended Status 1
		042	Get Extended Status 2
l i		043	Get Extended Status 3
		100	Start List
		103	Start List (High Priority)
		116	Restart
		123	Cancel List
		131	Unit Status
		132	Trespass
		133 777	Get List Status Reset
		///	uesei

Bits	Name	Contents or Function	
15	Return Request	If set: Except for the Reset command, the controller performs these tasks after a command executes:	
		 Loads the status registers (DIA, DIB, DIC) with the command's return information. 	
		Sets the Done flag when the Busy flag is cleared.	
		Not set: The controller does not load the status registers or set the Done flag.	

NOTES:

- The controller does not execute a PIO command until it receives an S pulse from the host.
- Setting the Done flag generates an interrupt request to the host.
- If a PIO command results in an error, the controller will first load error return information in the status registers. Then, it will set the Done flag—regardless of the return request bit setting.

Retrieving Interrupt Information

The DIA, DIB, and DIC instructions transfer the contents of the controller's status registers to specified host accumulators. You use these instructions to retrieve interrupt information from the controller. Figure 3-4 shows sample code for retrieving interrupt information. Figure 3-5 shows how this transfer takes place.

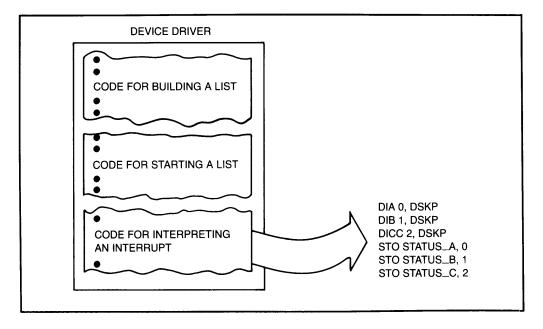


Figure 3-4. Sample Code: Retrieving Interrupt Information

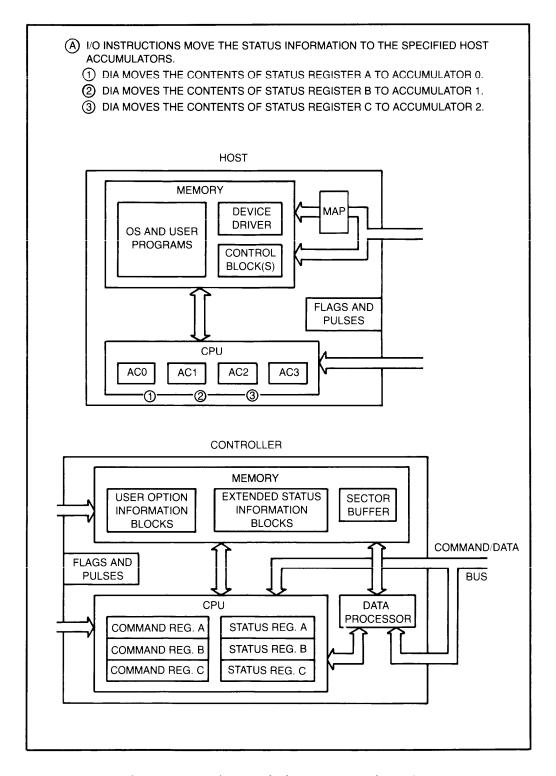


Figure 3-5. Action: Retrieving Interrupt Information

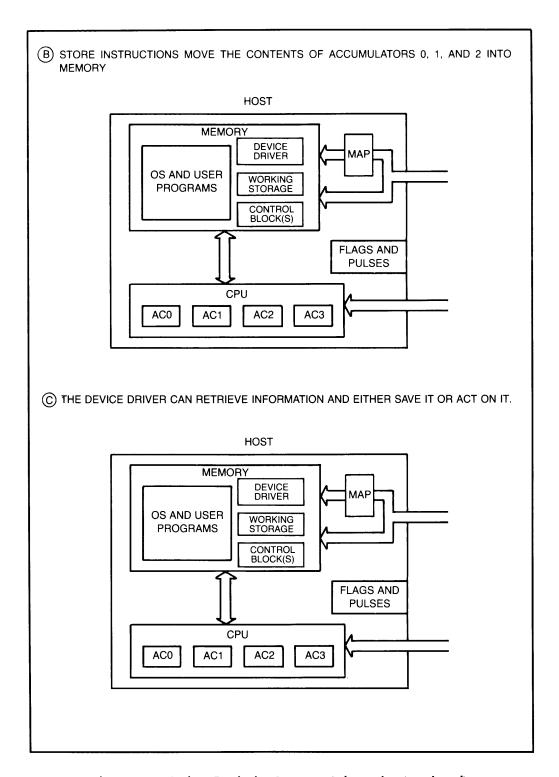


Figure 3-5. Action: Retrieving Interrupt Information (continued)

Use interrupt information to determine interrupt recovery procedures. After handling the interrupt, your interrupt routine must service it by issuing a C pulse.

DIC retrieves the contents of status register C. Bits 4-5 indicate what kind of interrupt occurred:

• Asynchronous: bits 4-5 = 00

• Synchronous: bits $4-5 \neq 00$

If an asynchronous interrupt occurs, bits 6-15 of status register C contain a code that represents what kind of error occurred. If a synchronous interrupt occurs, these bits echo the command code and return request bit setting of the most recent command sent to the controller.

Bits 0-3 of status register C are valid whenever the Busy flag is clear. The remaining bits are valid whenever the Done flag is set.

DIA and DIB retrieve additional return information from status registers A and B. When these registers contain a double-word operand, register A contains the high-order word and B contains the low-order word. The contents of these registers are valid when the Done flag is set.

Asynchronous Interrupts

Table 3-2 lists each asynchronous interrupt and the contents of the status registers when it occurs. Chapter 4 describes how to recover from these interrupts.

Table 3-2. Status Register Contents: Asynchronous Interrupts

If bits 6-15 of status register C contain this octal code:	Then this interrupt occurred:	And, the CB or CB list status is:	And, registers A and B will contain:
000	Null Interrupt	-	Unused
001	Controller Panic	_	Stack pointer and program counter
002	CB Execution Error: Soft Errors	List complete	Double-word address of the first CB in the list
003	CB Execution Error: Hard Errors	List terminated	Double-word address of first CB in the list
004	I Bit Set	CB complete	Double-word address of the interrupting CB
005	No errors	List complete	Double-word address of the first CB in the list
006	CB Termination Error: Cancel List	List terminated	Double-word address of the first CB in the list

Table 3-2. Status Register Contents: Asynchronous Interrupts (continued)

If bits 6-15 of status register C contain this octal code:	Then this interrupt occurred:	And, the CB or CB list status is:	And, registers A and B will contain:
007	Soft Error: S Bit Set	CB not complete	Double-word address of the first CB in the list
010	CB Interpretation Error: Status Word Not 0	List terminated; status word not written	Double-word address of the first CB in the list
011	CB Interpretation Error: Illegal CB Command	List terminated	Double-word address of the first CB in the list
012	CB Interpretation Error: CB Range Error	List terminated	Double-word address of the first CB in the list
013	CB Interpretation Error: Illegal Unit Number	List terminated	Double-word address of the first CB in the list
014	CB Interpretation Error: Illegal Link Address	List terminated	Double-word address of the first CB in the list
015	CB Interpretation Error: Illegal Page Number List Address	List terminated	Double-word address of the first CB in the list
016	CB Interpretation Error: Illegal Transfer Address	List terminated	Double-word address of the first CB in the list
017	CB Interpretation Error: Illegal Transfer Count	List terminated	Double-word address of the first CB in the list
020	Unreadable CB	List terminated; status word not written	Double-word address of the first CB in the list
021	Unwritable CB	List terminated; status word may not have been written	Double-word address of the first CB in the list
077	Power Fail	List terminated	Double-word address of the first CB in the list.

NOTES:

1. Most asynchronous return addresses point to the first CB in the list. This means you must scan an interrupting CB list to find failing CBs or to track soft errors and retries. You can improve performance if you scan lists only when abnormal interrupts occur, or when you have programmed the controller to interrupt on each CB (code 4).

- 2. The controller retries only disk subsystem errors: codes 2, 3, and 7.
- 3. Interpretation errors indicate system errors, and must be handled by the operating system. You must scan a list that generates an interpretation error.
- 4. The Restart command is valid only for codes 3 and 7.

Synchronous Interrupts

Table 3-3 shows what the status registers contain after you issue a PIO command. Usually, the first six bits of status register C depend on the just executed command and whether an error occurred. Therefore, these bits are not shown in this table. The exception is the Reset command, which writes 0001000000000000 in status register C after the on-board diagnostics are completed.

Table 3-3. Status Register Contents: Synchronous Interrupts

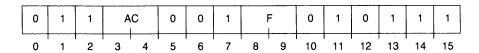
If you issued this command:	Status Register A will contain:	Status Register B will contain:	Status register C (bits 6-15) will contain:
Program Load (000)	Unused	Unused unless a PIO command execution error occurs	00000000R
Begin (002)	Unused unless a PIO command execution error occurs	Unused unless a PIO command execution error occurs	00000010R
Diagnostic Mode (024) (Enter/Exit)	Unused	Unused	000010100R
Sysgen (025)	Unused unless a PIO command execution error occurs	Unused unless a PIO command execution error occurs	000010101R
Set Mapping Information (026)	Mapping options	Mapping options	000010110R
Get Mapping Information (027)	Mapping options	Mapping options	000010111R
Set Interface Information (030)	Unused	Unused unless a PIO command execution error occurs	000011000R
Get Interface Information (031)	Unused	Unused unless a PIO command execution error occurs	000011001R
Set Controller Information (032)	Unused	Unused unless a PIO command execution error occurs	000011010R
Get Controller Information (033)	Unused	Unused unless a PIO command execution error occurs	000011011R
Set Unit Information (034)	Unused	Unused unless a PIO command execution error occurs	000011100R

Table 3-3. Status Register Contents: Synchronous Interrupts (continued)

If you issued this command:	Status Register A will contain:	Status Register B will contain:	Status register C (bits 6-15) will contain:
Get Unit Information (035)	Unused	Unused unless a PIO command execution error occurs	000011101R
Get Extended Status 0 (040)	Unused	Unused unless a PIO command execution error occurs	000100000R
Get Extended Status 1 (041)	Unused	Unused unless a PIO command execution error occurs	000100001R
Get Extended Status 2 (042)	Unused	Unused unless a PIO command execution error occurs	000100010R
Get Extended Status 3 (043)	Unused	Unused unless a PIO command execution error occurs	000100011R
Start List (100)	High-order address word of first CB in started list	Low-order address word of first CB in started list	001000000R
Start List (High Priority) (103)	High-order address word of first CB in started list	Low-order address word of first CB in started list	001000011R
Restart (116)	Unit number	Unused	001001110R
Cancel List (123)	High-order word address of the active CB in the terminated list; 0 if list not active	Low-order word address of the active CB in the terminated list; 0 if list not active	001010011R
Unit Status (131)	Unused	Unit status word	001011001R
Trespass (132)	Unit Number	Unused	001011010R
Get List Status (133)	List status	Number of current CB	001011011R
Reset (777)	Unused	Internal diagnostic re- sults (zero if no errors)	0000000000

DIA

DIA[f] ac,DSKP

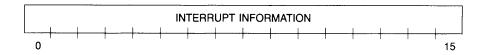


Use: Retrieve interrupt information recorded in status register A.

Effect: Loads the contents of the controller's status register A into the specified host accumulator (ac). The register contents are valid only when the Done flag is set.

After the data transfer, the host sets the Busy and Done flags according to the function specified by f. The previous contents of the specified accumulator are lost.

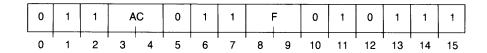
After the instruction executes, both the accumulator and status register A contain the following:



Bits	Name	Contents or Function
0-15	Status Register A	Synchronous: Depends on the command that was issued to the control-
		ler (see Table 3-3).

DIB

DIB[f] ac, DSKP

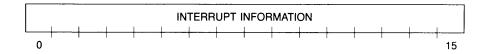


Use: Retrieve interrupt information recorded in status register B.

Effect: Loads the contents of the controller's status register B into the specified host accumulator (ac). The register contents are valid only when the Done flag is set.

After the data transfer, the host sets the Busy and Done flags according to the function specified by f. The previous contents of the specified accumulator are lost.

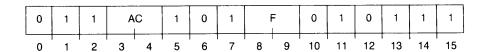
After the instruction executes, both the accumulator and status register B contain the following:



Bits	Name	Contents or Function
0-15	Status Register B	Asynchronous: Depends on the asynchronous interrupt code recorded in bits 6-15 of status register C (see Table 3-2).
		Synchronous: Depends on the command that was issued to the controller (see Table 3-3).

DIC

DIC[f] ac, DSKP



Use: Retrieve interrupt information recorded in status register C.

Effect: Loads the contents of the controller's status register C into the specified host accumulator (ac). The register contents are valid only when the Done flag is set.

After the data transfer, the host sets the Busy and Done flags according to the function specified by f. The previous contents of the specified accumulator are lost.

After the instruction executes, both the accumulator and status register A will contain the following:



Bits	Name	Contents or Function	
0-3	Status	The controller execution state (see Chapter 4).	
4-5	ccs	The controller completion status:	
		Bit Setting Meaning 00 Asynchronous interrupt occurred 01 PIO illegal command error: invalid command code 10 PIO command execution error: unsuccessful command execution 11 PIO command completed: successful command execution	
6-15	Command Echo or Interrupt Code	Synchronous: Bits 6-14 echo the last command sent to the controller. Bit 15 echoes the return request bit setting. Asynchronous: The type of asynchronous interrupt (see Table 3-2).	

PIO COMMANDS

PIO commands fall into these groups:

- Control and initialization commands Override initial microcode values.
- Diagnostic commands Perform diagnostic tests on the controller.
- Extended status commands Examine the state of the disk subsystem when an error occurs.
- List control commands Perform CB operations.
- State independent commands Invoke the diagnostic command set or reset the controller.
- Valid while reset commands: Move initial microcode values into host memory and boot the controller's processor.

Send PIO commands and arguments to the controller with the DOA, DOB, and DOC instructions. You can use these instructions to load the controller's command registers in any order. Command execution does not begin until you issue an S pulse.

The following sections detail all but the diagnostic commands. (Diagnostic commands are used primarily by Data General field engineers and are described only in internal documents).

All PIO commands are described in the following format:

- 1. The command name with its octal code in parentheses followed by a brief command description.
- 2. A coding information table containing detailed command and status register information. The "Sending the Command" column lists the information you need to transfer from the host to the command registers. The "Return Information" column lists the information returned after normal (no error) execution.
- 3. A detailed command description.
- 4. Error conditions, if applicable.

Standard Return Format

The standard return is a synchronous interrupt. The following command descriptions refer to a "standard return format" for status register C. This format is as follows:

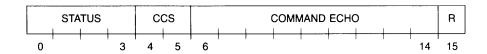


Figure 3-6. Standard Return Format

Bits	Name	Contents or Function
0-3	Status	Command execution status (i.e., the current state of the controller).
4-5	ccs	Command completion status (i.e., invalid command code, successful or unsuccessful command execution).
6-14	Command Echo	Echoes the last command sent to the controller.
15	R	Echoes the return request bit sent to the controller.

Control and Initialization Commands

Table 3-4 lists the control and initialization commands. You use these commands to transfer user options between the host and the controller. User options include mapping, interface, controller, and unit options. For each type of option, PIO commands let you assign or retrieve current options. A Get command returns any information previously set by a Set command.

You use PIO registers to specify mapping option values; you use information blocks to specify the other user options.

For all Get and Set commands, command errors are reported as PIO command execution errors. If an error occurs, status register B may indicate an ending memory address error (bit 7 set) or a BMC error (bit 12 set). Chapter 4 describes errors in detail.

Table 3-4. PIO Control and Initialization Commands

Octal Code	Command Name	Function
025	Sysgen	Sets default initialization parameters.
026	Set Mapping Information	Sets mapping options.
027	Get Mapping Information	Retrieves mapping options.
030	Set Interface Information	Sets interface options.
031	Get Interface Information	Retrieves interface options.
032	Set Controller Information	Sets controller options.
033	Get Controller Information	Retrieves controller options.
034	Set Unit Information	Sets unit options.
035	Get Unit Information	Retrieves unit options.
131	Unit Status	Retrieves the unit status word.

Sysgen (025)

Sets up the current initialization parameters as default parameters.

Sending the Command		Return Information	
Command Register	Content	Status Register	Content
A	Unit number	А	Unused unless a PIO command ex- ecution error occurs
В	Unused	В	Unused unless a PIO command ex- ecution error occurs
С	000000000010101R	С	Standard return format

Detailed command description

This command instructs the controller to set up its current initialization parameters as default parameters. Subsequent power-ups or PIO Resets, followed by Begin, will automatically bring in these default parameters. Therefore, the Get and Set commands for the mapping, the interface, the controller, and the unit information command sequences can be bypassed on, for example, power fail recovery.

Get Controller Information (033)

Retrieves controller options.

Sending the Command		Return Information	
Command Register	Content	Status Register	Content
А	High-order address word of the controller information block	Α	Unused
В	Low-order address word of the controller information block	В	Unused unless a PIO command execution error occurs
С	000000000011011R	С	Standard return format

Detailed command description

This command retrieves the controller information block from the controller and stores it at the specified address in host memory. The values stored in this information block specify controller options. Once you retrieve the block, you can examine or change controller options using offsets from the host address.

You use this command with the Set Controller Information (032) command. For example, to change controller options, you issue a Get command, load the new value into the appropriate word portion of the block, then issue a Set command.

Get Interface Information (031)

Retrieves interface options.

Sending the Command		Return Information	
Command Register	Content	Status Register	Content
Α	High-order address word of the interface information block	Α	Unused
В	Low-order address word of the interface information block	В	Unused unless a PIO command execution error occurs
С	00000000011001R	С	Standard return format

Detailed command description

This command retrieves the interface information block from the controller and stores it at the specified address in host memory. The values stored in this information block specify interface options. Once you retrieve the block, you can examine or change interface options using offsets from the host address.

You use this command with the Set Interface Information (030) command. For example, to change interface options: issue a Get command, load the new value into the appropriate word portion of the block, then issue a Set command.

Get Mapping Information (027)

Retrieves mapping options.

Sending the Command		Return Information	
Command Register	Content	Status Register	Content
А	Unused	Α	Mapping options (see note)
В	Unused	В	Mapping options (see note)
С	000000000010111R	С	Standard return format

Detailed command description

This command retrieves current mapping options. You use it with the Set Mapping Information (026) command which lets you change mapping options.

After a power up, IORST, or Reset, followed by a Begin command, you must issue a Get command to retrieve available mapping options. The section below lists the mapping options returned in status registers A and B. You must select, at most, one of the available options (see Chapter 2 for details). You use the Set command to inform the device of the selected options and map slots. Subsequent Get commands return the selected mapping option, first map slot, and number of map slots selected.

NOTE: The PIO registers contain no device-specific information. Instead, they contain the following mapping options:

Register	Bits	Meaning If Set:
А	0	DMA over the Data Channel (not supported)
	1	DMA over the BMC
	2-5	Unused
	6-15	First Map Slot Assigned
В	0	Mapping with Map Slot Load Interrupts
	1	Mapping with Internal BMC Physical Remapping
	2	Mapping with Upstream Loading
	3	Upstream Loading with Host Page Tables
	4-7	Unused
	8-15	Number of Map Slots Assigned

Get Unit Information (035)

Retrieves unit options.

	Sending the Command		Return Information	
Command Register	Content	Status Register	Content	
A	High-order address word of the unit information block	Α	Unused	
В	Low-order address word of the unit information block	В	Unused unless a PIO command execution error occurs	
С	000000000011101R	С	Standard return format	

Detailed command description

This command retrieves the unit information block from the controller and stores it at the specified address in host memory. The values stored in this information block specify unit options. Once you retrieve the block, you can examine or change unit options using offsets from the host address.

You use this command with the Set Unit Information (032) command. For example, to change unit options: issue a Get command, load the new value into the appropriate word portion of the block, then issue a Set command.

Error conditions

The correct unit number must appear in the unit information block or the controller will generate an error.

Set Controller Information (032)

Sets controller options.

Sending the Command			Return Information	
Command Register	Content	Status Register	Content	
А	High-order address word of the controller information block	Α	Unused	
В	Low-order address word of the controller information block	В	Unused unless a PIO command execution error occurs	
С	000000000011010R	С	Standard return format	

Detailed command description

This command assigns new values to the controller information block. The controller information block specifies controller options. Each option is offset from the address of the controller information block.

You use this command with the Get Controller Information (033) command. For example, to change controller options issue a Get command, load the new value into the appropriate word portion, then issue a Set command.

Error conditions

This command is valid only when the controller is idle (has no active CBs).

Set Interface Information (030)

Sets interface options.

	Sending the Command		Return Information	
Command Register	Content	Status Register	Content	
А	High-order address word of the interface information block	Α	Unused	
В	Low-order address word of the interface information block	В	Unused unless a PIO command execution error occurs	
С	00000000011000R	С	Standard return format	

Detailed command description

This command assigns new values to the interface information block. The interface information block specifies interface options. Each option is offset from the address of the interface information block.

You use this command with the Get Interface Information (033) command. For example, to change interface options: issue a Get command, load the new value into the appropriate word portion, then issue a Set command.

Error conditions

This command is valid only when the controller is idle (has no active CBs).

Set Mapping Information (026)

Sets mapping options.

Sending the Command		Return Information	
Command Register	Content	Status Register	Content
Α	Mapping options	Α	Same as command register A
В	Mapping options	В	Same as command register B
С	000000000010110R	С	Standard return format

Detailed command description

This command lets you set new mapping options. Mapping options are not available to the controller before you issue this command. Therefore, you must issue this command before you can issue any address-specifying PIO command.

You use this command with the Get Mapping Information (027) command. The Set command lets you specify mapping options; the Get command lets you retrieve current mapping information.

The Get Mapping Information command description and Chapter 2 describe mapping options.

Error conditions

This command is valid only when the controller is idle (has no active CBs). You may set only one of bits 1 - 3 in DIB.

Set Unit Information (034)

Sets unit options.

Sending the Command		Return Information	
Command Register	Content	Status Register	Content
А	High-order address word of the unit information block	Α	Unused
В	Low-order address word of the unit information block	В	Unused unless a PIO command execution error occurs
С	00000000011100R	С	Standard return format

Detailed command description

This command assigns new values to the unit information block. The unit information block specifies unit options. Each option is offset from the address of the unit information block.

You use this command with the Get Unit Information (033) command. For example, to change unit options: issue a Get command, load the new value into the appropriate word portion, then issue a Set command.

Error conditions

The correct unit number must appear in the unit information block or the controller will generate an error.

This command is valid only when the drive is idle (has no active CBs).

Unit Status (131)

Retrieves the unit status word.

Sending the Command		Return Information	
Command Register	Content	Status Register	Content
Α	Unit number	Α	Unused
В	Unused	В	Unit status word
С	000000001011001R	С	Standard return format

Detailed command description

This command retrieves the unit status word for the specified unit.

To check a unit's status, transfer its unit number to command register A. Then issue the command. The controller returns the unit status word for the specified unit in status register B (see Chapter 4).

Extended Status Commands

Table 3-5 lists the extended status commands. When a hard error occurs, the controller freezes the Disk Subsystem's state. You can then use these commands to check the subsystem's state. Each command retrieves the extended status information block for one of the controller's drives. Extended status information blocks are described in Chapter 2; errors are described in Chapter 4.

Table 3-5. PIO Extended Status Commands

Octal Code	Command Name	Function
040	Get Extended Status 0	Retrieves extended status for drive 0.
041	Get Extended Status 1	Fletrieves extended status for drive 1.
042	Get Extended Status 2	Retrieves extended status for drive 2.
043	Get Extended Status 3	Retrieves extended status for drive 3.

These commands are valid only if two conditions are met:

- 1. A CB execution error has occurred during a drive operation.
- 2. You have not yet issued a Restart command for the drive.

When you issue an extended status command, the controller moves a copy of the extended status block for the specified drive into host memory. The block reflects the drive's status information at the time the last error occurred. You can use the address returned by the command as an offset to determine error conditions.

You can issue an extended status command for a soft CB execution error only if the S bit in the unit information block was set at the time of the error. If the S bit was set, the controller generates an extended status block for each error.

You must issue a Restart to begin execution again. A Restart command clears the CB execution error and resumes CB execution. An extended status block is not available after you issue a Restart command.

The following sections summarize the transfer information for these commands.

Get Extended Status 0 (040)

Retrieves extended status for drive 0.

Sending the Command		Return Information	
Command Register	Content	Status Register	Content
Α	High-order address word of the extended status information block for drive 0	А	Unused
В	Low-order address word of the extended status information block for drive 0	В	Unused unless a PIO command execution error occurs
С	00000000100000R	С	Standard return format

Get Extended Status 1 (041)

Retrieves extended status for drive 1.

Sending the Command		Return Information	
Command Register	Content	Status Register	Content
Α	High-order address word of the extended status information block for drive 1	Α	Unused
В	Low-order address word of the extended status information block for drive 1	В	Unused unless a PIO command execution error occurs
С	00000000100001R	С	Standard return format

Get Extended Status 2 (042)

Retrieves extended status for drive 2.

Sending the Command		Return Information	
Command Register	Content	Status Register	Content
A	High-order address word of the extended status information block for drive 2	А	Unused
В	Low-order address word of the extended status information block for drive 2	В	Unused unless a PIO command execution error occurs
С	00000000100010R	С	Standard return format

Get Extended Status 3 (043)

Retrieves extended status for drive 3.

Sending the Command		Return Information	
Command Register	Content	Status Register	Content
А	High-order address word of the extended status information block for drive 3	A	Unused
В	Low-order address word of the extended status information block for drive 3	В	Unused unless a PIO command execution error occurs
С	000000000100011R	С	Standard return format

List Control Commands

Table 3-6 lists the PIO list control commands. You use these commands to perform CB list operations.

Table 3-6. PIO List Control Commands

Octal Code	Command Name	Function
123	Cancel List	Cancels the specified CB list.
133	Get List Status	Retrieves status information about the currently executing CB list.
116	Restart	Restarts or terminates CB execution after an error.
100	Start List	Starts CB list execution.
103	Start List (High Priority)	Starts high priority CB list execution.

Cancel List (123)

Cancels the specified CB list.

Sending the Command		Return Information	
Command Register	Content	Status Register	Content
А	High-order address word of the CB list to cancel	Α	High-order word address of the active CB in the terminated list; 0 if list not active
В	Low-order address word of the CB list to cancel	В	Low-order word address of the active CB in the terminated list; 0 if list not active
С	000000001010011R	С	Standard return format

Detailed command description

This command tells the controller not to execute the specified CB list.

If the controller is currently executing the specified list, it brings execution to an orderly halt:

- 1. It sets the CB terminated bit in the CB status word.
- 2. It generates a CB Termination Error: Cancel List (006) asynchronous interrupt.

If the controller is not executing the list, this command cancels the previous Start List command for the list, and the list is not executed.

If the controller has finished executing the list but the host has not acknowledged the asynchronous interrupt, the controller will reissue the interrupt as if it had not received the Cancel List command. It will not issue a Cancel List interrupt since it has already written the CB status word.

Error conditions

The list address must be identical to the address specified for the Start List command that began the list.

Get List Status (133)

Retrieves status information about the currently executing CB list.

Sending the Command		Return Information	
Command Register	Content	Status Register	Content
A	High-order address word of the first CB in the list	Α	List status information
В	Low-order address word of the first CB in the list	В	Number of the CB currently executing in the list
С	000000001011011R	С	Standard return format

Detailed command description

This command lets you check a list's status. The list may or may not be executing. If the list is not active, the CB Done bit may be set for every CB in the list, but the interrupt signalling that the entire list is completed may not have been generated.

List status information is returned in status register A:

Bits	Meaning If Set:
0	List is Active
1	Controller in Error Recovery
2	Waiting for a Restart
3	Timeouts have occurred
4-15	Unused

This command is useful if the controller has been running for a long time without interrupting the host (by executing long lists or extensive retries).

Error conditions

This command is legal only after a Set Mapping Information command has been issued. Otherwise, a PIO illegal command error occurs.

The list address must be identical to the address specified for the Start List command that began the list.

Restart (116)

Restarts or terminates CB execution after an error.

Sending the Command		Return Information	
Command Register			Content
Α	Unit number to restart	Α	Restarted unit number
В	Code word	В	Unused
С	00000001001110R	С	Standard return format

Detailed command description

This command clears the error state of the specified unit, and resumes CB execution according to the specified code word.

Restart is valid only if the controller generated one of these asynchronous interrupts for the specified unit:

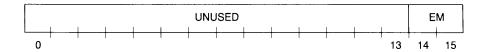
- Hard CB Execution Error (asynchronous interrupt code 3)
- Soft CB Execution Error; S bit set in the unit information block (asynchronous interrupt code 7)

The controller stops executing any lists accessing the unit until it receives a Restart command. Therefore, you must issue the Restart command before operations can continue on the interrupted unit.

Extended Status is available only between the interrupt and Restart.

Code word

The code word has the following format:



Bit	Contents or Function		
0-13	Unused		
14-15	Execution Mo	ode code:	
	Bit Setting	Meaning	
	00	00 Retry the CB that was in error	
	01 Abandon the CB that was in error		
	Abandon the entire list containing the CB that was in error		
	11	Illegal	

Error conditions

If you issue this command and an asynchronous interrupt 3 or 7 has not occurred, the controller returns a PIO illegal command error.

If the unit is not in error state when you issue this command, an error results.

If you specify 11 for the execution mode, a PIO command execution error will result.

Start List (100)

Starts CB list execution.

Sending the Command		Return Information	
Command Register	Content	Status Register	Content
А	High-order address word of the first CB in the list	А	Same as command register A
В	Low-order address word of the first CB in the list	В	Same as command register B
С	000000001000000R	С	Standard return format

Detailed command description

This command starts executing a CB list, beginning with the specified address. The controller executes the CBs in the list sequentially.

There are actually two separate Start List commands: (100) and (103). Start List (100) assigns the lowest priority, and Start List (103) assigns the highest. That is, lists started with a Start List (103) command will be executed before lists started with a Start List (100) command. The two low-order bits of the command code indicate the priority assigned to a list.

The controller may return from this command as soon as it has verified that the specified CB address is legal; it does not need to fetch the first CB before returning.

Error conditions

If the CB address is illegal (outside the controller's fetching range), a PIO command execution error results. (A mapped address is outside the controller's fetching range if bits 1-11 of the high-order address word are not zero. An unmapped address is outside the controller's fetching range if bits 1-10 of the high-order address word are not zero — unless upstream map loading has been selected in a Set Mapping Information command.)

Start List — High Priority (103)

Starts CB list execution (higher priority than Start List).

Sending the Command			Return Information
Command Register	Content	Status Register	Content
А	High-order address word of the CB list	Α	Same as command register A
В	Low-order address word of the CB list	В	Same as command register B
С	000000001000011R	С	Standard return format

Detailed command description

This command is the same as the Start List (100) command, but it assigns the highest priority to a list. All high priority CBs for a unit will execute before any lower priority CBs.

State Independent Commands

Table 3-7 lists the state independent commands. After an error, these commands let you reset the controller or implement the diagnostic command set. These commands are state independent; that is, you may issue any of the commands no matter what the controller error state is.

Table 3-7. PIO State independent Commands

Octal Code Command Name Function			
024	Diagnostic Mode	e Turns diagnostic mode on or off.	
777	Reset	Resets the controller hardware.	

After the controller completes either of these commands, it is in reset state. Therefore, you must reinitialize the controller with a Begin command.

Diagnostic Mode (024)

Turns Diagnostic mode on or off.

Sending the Command		Return Information	
Command Register	Content	Status Register	Content
А	Enter: (125252) Exit: 0	Α	Unused
В	Enter: (125252) Exit: 0	В	Unused
С	000000000010100R	С	Standard return format

Detailed command description

This command turns Diagnostic mode on and off. When Diagnostic mode is on, the controller recognizes the diagnostic command set. When it is off, those commands result in unimplemented PIO command errors or illegal CB operation code errors.

To turn Diagnostic mode on, DOA and DOB must each contain the alternating bit pattern (125252). To turn it off, DOA and DOB must both be set to 0.

Attempting to turn Diagnostic mode on when it is already on, or off when it is already off, has no effect.

Diagnostic commands are used primarily by Data General Field Engineers and as such are described only in internal documents.

Error conditions

If DOA and DOB are not correctly encoded, a PIO command execution error results.

Reset (777)

Resets the controller hardware.

Sending the Command		Return Information	
Command Register	Content	Status Register	Content
Α	Unused	Α	Unused
В	Unused	В	Results of internal diagnostics
С	000000111111111R	С	000100000000000

Detailed command description

This command resets the controller hardware. It is always legal. During a Reset command, the controller resets the Busy and Done flags, clears all interrupts, and executes internal diagnostics.

If an internal diagnostic error occurs, the controller puts the results into status register B. Each bit corresponds to a diagnostic test:

Bit	Meaning If Set		
0	Disk Interface FIFO's test		
1	Sector Buffer and FIFO's test		
2	Dirty Bit Ram test		
3	Error Flops test		
4	ECC/CRC test		
5	Sequencer/ALU test		
6	Microcode PROM test		
7	Control Tag Bus test		
8	Command Data Bus test		
9	Command Registers test		
10	Burst Length Compare test		
11	Transfer Counter test		
12	Address Counter test		
13	Interrupts test		
14	RAM test		
15	PROM test		

The controller resets bits 0-3 of status register C. When the command completes, with or without error, the controller sets these bits to 0001 to indicate that the reset sequence is also complete.

You can issue a Program Load, Begin, or Unit Status command immediately after a Reset command. If a Reset command fails, the internal diagnostic results remain in status register B, and the controller generates a PIO illegal command error (sets bits 4-5 of status register C to 01). You must reissue the Reset command before you issue your next command. If the Reset command is successful, the controller executes the next command.

NOTES:

- 1. The controller "forgets" everything it was doing when it receives a Reset command. Therefore, you must issue a Begin command to reinitialize the controller.
- 2. Power on Reset is synonymous with the Reset command. Therefore, if power is going down, your routine should try to save uncompleted CB list addresses. You can then restart these lists when power returns.
- 3. IORST is also synonymous with the Reset command except that it does not perform some internal diagnostics.
- 4. The controller will not generate an interrupt at the completion of a Reset command, regardless of the state of the return request bit (R).

Valid While Reset Commands

Table 3-8 lists the commands that are valid after a reset sequence.

Table 3-8. PIO Valid While Reset Commands

Octal Code	Command Name	Function
002	Begin	Causes the controller to boot itself.
024	Diagnostic Mode	Turns diagnostic mode on.
000	Program Load	Boots loader into host memory.
131	Unit Status	Retrieves a unit's status word.
132	Trespass	Issues hardware trespass to specified unit.
777	Reset	Resets the controller's hardware.

When a Reset command completes, bits 0-3 of status register C equal 0001. If status register B does not contain zero, an internal diagnostic error occurred, and the contents of status register B indicate the type of error.

If an error occurs, legal commands are Diagnostic Mode and Reset. If no error occurs, all of the commands listed in Table 3-8 are legal. If you issue any other command, the controller will generate a PIO illegal-command error (that is, bits 4-5 of status register C are set equal to 01).

The Diagnostic Mode, Unit Status, and Reset commands were described earlier; the sections below describe the Begin and Program Load commands.

Begin (002)

Causes the controller to boot itself.

Sending the Command		Return Information	
Command Register	Content	Status Register	Content
Α	Unit number	Α	Unused unless a PIO command ex- ecution error occurs
В	Unused	В	Unused unless a PIO command ex- ecution error occurs
С	000000000000010R	С	Standard return format

Detailed command description

This command causes the controller to boot its programs from the specified drive. Controller programs reside on a private area of the master disk. The code is replicated 16 times within this area for reliability.

A Begin command is valid only after a successful reset sequence (i.e., status register C bits 0-3 = 0001).

If the Begin command completes successfully, status register C is set to the initial state: bits 0-3 = 1000. Also, if the return request bit is set, bits 4-5 will equal 11 to reflect a good return via the interrupt request.

If the Begin command fails, the controller generates a PIO command execution error interrupt and sets bits 4-5 to 10. Bits 0-3 remain 0001.

Error conditions

If a PIO command execution error results, the controller interrupts and writes the unit status word into status register A and additional status information into status register B (see "Begin and Program Load" in Chapter 4).

Program Load (000)

Boots 256 words from drive 0, address 0 into host memory.

Sending the Command		Return Information	
Command Register	Content	Status Register	Content
A	Unit number	А	Unused unless a PIO command execution error occurs
В	Unused	В	Unused unless a PIO command ex- ecution error occurs
С	000000000000000R	С	Standard return format

Detailed command description

This command boots 256 words from device address 0 of the specified drive into the lowest portion of host memory. These words contain the information you need to boot your system from disk.

The command code for a Program Load is all zeroes, so that issuing an NIOS after an IORST always results in a program load from unit 0.

Error conditions

If a PIO command execution error results, the controller interrupts and writes the CB unit status word into status register A and the CB error status word into status register B (see "Begin and Program Load" in Chapter 4).

Trespass (132)

Issues a hardware trespass to the specified unit.

Sending the Command		Return Information	
Command Register			Content
A	Unit number	Α	Same as command register A
В	Unused	В	Same as command register B
С	000000001011010R	С	Standard return format

Detailed command description

This command causes the controller to issue a hardware trespass to the specified unit. When the disk reaches the first sector mark, it interrupts the controller. The trespassed controller releases the port, and the trespassing controller reserves it. The trespassing controller then starts (or continues) executing any lists queued for the reserved port.

The controller uses an algorithm for reserving and releasing units that ensures that a unit will be available eventually (see "Dual Port Operation" in Chapter 2). Therefore, you will not need to use the Trespass command very often. You should issue the Trespass command only when disk access is needed immediately, or when it seems that the other port has failed and will not release the unit.

NOTE: The trespassing host should inform the trespassed host that it is issuing a Trespass command. The trespassed host can then cancel any lists for the specified unit.

CB COMMANDS

CB commands are disk commands that can be divided into these groups:

General commands — Perform a null disk operation or move the heads.

- Read commands Perform disk read operations.
- Write commands Perform disk write operations.

To perform a disk operation, you:

- 1. Load bits 6-15 of CB word 2 with a disk command operation code.
- 2. Load bits 8-15 of CB word 9 with the unit number.
- 3. For read and write operations, load these CB fields:
 - The page number list address
 - The transfer address
 - The device address
 - The transfer count
- 4. Issue the appropriate PIO command (via an instruction) to execute the control block.

The following sections detail these commands. Each description depicts the content of CB word 2.

General Commands

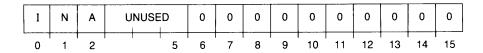
Table 3-9 lists the general CB commands. These commands do not affect disk operations.

Table 3-9. General CB Commands

Octal Code	Command Name	Function
000	No Operation	Performs a null operation on the specified unit.
400	Recalibrate Disk	Begins the recalibrate disk procedure.

No Operation (000)

Performs a null operation on the specified unit.



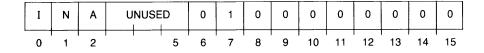
Detailed command description

This command performs a null operation on the specified unit. It does not affect the drive or any data on the disk.

To implement this command, create an all-zero CB, and specify the unit number.

Recalibrate Disk (400)

Begins the recalibrate disk procedure.



Detailed command description

This command begins the recalibrate procedure for the specified unit. After the procedure completes, the heads are positioned at the recalibrate cylinder.

You can also implement this command if you create an all-zero CB except for the unit number and (400) in the the operation code field.

Read Commands

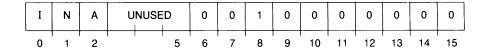
Table 3-10 lists the CB commands that let you perform different disk read operations.

Table 3-10. CB Read Commands

Octal Code	Command Name	Function
200	Read	Places data from the disk into host memory.
201	Read/Verify	Compares data from the disk with data in host memory.
205	Read/Verify Single Word	Compares a single data word to each data word in a sector.
210	Read Raw Data	Reads the header, data, and check words (checksum and ECC) of a sector into host memory.
220	Read Headers	Reads disk sector header into host memory.
242	Read with Modified Bit Map	Reads a 256-word modified bit map and up to 4096 sectors of contiguous data into host memory.

Read (200)

Places data from the disk into host memory.

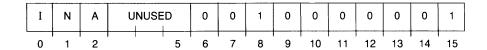


Detailed command description

This command reads data from the disk into host memory. You specify how much data (i.e., how many sectors) with the CB transfer count.

Read/Verify (201)

Compares data from the disk with data in host memory.

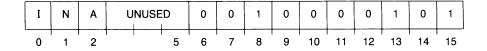


Detailed command description

This command compares, word for word, data on the disk with data in host memory. You specify the amount of data with the CB transfer count. If the data does not match, a verify error occurs. (A verify error is recorded in the CB error status word.)

Read/Verify Single Word (205)

Compares a single data word to each word of data in a sector.

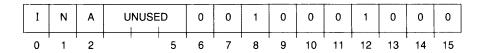


Detailed command description

This command compares a single word of data in host memory with each word of data in a sector. If the data does not match, a verify error occurs. (A verify error is recorded in the CB error status word.)

Read Raw Data (210)

Reads the header, data, and check words (checksum and ECC) of a sector into host memory.



Detailed command description

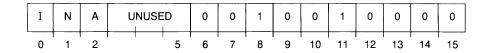
This command reads the entire data portion of a disk sector into host memory—including data that is normally transparent to the host (i.e., Header and ECC).

The controller does not test the headers, check for defects, or perform ECC calculations when it executes a Read Raw Data command. The data is read into host memory just as it appears on the disk, hence the name "raw data." For each sector you specify in the CB transfer count, the controller will read the following into host memory:

- 3 header words
- 256 data words
- A data checksum word
- 4 ECC words

Read Headers (220)

Reads a disk sector header into host memory.



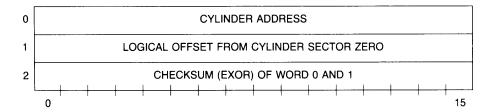
Detailed command description

This command reads a specified number of disk sector headers into host memory.

You specify the number of headers with the CB transfer count. For example, if you set the transfer count to 10 sectors, the controller will read 30 header words into host memory.

Disk sector header format

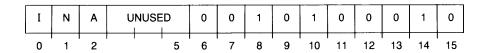
A disk sector header has this format:



Word	Bits	Name	Contents or Function
0	0-15	Cylinder Address	The cylinder address the controller was using when the read occurred.
1	0-15	Logical Offset from Cylinder Sector Zero	Logical sector offset the controller was using when the read occurred.
2	0-15	Checksum of Word 0 and 1	The checksum words 0 and 1 of the disk sector header.

Read with Modified Bit Map (242)

Reads a 256-word modified bit map and up to 4096 sectors of contiguous data into a specified address in host memory.



Detailed command description

This command reads a 256-word modified bit map and up to 4096 sectors of contiguous data into a specified address in host memory.

The complete bit map is transferred to the host. Each bit in the map corresponds to one sector of transferred data. The high-order bit of the map corresponds to the first sector of data transferred. If a sector's modified bit is set, the corresponding bit in the map will be set. Chapter 2 describes setting and clearing modified bits (see "Unit Information Block").

Since the modified bit map is 256 words, the data transfer always begins at the CB transfer address plus 256. You specify the number of sectors to scan with the CB transfer count. The maximum transfer count for this command is 4096 sectors (256-word map \times 16 bits per word = a 1:1 bit and sector mapping of 4096).

This command always transfers the whole 256-word bit map, but will transfer only the sectors whose modified bits are set. The bit map is not included in the transfer count.

The return transfer count is the number of scanned sectors whose modified bits were set. This count therefore reflects the number of sectors read into host memory.

Write Commands

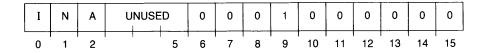
Table 3-11 lists the CB commands that let you perform different disk write operations.

Command Name Octal Code Function Write 100 Writes data from host memory onto the disk. 101 Write/Verify Writes data from host memory onto the disk and performs a read/verify operation. 104 Write Single Writes a single word from host memory, repetitively, onto the Word 105 Write/Verify Writes a single word from host memory onto the disk and performs a read/verify single word operation. Single Word Write with Writes up to 4096 sectors of data onto the disk using a 142 Modified Bit 256-word bit map. Мар

Table 3-11. CB Write Commands

Write (100)

Writes data from host memory onto the disk.

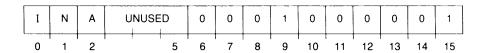


Detailed command description

This command writes data from host memory onto the disk.

Write/Verify (101)

Writes data from host memory onto the disk and performs a read/verify operation.



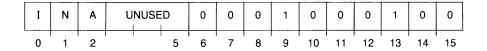
Detailed command description

This command first performs a normal write operation (i.e., writes data from host memory onto the disk). It then performs a read/verify operation to determine whether the write was successful.

The verify operation takes one additional spin of the disk for each track involved in the transfer.

Write Single Word (104)

Writes a single word from host memory, repetitively, onto the disk.



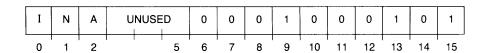
Detailed command description

This command writes a single word from host memory repetitively to the disk. The data area in host memory is therefore just one word long.

You specify the number of disk block writes for this command in the CB transfer count.

Write/Verify Single Word (105)

Writes a single word from host memory onto the disk and performs a read/verify single-word operation.



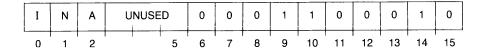
Detailed command description

This command first performs a normal single-word write operation (i.e., writes a single word from host memory onto the disk). It then performs a read/verify single-word operation to determine whether the write was successful.

The verify operation takes one additional spin of the disk for each track involved in the transfer.

Write with Modified Bit Map (142)

Writes up to 4096 sectors of contiguous data, beginning at the specified address, onto the disk.



Detailed command description

This command writes up to 4096 sectors of contiguous data beginning at a specified disk address. The amount and location of the transfer is indicated by the 256-word modified bit map.

The complete bit map is transferred to the controller. Each bit in the map corresponds to one sector of transferred data. The high-order bit of the map corresponds to the first sector of data transferred. If a sector is to be written, the corresponding bit in the map must be set.

Since the modified bit map is 256 words, the data transfer always begins at the CB transfer address plus 256. You specify the number of sectors to be written with the CB transfer count. This count indicates the number of bits in the bit map to be considered. The maximum transfer count for this command is 4096 sectors (256-word map \times 16 bits per word = a 1:1 bit and sector mapping of 4096).

This command always transfers the whole 256-word bit map. However, the size of the data transfer is the lesser of the following two numbers:

- The number of sectors whose associated bits in the bit map are set.
- The number of sectors specified in the transfer count (the bit map is not included in this count).

This command sets the modified bits for all transferred sectors to the state of the modified bit in the unit information block (i.e., set or cleared).

The return transfer count indicates the number of sectors actually written.

CHAPTER 4 SUBSYSTEM STATUS INTERFACE

This chapter describes the Disk Subsystem's status interface. Most of the chapter describes how the controller and drive report errors, interrupts, and status words. The first section introduces some terms used throughout the chapter.

CONTROLLER STATES TERMINOLOGY

Table 4-1 lists terms associated with controller states.

Table 4-1. Controller States Terminology

Term	Description
Busy	The controller is performing a task. The controller's Busy flag indicates whether the controller is busy.
Done	The controller has completed the previous task and is waiting for the next one; the host interrupt is active. The controller's Done flag indicates this state.
IORST	A mnemonic that means I/O reset. If you issue the assembler IORST instruction, the controller treats it like the Reset command except that it does not perform some internal diagnostics.
Power-up	The controller is reset and ready, but not initialized.
Ready	The drive is ready to perform an operation.
Reset	The controller resets its hardware. It resets Busy and Done flags, clears all interrupts, and executes internal diagnostics.

ERROR SUMMARY

The controller reports a number of different types of errors. Figure 4-1 illustrates error classifications. The two major classifications are generated by PIO commands and control blocks.

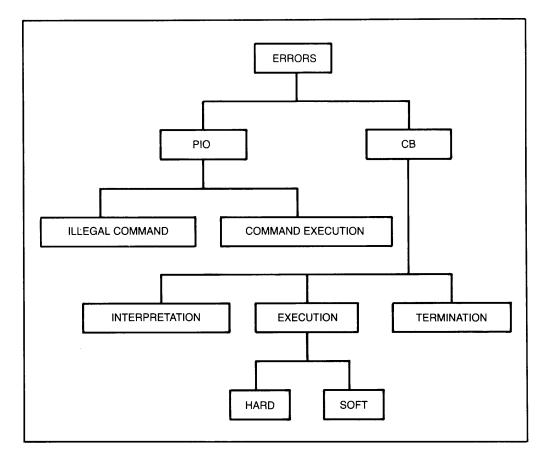


Figure 4-1. Error Classifications

Table 4-2 summarizes the primary and secondary mechanisms for reporting these errors. The sections following the table expand this information.

CB Execution Error

CB execution errors are the only errors the controller retries. They may be soft or hard errors.

Soft Errors

A soft CB execution error occurs when the controller retries an operation for any reason and the retry is successful.

When a soft CB execution error occurs, the controller does the following:

- It sets bit 2 of the CB status word.
- It sets bits 4 and 5 of status register C to 00 to indicate an asynchronous interrupt has occurred.

Table 4-2. Error Reporting

Error Class	Primary Reporting Mechanism	Secondary Reporting Mechanism
CB Execution Error	Status register C: Bits 4-5 = 00 Bits 6-15 = asynchronous interrupt code Bit 0 or 2 of the CB status	CB error status CB unit status Extended status
CB Interpretation Error	word = 1 Status register C: Bits 4-5 = 00 Bits 6-15 = asynchronous interrupt code	None
	Bit 1 of the CB status word = 1	
CB Termination Error Status register C: Bits 4-5 = 00 Bits 6-15 = 000000110 Bit 3 of the CB status		None
PIO Command Execution Error	word = 1 Status register C: Bits 4-5 = 10 Bits 6-15 = PIO command code	Status register B contains specific faults for these commands: Begin, Get Information, Sysgen, and Set Information
PIO Illegal Command Error	Status register C: Bits 4-5 = 01 Bits 6-15 = PIO command code	None

- If the S bit in the unit information block is set:
 - $-\,\text{It}$ writes asynchronous interrupt code (007) into bits 6-15 of status register C.
 - It suspends CB execution for the associated unit.
 - It generates the interrupt.
 - It waits for a Restart command.
- If the S bit is not set:
 - It writes (002) into bits 6-15.
 - It continues CB execution.
 - It generates the interrupt at the end of the list.

Hard Frrors

A hard CB execution error occurs when the controller cannot retry an error successfully. The controller retries an error until the retry count is exceeded.

When the retry count is exceeded, the controller does the following:

- It sets bit 0 of the CB status word.
- It suspends CB execution for the associated unit.
- It sets status bits 4 and 5 of status register C to 00.
- It writes (003) (the asynchronous interrupt code for a hard CB execution error) into bits 6-15 of status register C.
- It generates the interrupt.
- Finally, it waits for a Restart command.

For both soft and hard errors, the controller writes the CB error and unit status words and the number of retries performed. The error status word indicates the specific soft or hard error that occurred. The unit status word further classifies an error.

CB Interpretation Error

A CB interpretation error occurs in two instances:

• When you specify an illegal value for any of these CB fields:

CB status Transfer count
CB operation code Transfer address
Link address Unit number

Page number list address

NOTE: Addresses are illegal if they are outside the controller's fetching range. A CB status word is illegal if its initial value is not zero. A transfer count is illegal if it is outside the specified command's range. A unit number is illegal if it is not 0, 1, 2, or 3.

 When the sum of a CB's device address and transfer count exceed the disk's capacity.

If a CB interpretation error occurs, the controller performs no retries. Instead, it does the following:

- It sets bit 1 of the CB status word.
- It sets status bits 4 and 5 of status register C to 00.

- It writes an asynchronous interrupt code into bits 6-15 of status register C.
- It terminates the list.
- It generates the interrupt.

Asynchronous interrupts are detailed later in this chapter. Here are the interrupts that indicate CB interpretation errors:

- Status Word Not Zero (010)
- Illegal CB Command (011)
- CB Range Error (012)
- Illegal Unit Number (013)
- Illegal Link Address (014)
- Illegal Page Number List Address (015)
- Illegal Transfer Address (016)
- Illegal Transfer Count (017)

CB Termination Error

A CB termination error occurs when you issue a Cancel List command to the controller. A Cancel List is not an error as such, but the controller treats it as one. That is, the controller generates an asynchronous interrupt when it receives a Cancel List command.

If this error occurs, the controller performs no retries. Instead:

- It sets bit 3 of the CB status word.
- It sets status bits 4 and 5 of status register C to 00.
- It writes (006) (the asynchronous interrupt code for CB Termination Error: Cancel List) into bits 6-15 of status register C.
- It generates the interrupt.

You should not assume the controller has cancelled the list until you receive this interrupt from the controller. This interrupt is generated in addition to any other interrupts that may be generated by the Cancel List command (that is, the return request bit is set).

PIO Command Execution Error

Any of the following will cause a PIO command execution error:

- The controller boot is unsuccessful (the Begin command failed).
- You issue the Diagnostic Mode command, and status registers A and B do not contain 0 or (125252).
- You issue the Set Mapping Information command, and the last map slot assigned to the controller is greater than (1777). (The last map slot assigned is the first map slot assigned plus the number of map slots assigned. The upper bounds is (1777) because this controller does not support DCH.)
- You issue the Restart command when there was no CB error.
- You supply a CB address outside the controller's fetching range.
- A BMC error on Get/Set information commands.
- You issue commands to set mapping or set information, and the controller or unit is not idle.

If this error occurs, the controller performs no retries. Instead:

- It sets bits 4 and 5 of status register C to 10.
- If you issued a Begin, Set, or Get command, it writes error information into status registers A and B.
- It generates a synchronous interrupt to the host.

PIO Illegal Command Error

If you do either of the following, a PIO illegal command error occurs:

- You try to execute an unimplemented PIO command.
 - In this case, you load an invalid command code into bits 6-14 of command register C. The controller cannot interpret an invalid code.
- You try to execute a PIO command at the wrong time.
 - In this case, you try to execute a command out of controller state sequence. Reset and Diagnostic Mode are legal commands at anytime. "Command Status," later in this chapter, describes when the other PIO commands are legal.

If this error occurs, the controller performs no retries. Instead:

• It sets bits 4 and 5 of status register C to 01.

- It writes the illegal command code into bits 6-14 of status register C.
- It generates a synchronous interrupt to the host.

STATUS WORDS

Status words are a group of bits that indicate certain controller or drive states. These bits may indicate error conditions.

Status words are reported through these mechanisms:

- Control Block Written after a CB executes.
- Extended Status Written, on request, if a CB execution error occurs.
- Status Registers Written when an interrupt occurs.
- PIO Command Written into status registers after certain PIO commands execute.

The following sections discuss all of the status words reported through these mechanisms, except extended status words. Extended status words are useful to Data General personnel only, and as such are discussed in internal documents.

Control Block Status Words

A control block (CB) contains these status words:

- CB status in word 11.
- Error status in word 14.
- Unit status in word 15.

The following sections detail these status words; Chapter 2 explains control blocks.

CB Status

The controller writes the CB status word after the following:

- All operations for the CB have ceased.
- The controller has written return information for the CB.

If the interface information block allows, the controller always writes the CB status word, whether the CB completes successfully or with errors.

The CB status word must be 0 when you enqueue the CB to the controller.

When the CB completes, the status word has the following format:

Table 4-3. CB Status Word

Bits	Name	Meaning If Set:
0	Any CB hard execution error	A CB hard execution error occurred during CB execution (see Note 1).
1	Interpretation error	A CB interpretation error occurred (see Note 2).
2	Soft execution errors; controller recovered	Soft errors occurred during CB execution, but the controller recovered (see Note 1).
3	CB termination by Cancel List command	The CB list was terminated by a Cancel List command (see Note 3).
4	ECC correction needed	ECC correction was used to recover from an execution error.
5	ECC correction failed	ECC correction was applied, but failed to correct the error.
6-14	_	Unused.
15	CB Done bit	The controller has finished processing the CB (see Note 4).

NOTES:

1. If a soft or hard error occurs, the controller writes the extended status information block and generates an asynchronous interrupt. This block reflects the state of the controller and drive when the last execution error occurred.

If you want status after every retry, you must set the S bit in the unit information block. This causes an interrupt on each soft error, with attendant status reported.

You must always issue the Restart command after servicing the interrupt.

- 2. A CB interpretation error is reported as an asynchronous interrupt in status register C.
- 3. After a Cancel List command, the controller does the following:
 - a. Terminates the command (without retries).
 - b. Inhibits execution of remaining CBs in the list.
 - c. Writes the CB status word with bit 3 set.
 - d. Writes (006) (asynchronous interrupt code for CB Termination Error: Cancel List) into status register C.
 - e. Writes the address of the first CB in the list into status registers A and B.

4. The controller sets the Done bit last — after it has finished all other processing for a CB, but before it generates the interrupt. Thus, you can traverse CB lists, checking this bit setting, to determine which CBs the controller has completed. This is useful, for example, to recover after a power failure.

Error Status

The error status word indicates the last occurrence of any CB error. It is valid only if bit 0 or 2 of the CB status word is set. If bit 0 is set, a hard CB execution error occurred. If bit 2 is set, a soft error occurred.

The error status word indicates what kind of soft or hard error occurred. If a soft error occurred, the controller retried the operation. The CB retries performed field (CB word 16) indicates the number of attempts needed to recover.

The error status word has this format:

Table 4-4. CB Error Status Word

Bits	Name	Meaning If Set:
0	Interrupt Timeout	The controller timed out while waiting for an interrupt from the drive. It missed the interrupt. (The timer is variably programmed depending upon the disk task.)
		Controller action: Issues one hard reset to the drive and retries the operation.
1	Drive Interface Fault	The controller timed out while trying to pass control or status bytes to or from the drive. This indicates a breakdown in controller/drive communication.
<u> </u>		Controller action: Issues one hard reset to the drive and retries the operation.
2	2901 Timeout	The controller timed out because its 2901 processor did not complete a command within the specified time. (The 2901 can read or write an entire cylinder at a time. Therefore, this timeout must allow for at least 16 spins of the disk. The timer is programmed accordingly.)
		Controller action: Logs a controller retry and restarts the command if the retry count has not been exceeded.
3	Buffer Overflow (Data Late)	The disk requested data faster than the BMC could supply it, or the disk supplied data faster than the BMC could remove it. This bit corresponds to the "data late" condition.
		Controller action: Logs a controller retry and restarts the command if the retry count has not been exceeded.
4	Controller Detected Command	The controller detected a checksum error with the status block transferred from the drive.
	Checksum Error	Controller action: Logs a controller retry and restarts the command if the retry count has not been exceeded.

Table 4-4. CB Error Status Word (continued)

Bits	Name	Meaning If Set:
5	Drive Error	The controller could not complete an operation because the error bit in the unit status word was set (bit 1). (The drive indicated an error.)
		Controller action: Logs a unit retry, resets the drive error with a soft reset, and restarts the command.
6	BMC Timeout	The BMC failed to complete the data transfer.
	:	Controller action: Logs a controller retry and restarts the command if the retry count has not been exceeded.
7	Ending Memory Address Error	The BMC transferred too much or too little data.
	Address Ello	Controller action: Logs a controller retry and restarts the command if the retry count has not been exceeded.
8	Data Checksum Error	A checksum miscompare occurred on a data field during a data transfer. (The checksum provides protection for the ECC.)
		Controller action: Logs a controller retry and restarts the command if the retry count has not been exceeded. Otherwise, logs a read retry and tries to correct the error.
9-10	_	Reserved.
11	Verify Error	The controller sensed a miscompare between data read from the disk and data read from the BMC on a verify operation.
		Controller action: Logs a controller retry and restarts the command if the retry count has not been exceeded.
12	BMC Error	Faulty BMC operations.
		Transfers into the host: The BMC checks parity on the address and data lines and reports errors to the controller.
		Transfers out of the host: The controller checks the parities.
		Controller action: Logs a controller retry and restarts the command if the retry count has not been exceeded.
13	Data Parity Error	The controller detected a parity error on data received from the drive.
		Controller action: Logs a controller retry and restarts the command if the retry count has not been exceeded.
14	ECC Detected Error	The ECC registers contained a nonzero remainder.
		Controller action: Logs a controller retry and restarts the command if the retry count has not been exceeded. Otherwise, logs a read retry and tries to correct the error.
15	Header Noncompare	A header miscompare occurred for a sector that should have been valid.
		Controller action: Logs a controller retry and restarts the command if the retry count has not been exceeded.

Unit Status

The unit status word indicates a drive's state after it completes an operation. The status applies to the unit specified in CB word 9.

Table 4-5 shows the format of the unit status word. Note that bits 0-7 are always valid. The other status bits are valid when bit 2 (Ready) is set to 1, bit 3 (Busy) is clear, and bits 4-5 (Port Reserve Bits) point to the current port (port 01).

Table 4-5. CB Unit Status Word

Bits	Name	Meaning If Set:	
0	Command Failed	The drive terminated an operation initiated by the controller (see Note 1).	
1	Power Fail	The drive detected a power fail (see Note 2).	
2	Ready	The following conditions exist for the drive:	
		Power applied	
		On-line status attained	
		Proper spindle speed achieved	
		Proper operating temperature sensed	
		Power-up diagnostics completed	
		Controller communications enabled	
3	Busy	The drive received a command from the controller. The drive clears this bit when it completes execution of the command, or when an error results.	
4-5	Port Reserve Bits	Indicate which port has control:	
		Bit Setting Meaning 00 Available to either port 01 Reserved by this port 10 Reserved by other port 11 Unused	
6-7	Unit Number	Indicate the drive's unit number:	
		Bit Setting Meaning 00 Unit 0 01 Unit 1 10 Unit 2 11 Unit 3	
8	Logic Fault	The drive sensed an internal fault in the logic circuitry.	
9	Power Fault	The drive sensed an internal fault in the power circuitry.	
10	Servo Data Fault	The drive sensed an internal fault in the servo data circuitry.	

Table 4-5. CB Unit Status Word (continued)

Bits	Name	Meaning If Set:
11	Positioner Fault	The drive sensed an internal fault in the positioner circuitry.
12	Read/Write Fault	The drive sensed an internal fault in the read/write circuitry.
13	Bus Fault	The drive sensed an internal fault in the bus between the drive and the controller.
14-15		Reserved.

NOTES:

1. The command failed bit is linked with a command sent to the drive.

The drive sets this bit if it terminates an operation initiated by the controller. The bit stays set until the controller sends another command to the drive. When the drive receives the command, it clears the command failed bit.

When the drive sets this bit, all drive status is frozen until you issue a Restart command. The condition that caused a drive to terminate an operation is encoded in the drive's status report.

2. When an active unit experiences an impending loss of AC power, the drive sets this interrupt. It can be cleared by a C Pulse.

Whether or not the CPU senses a powerfail determines recovery procedures.

When one or more units report a power fail that is not sensed by the CPU, a
Restart command is the preferred method of recovery.

If the power fail is on execution of a CB, the controller acts exactly as it does with a CB execution error.

- It writes the unit status into the active CB.
- It writes the CB address in DIA/DIB.
- It generates an asynchronous interrupt to the host (077).
- It waits for a Restart command for that unit.

After the Restart command, any commands sent to the failed device cause a Hard CB error (either a Timeout or an Interface Fault.)

If the power fail is on a PIO command, a PIO execution error results and is reported to the host after a 100 millisecond delay.

- When the CPU senses power failure, the preferred recovery method is as follows:
 - 1. Get Unit Status from the unit to which Begin is to be issued. (Repeat this step if Unit Status does not have Ready set or if another power fail is reported.)
 - 2. Issue the Begin command.
 - 3. Reissue all Set Information block commands.
 - 4. Reissue all unfinished lists.

If the controller power was lost, you must issue the Reset, Begin, and initialization command sequence. (If the Sysgen command was issued before the power loss, the initialization sequence is omitted.) The operating system is responsible for saving and restarting interrupted CBs.

Status Register Status Words

The controller writes status words into its status registers (A, B, and C) when an interrupt occurs. These words depend on the command and the type of interrupt. Command dependent status words are described next ("PIO Command Status Words"); interrupt dependent status words are described in Chapter 3. This section describes the contents of status register C.

Here is what register C contains after an interrupt:

- Status bits that indicate the command state.
- Command completion status bits that indicate the type of interrupt.
- An interrupt code, if an asynchronous interrupt occurred.
- A command code and return request bit setting, if a synchronous interrupt occurred.

The command code and return request bit setting are also described in Chapter 3. The sections below describe the status bits, command completion status bits, and asynchronous interrupt codes.

Command Status

Bits 0-3 of status register C are called command status bits. These four bits indicate the controller command execution states. Note that the Diagnostic Mode (024) and Reset (777) commands are valid at anytime.

Table 4-6 shows command status values at various command execution states.

Table 4-6. Command Status

Bit Setting	Meaning
0000	A reset sequence is in progress, caused by a power up, IORST, or a Reset (777) command.
0001	The reset sequence is complete.
	If status register B does not contain zero, an internal error occurred and the register's contents indicate the type of error.
-	If an error occurred, legal commands are:
	Diagnostic Mode (024) Reset (777)
	If no error occurred, legal commands are:
	Program Load (000) Begin (002) Diagnostic Mode (024) Unit Status (131) Trespass (132) Reset (777)
	Any other command will cause a PIO illegal command error (bits 4 and 5 are set to 01).
	If a Begin is issued, the controller boots its processor code from the disk. If an error occurs during the boot, a PIO command execution error results, and the controller sets bits 4 and 5 to 10.
1000	Begin command executed successfully; controller is ready but not initialized. Legal commands are:
	Diagnostic Mode (024) Set Mapping Information (026) Get Mapping Information (027) Unit Status (131) Trespass (132) Reset (777)
	Any other command will cause a PIO illegal command error (bits 4 and 5 are set to 01).
11xx	Set Mapping Information executed; controller is ready and initialized with default parameters. All commands are now legal except the following:
	● Begin (002)
	Any diagnostic command (6xx)
	Program Load (000)
	Diagnostic commands are not valid, since the controller is not in diagnostic mode.
01xx	Diagnostic Mode. All commands from the previous state and from the diagnostic command set are now valid. Exiting diagnostic mode causes a return to the previous state.

Table 4-6. Command Status (continued)

Bit Setting	Meaning	
The following command status values indicate how many more Start List commands the controller can accept:		
x100	Not full: At least 2 more Start List commands can be accepted.	
x101	Near full: One more Start List command can be accepted.	
x111	Full: No more Start List commands can be accepted.	

Command Completion Status

The command completion status bits (status register C; bits 4-5) indicate whether an asynchronous or synchronous interrupt has occurred. You should test these bits first to see if an error occurred (see Table 4-7).

Table 4-7. Command Completion Status

Bit Setting	Meaning
00	An asynchronous interrupt occurred.
01	A PIO illegal command error occurred (you supplied an invalid command code).
10	A PIO command execution error occurred (the controller could not complete the command successfully).
11	The PIO command completed successfully.

Asynchronous Interrupts

The controller uses asynchronous interrupts to inform the host of events such as the completion of a CB or the occurence of certain CB errors. An asynchronous interrupt can be generated only when Busy and Done are clear. To generate the interrupt, the controller encodes DIC to indicate interrupt type and sets the Done flag.

Return information from a PIO command will supercede asynchronous information information loaded into DIA/B/C. When this occurs, the asynchronous interrupt will be reissued when Busy and Done are again clear.

If the command completion status is 00, then status register C indicates that an asynchronous interrupt has occurred. Bits 6-15 contain an asynchronous interrupt return code. This is an octal code that indicates what kind of asynchronous interrupt occurred. Table 4-8 lists the octal asynchronous interrupt codes and their meanings, and the controller's and host's action, if any.

Once an asynchronous interrupt has occurred, the host must service it before issuing any new asynchronous interrupts. To do this, the host issues a C pulse when the Done flag is set and DIA/B/C are loaded with asynchronous interrupt return information.

Table 4-8. Asynchronous Interrupts

Octal Code	Interrupt Name	Meaning
0	Null Interrupt	The controller received a P pulse from the host while the Busy flag was set and there are no other interrupts pending.
		Controller action: The controller is in the "Interrupt on Not Busy" state. A null interrupt indicates that the controller is not busy.
1	Controller Panic	Stack pointer and program counter at error (see Note 1).
		Controller action: Stops dead.
		Host action: Reset and reinitialize the controller.
2	CB Execution Error: Soft Errors	The controller encountered errors during execution, but recovered from them. The list is completed.
		Controller action: Issues soft reset. Sets bit 2 (soft execution errors; controller recovered) in the CB status word. Writes extended status information block.
		Host action: None required.
3	CB Execution Error: Hard Errors	The controller encountered errors during execution and could not recover from them.
		Controller action: Tried to recover, but could not succeed before exceeding the retry count. Sets bits 0 (any CB hard execution) and 15 (CB Done bit) in the CB status word. Writes extended status information block.
		Host action: Scan CB list to find the CB that caused the error. Correct error, if possible. Issue the Restart command.
4	l Bit Set	The controller executed a CB successfully and the CB I bit was set.
		Controller action: Continues list execution if there are more CBs in the list (see Note 2).
		Host action: Service the interrupt.
5	No Errors	The controller executed all CBs in a list successfully.
		Controller action: Sets bit 15 (CB Done bit) in the CB status word.
6	CB Termination Error: Cancel List	The controller received a Cancel List command from the host.
		Controller action: Sets bit 3 (CB termination by Cancel List command) in the CB status word.
7	Soft Error: S Bit	Occurs if three conditions are met:
	Set	The unit information block S bit is set.
		 The controller information block retry counts are not full.
		A CB execution error has occurred.

 Table 4-8.
 Asynchronous Interrupts (continued)

Octal Code	Interrupt Name	Meaning
		Controller action: Saves extended status information block.
10	CB Interpretation Error: Status Word Not 0	Host action: Issue the Restart command.
		The initial value of a CB status word (CB word 11) is nonzero.
		Controller action: Does not retry or overwrite the CB status word. Terminates list execution.
		Host action: Initialize CB status word to 0. Reissue the Start List command for the CB in error.
11	CB Interpretation Error: Illegal CB Command	A CB's command code is illegal.
		Controller action: Does not retry. Terminates list execution. Sets bit 1 (interpretation error) of the CB status word.
		Host action: Correct CB command. Reissue the Start List command for the CB in error.
12	CB Interpretation Error: CB Range Error	The sum of the device address and the transfer count exceeded the capacity of the disk, or a diagnostic operation code did not fall within the diagnostic area.
		Controller action: Does not retry. Terminates list execution. Sets bit 1 (interpretation error) of the CB status word.
		Host action: Correct device address, transfer count, or diagnostic operation code. Reissue the Start List command for the CB in error.
13	CB Interpretation Error: Illegal	A CB's unit number is illegal.
	Unit Number	Controller action: Does not retry. Terminates list execution. Does not write the CB status word.
		Host action: Correct unit number. Reissue the Start List command for the CB in error.
14	CB Interpretation Error: Illegal Link Address	A CB's link address is outside the controller's fetching range (see Note 3).
	LINK Address	Controller action: Does not retry. Terminates list execution. Sets bit 1 (interpretation error) of the CB status word.
		Host action: Correct link address. Reissue the Start List command for the CB in error.
15	CB Interpretation Error: Illegal Page Number List Address	A CB's page number list address is outside the controller's fetching range (see Note 3).
		Controller action: Does not retry. Terminates list execution. Sets bit 1 (interpretation error) of the CB status word.
		Host action: Correct page number list address. Reissue the Start List command for the CB in error.

Table 4-8. Asynchronous Interrupts (continued)

Octal Code	Interrupt Name	Meaning
16	CB Interpretation Error: Illegal Transfer Address	A CB's transfer address is outside the controller's fetching range (see Note 3).
		Controller action: Does not retry. Terminates list execution. Sets bit 1 (interpretation error) of the CB status word.
		Host action: Correct transfer address. Reissue the Start List command for the CB in error.
17	CB Interpretation Error: Illegal Transfer Count	A CB's transfer count is outside the specified command's range. For example, a Read Modified Bit Map with a transfer count greater than 4096.
		Controller action: Does not retry. Terminates list execution. Sets bit 1 (interpretation error) of the CB status word.
		Host action: Correct transfer count. Reissue the Start List command for the CB in error.
20	Unreadable CB	The controller could not read the CB (see Note 4).
		Controller action: Does not retry or write the CB status word. Terminates list execution.
		Host action: Correct the problem and reissue the Start List command for the CB in error.
21	Unwritable CB	The controller could not write completion information (e.g., Status Word) after executing a CB.
		Controller action: Does not retry or write the CB status word. Terminates list execution.
		Host action: Correct the problem and reissue the Start List command for the CB in error.
77	Power Fail	The entire computer system lost power or the just the disk subsystem lost power.
		Controller action: Does not retry. Terminates list execution. Sets bit 1 (power fail) in the unit status word.
		Host Action: If the entire system lost power, the host will restore power and issue the Reset, Begin, and controller initialization sequence. (The Reset command is the only valid command after a system power fail.)
		If only the drive lost power, restore power, issue a Restart command and continue.

NOTES:

- 1. The following will cause Controller Panic interrupts:
 - Jumping to location 0000H
 - P pulse received with an interrupt active

- Internal CB list corrupted
- Internal database corrupted
- Some sectors not transferred on I/O but no errors received
- Internal page list corrupted
- Internal semaphores corrupted
- Internal process descriptors corrupted
- 2. The controller will not generate the I Bit interrupt for the last CB in a list. Instead, it will generate one of the termination interrupts.
- 3. A mapped address is outside the controller's fetching range if bits 1-11 of the high-order word are not zero. An unmapped address is outside the controller's fetching range if bits 1-10 of the high-order word are not zero unless Upstream Map Loading has been selected.
- 4. Either of the following will cause unreadable or unwritable CBs:
 - Transient BMC errors during a CB fetch or page number list fetch.
 - Invalid mapping information for mapped CBs or page lists.

PIO Command Status Words

The following sections describe the status words that the controller returns after certain PIO commands execute: Begin, Get and Set, Get List Status, Program Load, and Unit Status commands.

Begin and Program Load

If a PIO command execution error occurs after a Begin or Program Load command, the controller takes the following actions:

- It interrupts the host.
- It writes the CB unit status word into status register A.
- It writes a modified version of the CB error status word into status register B.

The CB unit and error status words were detailed earlier in this chapter. The changes in the error status word affect bits 6, 7, 11, and 12 (see Table 4-9 and Table 4-10).

Table 4-9. Begin Command: Status Register B

Bits	Meaning If Set:	
0	Interrupt Timeout	
1	Drive Interface Fault	
2	2901 Timeout	
3	Buffer Overflow (Data Late)	
4	Controller Detected Checksum Error	
5	Drive Error	
6	Installed Code Checksum Error (see Note 1)	
7	Reserved (see Note 2)	
8	Data Checksum Error	
9-12	Reserved	
(see Note 3)		
13	Data Parity Error	
14	ECC Detected Error	
15	Header Noncompare	

Table 4-10. Program Load Command: Status Register B

Bits	Meaning If Set:
0	Interrupt Timeout
1	Drive Interface Fault
2	2901 Timeout
3	Buffer Overflow (Data Late)
4	Controller Detected Checksum Error
5	Drive Error
6	BMC Timeout
7	Ending Memory Address Error
8	Data Checksum Error
9-11	Reserved
(see Note 3)	
12	BMC Error
13	Data Parity Error
14	ECC Detected Error
15	Header Noncompare

NOTES:

1. A Begin operation cannot produce a BMC timeout error, so bit 6 in the Begin error status word indicates whether an installed code checksum error occurred.

When you issue a Begin command, the controller checks the disk's installed code. An installed code checksum error occurs when the controller detects a checksum miscompare between the calculated checksum and the actual checksum.

- 2. A Begin operation cannot generate an ending memory address error, so bit 7 is reserved.
- 3. A Begin operation cannot generate a verify or BMC error, so bits 11 and 12 are reserved.
- 4. A Program Load operation cannot generate a verify error, so bit 11 is reserved.

Get and Set

If a PIO command execution error occurs after a Get or Set command, the controller interrupts the host and writes the following information into status register B (Table 4-11):

Bits Name Meaning If Set: 0-6 Unused. **Ending Memory** 7 The ending memory address was not correct. Address Error Unused. 8-11 **BMC** Error Faulty BMC operations. 12 Transfers into the host: The BMC checks parity on the address and data lines and reports errors to the controller. Transfers out of the host: The controller checks the parities. Unused.

Table 4-11. Get and Set Command: Status Register B

Get List Status

13-15

After the Get List Status command executes, the controller writes the position (e.g., first—001, second—010) of the currently executing CB into status register B (see Table 4-12). It writes the following information into status register A:

Bits Name Meaning If Set: 0 List is Active The specified CB list is currently executing. The controller is trying to recover from an error in the specified CB list. Controller in 1 Error Recovery 2 Waiting for a The controller is waiting for you to issue a Restart command. Restart Timeouts have occurred on the active CB. 3 Timeouts have occurred Unused. 4-15

Table 4-12. Get List Status Command: Status Register A

Reset

When you issue the Reset command, the controller executes a series of internal diagnostics. When it finishes executing all of these diagnostics, it writes the results into status register B. If any bit is set, it means that the corresponding test failed. (See Table 4-13.)

Table 4-13. Reset Command: Status Register B

Bits	Meaning If Set:	
0	Disk Interface FIFO's test	
1	Sector Buffer and FIFO's	
	test	
2	Dirty Bit RAM test	
3	Error Flops test	
4	ECC/CRC test	
5	Sequencer/ALU test	
6	Microcode PROM test	
7	Control Tag Bus test	
8	Command Data Bus test	
9	Command Registers test	
10	Burst Length Compare test	
11	Transfer Counter test	
12	Address Counter test	
13	Interrupts test	
14	RAM test	
15	PROM test	

Unit Status

When you issue the Unit Status command, the controller copies the unit status word of the drive into status register B. The CB unit status word is described earlier in this chapter.

APPENDIX A PROGRAMMING SUMMARIES

This appendix summarizes the following disk subsystem controller and drive information:

- Controller Statistics
- Command Set Instructions, PIO Commands, and CB Commands
- Registers Command and Status
- Control and Information Blocks
- Status Words
- Mapping Options

CONTROLLER AND DRIVE SPECIFICATIONS

Tables A-1 and A-2 list general controller/drive specifications for Model 6236/6237 and the Model 6239/6240, respectively.

Table A-1. Model 6236/6237 Controller/Drive Specifications

Miscellaneous:	
Mnemonic	DSKP
Device code	24 (alternate 64)
Priority mask bit	7
Media/Heads:	
Number of disks	5
Number of data surfaces	8
Platter diameter	14 in.
Number of data heads	16
Number of servo heads	1
Bits per inch (bpi)	10,438
Flux changes per inch (fci)	7,829
Tracks per inch	714
Recording code	2-8
Capacity:	
Formatted capacity	354 Mbytes
Number of sectors/track	56
Number of data bytes per sector	512
Capacity per track	28.672 Kbytes
Capacity per cylinder	458.752 Kbytes

Table A-1. Model 6236/6237 Controller/Drive Specifications (continued)

Number of cylinders	789 (total)
User	786
Controller reserved	1
Internal diagnostics	1
Field Service diagnostics	1
Performance:	
Seek Time: Single track	8 ms
Average (1/3 stroke)	25 ms
Maximum (full stroke)	40 ms
Rotational latency	10 ms
Rotational speed	3000 rpm
Transfer rate	1.7 Mbytes/s
Error rate: Recoverable	10 ⁻¹⁰
Unrecoverable	10-12

Table A-2. Model 6239/6240 Controller/Drive Statistics

	Adianallana	
	Miscellaneous:	
	Mnemonic	DSKP
	Device code	24 (alternate 64)
	Priority mask bit	7
	Media/Heads:	
	Number of disks	5
l	Number of data surfaces	8
	Platter diameter	14 in.
	Number of data heads	16
	Number of servo heads	1
	Bits per inch (bpi)	14,154
ı	Flux changes per inch (fci)	10,615
	Tracks per inch	800
	Recording code	2-8
	Capacity:	
	Formatted capacity	592 Mbytes
	Number of sectors/track	75
	Number of data bytes per sector	512
	Capacity per cylinder	606.21 Kbytes
1	Number of cylinders:	981 (total)
	User	978 `
	Controller reserved	1
	Internal diagnostics	1
	Field Service diagnostics	1
	Performance:	
	Rotational latency	10.2 ms
	Rotational speed	2,940 rpm
	Transfer rate	2.20 Mbytes/sec
	Error rate: Recoverable	10-10
	Unrecoverable	10-12

COMMAND SET

This section summarizes instructions, PIO commands, and CB commands.

Instructions

Use the following syntax to code an instruction:

instruction_name[f] ac,DSKP

where:

instruction_name is the I/O instruction you want to issue:

DIA DIB DIC DOA DOB DOC

f is the pulse you want to issue (optional):

C = C pulse P = P pulse S = S pulse

ac is the host accumulator that contains the transfer information:

0 = AC0 1 = AC1 2 = AC2 3 = AC3

DSKP is the assembler mnemonic for the device code.

For a detailed description of I/O instructions, see Chapter 3.

PIO Commands

Table A-3 lists the PIO commands and their octal codes. Chapter 3 describes how to issue these commands.

Table A-3. PIO Commands

Octal Code	Command Name
000	Program Load
002	Begin
024	Diagnostic Mode (Enter/Exit)
025	Sysgen
026	Set Mapping Information
027	Get Mapping Information
030	Set Interface Information
031	Get Interface Information
032	Set Controller Information
033	Get Controller Information
034	Set Unit Information
035	Get Unit Information
040	Get Extended Status 0
041	Get Extended Status 1
042	Get Extended Status 2
043	Get Extended Status 3
100	Start List
103	Start List (High Priority)
116	Restart
123	Cancel List
131	Unit Status
132	Trespass
133	Get List Status
777	Reset

CB Commands

Table A-4 lists CB commands and their octal codes. Chapter 3 describes these commands in detail.

Table A-4. CB Commands

Octal Code	Command Name
000	No Operation
100	Write
101	Write/Verify
104	Write Single Word
105	Write/Verify Single Word
142	Write with Modified Bit Map
200	Read
201	Read/Verify
205	Read/Verify Single Word
210	Read Raw Data
220	Read Headers
242	Read with Modified Bit Map
400	Recalibrate Disk

REGISTERS

This section summarizes the controller's command and status registers.

Command Registers

Figure A-1 shows the controller's command registers. Table A-5 summarizes what the host accumulators must contain before you issue an instruction to the controller.

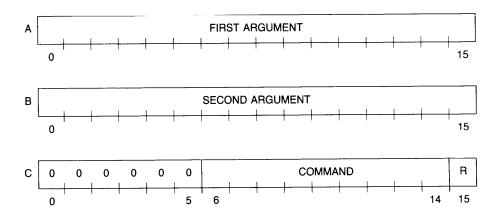


Figure A-1. Command Registers

Table A-5. Accumulator Contents

If you want to issue this command:	DOA accumulator should contain:	DOB accumulator should contain:	DOC accumulator should contain:
Program Load (000)	Unit number	Unused	000000000000000R
Begin (002)	Unit number	Unused	000000000000010R
Diagnostic Mode (024)	Enter: (125252) Exit: 0	Enter: (125252) Exit: 0	000000000010100R
Sysgen (025)	Unit Number	Unused	000000000010101R
Set Mapping Information (026)	Mapping options	Mapping options	000000000010110R
Get Mapping Information (027)	Unused	Unused	000000000010111R
Set Interface Information (030)	High-order address word of the interface information block	Low-order address word of the interface information block	00000000011000R
Get Interface Information (031)	High-order address word of the interface information block	Low-order address word of the interface information block	00000000011001R

 Table A-5.
 Accumulator Contents (continued)

If you want to issue this command:	DOA accumulator should contain:	DOB accumulator should contain:	DOC accumulator should contain:
Set Controller Information (032)	High-order address word of the controller information block	Low order address word of the controller information block	00000000011010R
Get Controller Information (033)	High-order address word of the controller information block	Low order address word of the controller information block	000000000011011R
Set Unit Information (034)	High-order address word of the unit infor- mation block	Low order address word of the unit information block	000000000011100R
Get Unit Information (035)	High-order address word of the unit infor- mation block	Low order address word of the unit infor- mation block	000000000011101R
Get Extended Status 0 (040)	High-order address word of the extended status information block for drive 0	Low-order address word of the extended status information block for drive 0	000000000100000R
Get Extended Status 1 (041)	High-order address word of the extended status information block for drive 1	Low-order address word of the extended status information block for drive 1	000000000100001R
Get Extended Status 2 (042)	High-order address word of the extended status information block for drive 2	Low-order address word of the extended status information block for drive 2	000000000100010R
Get Extended Status 3 (043)	High-order address word of the extended status information block for drive 3	Low-order address word of the extended status information block for drive 3	000000000100011R
Start List (100)	High-order address word of the CB list to execute	Low-order address word of the CB list to execute	00000001000000R
Start List (High Priority) (103)	High-order address word of the CB list to execute	Low-order address word of the CB list to execute	000000001000011R
Restart (116)	Unit number	Code Word	000000001001110R
Cancel List (123)	High-order address word of CB list to cancel	Low-order address word of CB list to cancel	000000001010011R
Unit Status (131)	Unit number	Unused	000000001011001R
Trespass (132)	Unit number	Unused	000000001011010R
Get List Status (133)	High-order address word of the first CB in a list	Low-order address word of the first CB in a list	000000001011011R
Reset (777)	Unused	Unused	000000111111111R

Status Registers: Asynchronous Interrupts

Figure A-2 shows the format of the controller's status registers after an asynchronous interrupt. Table A-6 summarizes what these registers contain if a certain interrupt occurred.

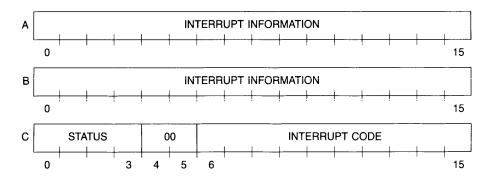


Figure A-2. Status Registers: Asynchronous Interrupts

Table A-6. Status Register Contents: Asynchronous Interrupts

If bits 6-15 of status register C contain this octal code:	Then this interrupt occurred:	And, registers A and B will contain:
000	Null Interrupt	Unused
001	Controller Panic	Stack pointer and program counter
002	CB Execution Error: Soft Errors	Double-word address of the first CB in the list
003	CB Execution Error: Hard Errors	Double-word address of first CB in the list
004	l Bit Set	Double-word address of the interrupting CB
005	No Errors	Double-word address of the first CB in the list
006	CB Termination Error: Cancel List	Double-word address of the first CB in the list
007	Soft Error: S Bit Set	Double-word address of the first CB in the list
010	CB Interpretation Error: Status Word Not 0	Double-word address of the first CB in the list
011	CB Interpretation Error: Illegal CB Command	Double-word address of the first CB in the list
012	CB Interpretation Error: CB Range Error	Double-word address of the first CB in the list
013	CB Interpretation Error: Illegal Unit Number	Double-word address of the first CB in the list

Table A-6. Status Register Contents: Asynchronous Interrupts (continued)

If bits 6-15 of status register C contain this octal code:	Then this interrupt occurred:	And, registers A and B will contain:
014	CB Interpretation Error: Illegal Link Address	Double-word address of the first CB in the list
015	CB Interpretation Error: Illegal Page Number List Address	Double-word address of the first CB in the list
016	CB Interpretation Error: Illegal Transfer Address	Double-word address of the first CB in the list
017	CB Interpretation Error: Illegal Transfer Count	Double-word address of the first CB in the list
020	Unreadable CB	Double-word address of the first CB in the list
021	Unwritable CB	Double-word address of the first CB in the list
077	Power Fail	Double-word address of the first CB in the list

Status Registers: Synchronous Interrupts

A synchronous interrupt occurs only when the return request bit is set or when an error occurs. Figure A-3 shows the format of the controller's status registers after a synchronous interrupt occurs. Table A-7 summarizes what these registers contain if you issued a certain command.

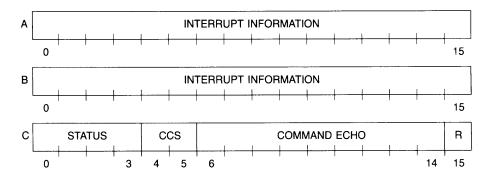


Figure A-3. Status Registers: Synchronous Interrupts

Table A-7. Status Register Contents: Synchronous Interrupts

If you issued this command:	Status Register A will contain:	Status Register B will contain:	Status register C (bits 6-15) will contain:
Program Load (000)	Unused	Unused unless a PIO command execution error occurs	000000000R
Begin (002)	Unused unless a PIO command execution error occurs	Unused unless a PIO command execution error occurs	00000010R
Diagnostic Mode (024) (Enter/Exit)	Unused	Unused	000010100R
Sysgen (025)	Unused unless a PIO command execution error occurs	Unused unless a PIO command execution error occurs	000010101R
Set Mapping Information (026)	Mapping options	Mapping options	000010110R
Get Mapping Information (027)	Mapping options	Mapping options	000010111R
Set Interface Information (030)	Unused	Unused unless a PIO command execution error occurs	000011000R
Get Interface Information (031)	Unused	Unused unless a PIO command execution error occurs	000011001R
Set Controller Information (032)	Unused	Unused unless a PIO command execution error occurs	000011010R
Get Controller Information (033)	Unused	Unused unless a PIO command execution error occurs	000011011R
Set Unit Information (034)	Unused	Unused unless a PIO command execution error occurs	000011100R
Get Unit Information (035)	Unused	Unused unless a PIO command execution error occurs	000011101R
Get Extended Status 0 (040)	Unused	Unused unless a PIO command execution error occurs	000100000R
Get Extended Status 1 (041)	Unused	Unused unless a PIO command execution error occurs	000100001R
Get Extended Status 2 (042)	Unused	Unused unless a PIO command execution error occurs	000100010R
Get Extended Status 3 (043)	Unused	Unused unless a PIO command execution error occurs	000100011R

Table A-7. Status Register Contents: Synchronous Interrupts (continued)

If you issued this command:	Status Register A will contain:	Status Register B will contain:	Status register C (bits 6-15) will contain:
Start List (100)	High-order address word of first CB in started list	Low-order address word of first CB in started list	001000000R
Start List (High Priority) (103)	High-order address word of first CB in started list	Low-order address word of first CB in started list	001000011R
Restart (116)	Unit number	Unused	001001110R
Cancel List (123)	High-order word address of the active CB in the terminated list; 0 if list not active	Low-order word ad- dress of the active CB in the terminated list; 0 if list not active	001010011R
Unit Status (131)	Unused	Unit status word	001011001R
Trespass (132)	Unit Number	Unused	001011010R
Get List Status (133)	List status	Number of current CB	001011011R
Reset (777)	Unused	Internal diagnostic results (zero if no errors)	000000000

CONTROL BLOCKS (CBs)

Figure A-4 shows the format of a control block. Chapter 2 describes these fields in detail.

INFORMATION BLOCKS

This section summarizes the controller information blocks:

- User Option Controller, interface, and unit
- Extended Status Ending memory address, controller error report, and drive error report

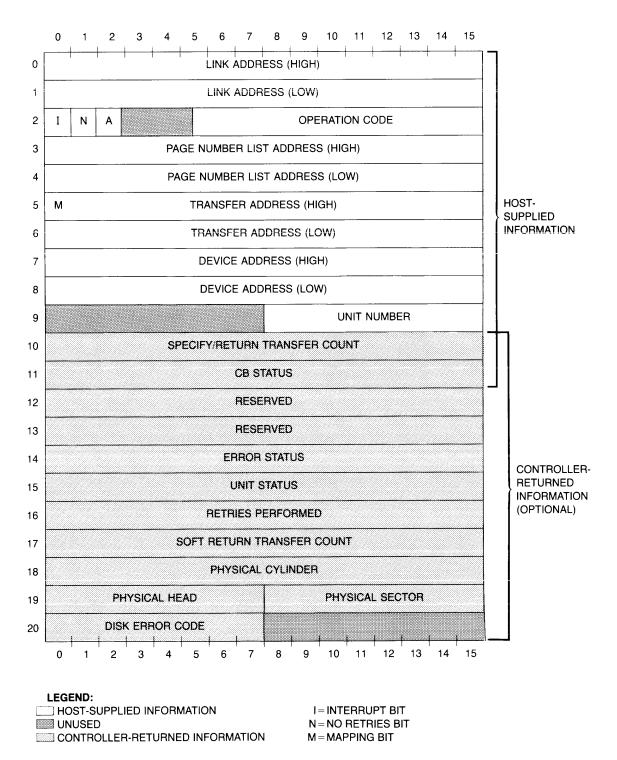
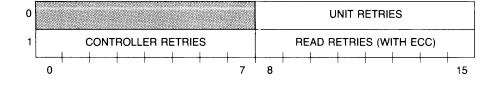


Figure A-4. Control Block

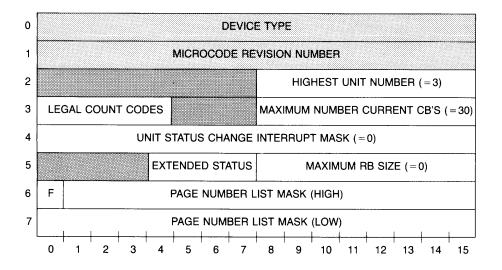
User Option

Figures A-5 through A-7 show the format of the controller, interface, and unit information blocks. Any specified values are default values for block fields. Chapter 2 describes these blocks in detail.



LEGEND: HOST-SUPPLIED INFORMATION UNUSED

Figure A-5. Controller Information Block



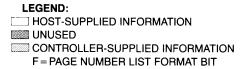


Figure A-6. Interface Information Block

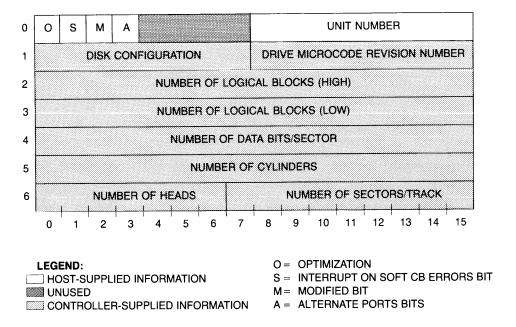


Figure A-7. Unit Information Block

Extended Status

The controller maintains extended status information in its memory. This status is stored in the extended status information block, which consists of these 32 words:

Words	Block Name	Contents
0-1	Ending Memory Address	Contains control information and the BMC address.
2-18	Controller Error Report	Contains a dump of the controller's 2901 registers.
19-23	-	Unused; all zeroes.
24-29	Drive Error Report	Contains an error code and drive status information.
30-31	_	Unused; all zeroes.

STATUS WORDS

This section summarizes the status words produced by the various reporting mechanisms:

- Control Block CB status, error status, unit status
- Status Register Command status (execution state and Start List), command completion status, and asynchronous interrupt codes
- Command Status Begin, Get and Set, Get List Status, Program Load, Reset, and Unit Status

Control Block Status Words

Tables A-8 through A-10 list the control block status words. Chapter 4 describes these status words in detail.

Table A-8. CB Status Word

Bits	Name
0	Any CB hard execution error
1	Interpretation error
2	Soft execution errors; controller recovered
3	CB termination by Cancel List command
4	ECC correction needed
5	ECC correction failed
6-14	Unused
15	CB Done bit

Table A-9. CB Error Status Word

Bits	Name
0	Interrupt Timeout
1	Drive Interface Fault
2	2901 Timeout
3	Buffer Overflow (Data Late)
4	Controller Detected Command Checksum Error
5	Drive Error
6	BMC Timeout
7	Ending Memory Address Error
8	Data Checksum Error
9-10	Reserved
11	Verify Error
12	BMC Error
13	Data Parity Error
14	ECC Detected Error
15	Header Noncompare

Table A-10. CB Unit Status Word

Bits	Name
0	Command Failed
1	Power Fail
2	Ready
3	Busy
4-5	Port Reserve Bits
6-7	Unit Number
8	Logic Fault
9	Power Fault
10	Servo Data Fault
11	Positioner Fault
12	Read/Write Fault
13	Bus Fault
14-15	Reserved

Status Register Status Words

Tables A-11 through A-14 summarize the contents of status register C. The contents of status registers A and B were summarized earlier (see "Registers"). The contents of all three registers are detailed in Chapters 3 and 4.

Table A-11 and Table A-12 summarize bits 0-3 of status register C. Table A-11 indicates controller execution states; Table A-12 indicates how many Start List commands the controller can accept.

Table A-11. Command Status: Execution State

Bit Setting	Meaning
0000	A reset sequence is in progress: caused by a power-up, IORST, or a Reset (777) command.
0001	The reset sequence is complete.
1000	Begin command executed successfully; controller is ready but not initialized.
11xx	Set Mapping Information executed; controller is ready and initialized with default parameters.
01xx	Diagnostic Mode. All commands from the previous state and from the diagnostic command set are now valid. Exiting diagnostic mode causes a return to the previous state.

Table A-12. Command Status: Start List

Bit Setting	Meaning
x100	Not full: At least two more Start List commands can be accepted.
x101	Near full: One more Start List command can be accepted.
x111	Full: No more Start List commands can be accepted.

Table A-13 summarizes bits 4 and 5 of status register C. These bits indicate what type of interrupt occurred and whether an error occurred at PIO execution time.

Table A-13. Command Completion Status

Bit Setting	Meaning
00	An asynchronous interrupt occurred.
01	A PIO illegal command error occurred (i.e., you supplied an invalid command code).
10	A PIO command execution error occurred (i.e., the controller could not complete the command successfully).
11	The PIO command completed successfully.

Table A-14 lists asynchronous interrupt codes and their meaning. These codes are written into bits 6-15 of status register C.

Table A-14. Asynchronous Interrupts

Octal Code	Interrupt Name
0	Null Interrupt
1	Controller Panic
2	CB Execution Error: Soft Errors
3	CB Execution Error: Hard Errors
4	l Bit Set
5	No Errors
6	CB Termination Error: Cancel List
7	Soft Error: S Bit Set
10	CB Interpretation Error: Status Word Not 0
11	CB Interpretation Error: Illegal CB Command
12	CB Interpretation Error: CB Range Error
13	CB Interpretation Error: Illegal Unit Number
14	CB Interpretation Error: Illegal Link Address
15	CB Interpretation Error: Illegal Page Number List Address
16	CB Interpretation Error: Illegal Transfer Address
17	CB Interpretation Error: Illegal Transfer Count
20	Unreadable CB
21	Unwritable CB
77	Power Fail

PIO Command Status Words

Tables A-15 through A-20 summarize the status words returned in status registers A and B when certain PIO commands execute. These words are detailed in Chapter 4.

Begin, Program Load, and Unit Status Commands

The controller writes the unit status word into status register B after you issue the Unit Status command. It also writes this word into status register A if a PIO command execution error occurs after you issue a Begin or Program Load command.

Table A-15. Unit Status Word

Bits	Meaning If Set:
0	Command Failed
1	Power Fail
2	Ready
3	Busy
4-5	Port Reserve Bits
6-7	Unit Number
8	Logic Fault
9	Power Fault
10	Servo Data Fault
11	Positioner Fault
12	Read/Write Fault
13	Bus Fault
14-15	Reserved

Table A-16 and Table A-17 summarize the contents of status register B if a PIO command execution error occurs after you issue a Begin or Program Load command.

Table A-16. Begin: Status Register B After an Error

Bits	Name
0	Interrupt Timeout
1	Drive Interface Fault
2	2901 Timeout
3	Buffer Overflow (Data Late)
4	Controller Detected Checksum Error
5	Drive Error
6	Installed Code Checksum Error
7	Reserved
8	Data Checksum Error
9-12	Reserved
13	Data Parity Error
14	ECC Detected Error
15	Header Noncompare

Table A-17. Program Load: Status Register B
After an Error

Bits	Name
0	Interrupt Timeout
1	Drive Interface Fault
2	2901 Timeout
3	Buffer Overflow (Data Late)
4	Controller Detected Checksum Error
5	Drive Error
6	BMC Timeout
7	Ending Memory Address Error
8	Data Checksum Error
9-11	Reserved
12	BMC Error
13	Data Parity Error
14	ECC Detected Error
15	Header Noncompare

Get and Set Commands

Table A-18 summarizes the contents of status register B if a PIO command execution error occurs after you issue a Get or Set command.

Table A-18. Get and Set Command Status

Bits	Name
0-6	Unused
7	Ending Memory Address Error
8-11	Unused
12	BMC Error
13-15	Unused

Get List Status Command

Table A-19 summarizes the contents of status register A after you issue a Get List Status command.

Table A-19. Get List Status Command Status

Bits	Meaning If Set:
0	List is Active
1	Controller in Error Recovery
2	Waiting for a Restart
3	Timeouts have occurred
4-15	Unused

Reset Command

Table A-20 summarizes the contents of status register B after you issue a Reset command.

Table A-20. Reset Command Status

Bits	Meaning If Set:	
0	Disk Interface FIFO's test	
1	Sector Buffer and FIFO's test	
2	Dirty Bit RAM test	
3	Error Flops test	
4	ECC/CRC test	
5	Sequencer/ALU test	
6	Microcode PROM test	
7	Control Tag Bus test	
8	Command Data Bus test	
9	Command Registers test	
10	Burst Length Compare test	
11	Transfer Counter test	
12	Address Counter test	
13	Interrupts test	
14	RAM test	
15	PROM test	

MAPPING OPTIONS

Table A-21 lists mapping options and indicates which are valid with this controller. These options are detailed in Chapter 2.

Table A-21. Mapping Options

Transferred To/From	Option	Default
Register A; bit 0	DMA over the Data Channel	Always 0; option not supported.
Register A; bit 1	DMA over the BMC	Always 1; only option supported.
Register B; bit 0	Mapping with Map Slot Load Interrupts	Always 0; not supported.
Register B; bit 1	Mapping with Internal BMC Physical Remapping	1 on power-up, Reset, or IORST.
Register B; bit 2	Mapping with Upstream Loading	1 on power-up, Reset, or IORST.
Register B; bit 3	Upstream Loading with Host Page Tables	1 on power-up, Reset, or IORST.
Register A; bits 6-15	First Map Slot Assigned	0 on power-up, Reset, or IORST.
Register B; bits 8-15	Number of Map Slots Assigned	0 on power-up, Reset, or IORST.

APPENDIX B OCTAL AND HEXADECIMAL CONVERSION

To convert a number from octal or hexadecimal (hex) to decimal:

- Locate the decimal equivalent for the octal or hex digit in that position in Table B-1 or B-2.
- Add the decimal equivalents to obtain the decimal number.

To convert a decimal number to octal or hexadecimal:

- Locate the decimal value in the appropriate table that is closest to number you want to convert.
- Note its octal or hex equivalent and column position.
- Find the decimal remainder.

Repeat this process until the remainder is 0. When the remainder is 0, all digits will have been generated.

Table B-1. Octal Conversion Chart

	8	8	8	8	8	8
0	0	0	0	0	0	0
1	32,768	4,096	512	64	8	1
2	65,536	8,192	1,024	128	16	2
3	98,304	12,228	1,536	192	24	3
4	131,072	16,384	2,048	256	32	4
5	163,840	20,480	2,560	320	40	5
6	196,608	24,576	3,072	384	48	6
7	229,376	28,672	3,584	448	56	7
	16	16	16	16	16	16
0	0	0	0	0	0	0
1	1,048,576	65,536	4,096	256	16	1
2	2,097,152	131,072	8,192	512	32	2
3	3,145,728	196,608	12,288	768	48	3
4	4,194,304	262,144	16,384	1,024	64	4
5	5,242,880	327,680	20,480	1,280	80	5
6	6,291,456	393,216	24,576	1,536	96	6
7	7,340,032	458,752	28,672	1,792	112	7
8	8,338,608	524,288	32,768	2,048	128	8
9	9,437,184	589,824	36,864	2,304	144	9
A	10,485,760	655,360	40,960	2,560	160	10
В	11,534,336	720,896	45,056	2,816	176	11
C	12,582,912	786,432	49,152	3,072	192	12
ם	13,631,488	851,968	53,248	3,328	208	13
E	14,680,064	917,504	57,344	3,584	224	14
F	15,728,640	983,040	61,440	3,840	240	15

APPENDIX C ASCII and EBCDIC Character Codes

Table C-1. ASCII Character Codes

			Key	
Decimal	Octal	Hex	Symbol	Mnemonic
0	000	00	† @	NUL
j 1	001	01	† A	SOH
2	002	02	↑ B	STX
3	003	03	↑ C	ETX
4	004	04	↑D	EOT
5	005	05	↑ E	EN
6	006	06	↑ F	ACK
7	007	07	∱G	BEL
8	010	08	↑ H	BS (BACKSPACE)
9	011	09	† 1	TAB
10	012	0A	∱J	NEW LINE
11	013	0B	∱K	VT (VERT. TAB)
12	014	0C	∱∟	FORM FEED
13	015	0D	∱M	CARRIAGE RETURN
14	016	0E	ŤN	SO
15	017	0F	†o	SI
16	020	10	PP	DLE
17	021	11	Ìn∳Q	DC1
18	022	12	∱R	DC2
19	023	13	∱s	DC3
20	024	14	∳ ⊤	DC4
21	025	15	ŤU	NAK
22	026	16	ÌV	SYN
23	027	17	Ť₩	ETB
24	030	18	ŤΧ	CAN
25	031	19	ŤΥ	EM
26	032	1A	∱Ζ	SUB
27	033	1B	↑ ESC	ESCAPE
28	034	1C	↑ \	FS
29	035	1D	†]	GS
30	036	1E	 	RS
31	037	1F	† _	US
32	040	20	SPACE	
33	041	21	!	
34	042	22	" (QUOTES)	
35	043	23	#	!
36	044	24	\$	
37	045	25	%	
38	046	26	&	
39	047	27	' (APOS)	
40	050	28	(
41	051	29)	
42	052	2A	*	
43	053	2B	+	
44	054	2C	, (COMMA)	
45	055	2D	-	
46	056	2E	. (PERIOD)	
47	057	2F	/	
48	060	30	0	
49	061	31	1	

Table C-1. ASCII Character Codes (continued)

			V.	
Decimal	Octal	Hex	Key Symbol	Mnemonic
50	062	32	2	
51	063	33	3	
52	064	34	4	
53	065	35	5	
54	066	36	6	
55	067	37	7	
56	070	38	8	
57	071	39	9	
58	072	3A	:	
59	073	3B	; <	
60	074	3C		
61	075	3D	=	
62	076	3E	>	
63	077	3F	?	
64	100	40	@	
65	101	41	Α	
66	102	42	В	
67	103	43	С	
68	104	44	D	
69	105	45	E	
70	106	46	F	
71	107	47	G	
72	110	48	Н	
73	111	49	4	
74	112	4A	j	
75	113	4B	К	
76	114	4C	L	
77	115	4D	M	
78	116	4E	N	
79	117	4F	0	
80	120	50	P	
81	121	51	Q	
82	122	52	R	
83	123	53	S	
84	124	54	T	
85	125	55	U	
86	126	56	V	
87	127	57 50	W	
88	130	58	X	
89	131	59	Y	
90	132	5A	Z	
91 92	133	5B	[
93	134 135	5C	\	
		5D] • OP •	
94 95	136 137	5E	↑ OR ^	
96	137	5F 60	← OR (-) ` (GRAVE)	
97	140	61	(• –)	
98	141	62	a	
99	142	63	b	
100	143	64	c d	
101	144	65	e e	
102	145	66	f	
103	147	67		
104	150	68	g h	
105	150	69	i	
106	152	6A		
107	153	6B	j k	
108	154	6C	, I	
109	155	6D	m	
	I 155		111	

Table C-1. ASCII Character Codes (continued)

Decimal	Octal	Hex	Key Symbol	Mnemonic
110	156	6E	n	
111	157	6F	0	
112	160	70	р	
113	161	71	q	
114	162	72	r	
115	163	73	S	
116	164	74	t	
117	165	75	u	
118	166	76	V	
119	167	77	w	
120	170	78	X	
121	171	79	у	1
122	172	7A	Z	
123	173	7B	{	
124	174	7C	1	
125	175	7D	}	
126	176	7E	~ (TILDE)	
127	177	7F	DEL (RUBOUT)	

Table C-2. EBCDIC Character Codes

			T
		EBCDIC Graphic or	
		Control	
Decimal	Hex	Characters	Explanation
0	00	NUL	Null
1	01	SOH	Start of Heading
2	02	STX	Start of Text
3	03	ETX	End of Text
4	04	PF	Punch Off
5	05	HT	Horizontal Tab
6	06	LC	Lower Case
7	07	DEL	Delete
8	08		
9	09	0.41.4	
10	0A	SMM	Start of Manual Message
11 12	0B	VT	Vertical Tab
13	0C	FF CD	Form Feed
13	0D 0E	CR SO	Carriage Return
15	0E 0F	SO SI	Shift Out Shift In
16	10	DLE	
17	11	DC1	Data Link Escape Device Control 1
18	12	DC1 DC2	Device Control 1 Device Control 2
19	13	TM	Tape Mark
20	14	RES	Restore
21	15	NL NL	New Line
22	16	BS	Back Space
23	17	IL	Idle
24	18	CAN	Cancel
25	19	EM	End of Medium
26	1A	CC	Cursor Control
27	1B	CU1	Customer Use 1
28	1C	IFS	Interchange File Separator
29	1D	IGS	Interchange Group Separator
30	1E	IRS	Interchange Record Separator
31	1F	IUS	Interchange Unit Separator
32	20	DS	Digit Select
33	21	SOS	Start of Significance
34	22	FS	Field Separator
35	23		
36	24	BYP	Bypass
37	25	LF_	Line Feed
38	26	ETB	End of Transmission Block
39	27	ESC	Escape
40	28		
41 42	29 24	CNA	Sat Made
42 43	2A 2B	SM CU2	Set Mode
43 44	2B 2C	CU2	Customer Use 2
44 45	2C 2D	ENQ	Enquiry
45 46	2D 2E	ACK	Enquiry Acknowledge
46 47	2E 2F	BEL	Acknowledge Bell
48	30	DEL	Dell
49	31		
50	32	SYN	Synchronous Idle
51	33	O I I	Synomonous rais
52	34	PN	Punch On
53	35	RS	Reader Stop
54	36	UC	Upper Case
55	37	EOT	End of Transmission
56	38		
57	39		

 Table C-2.
 EBCDIC Character Codes (continued)

	<u> </u>	LBCDIC.	
		EBCDIC Graphic or	
İ		Control	
Decimal	Hex	Characters	Explanation
58	ЗА		
59	3B	CU3	Customer Use 3
60	3C	DC4	Device Control 4
61	3D	NAK	Negative Acknowledge
62 63	3E 3F	SUB	Substitute
64	40	SP SP	Space
65	41	0'	Opace
66	42		
67	43		
68	44		
69	45		
70 71	46 47		
71 72	48		
73	49		
74	4A	¢	
75	4B		
76	4C	<	
77	4D	(
78 79	4E 4F	+	
80	50	\ &	
81	51	u u	
82	52		
83	53		
84	54		
85	55		
86 87	56 57		
88	57 58		
89	59		
90	5 A	!	
91	5B	\$ *	
92	5C	*	
93	5D)	
94 95	5E 5F	_,	
96	60		
97	61	/	
98	62		
99	63		
100	64 65		
101 102	65 66		
103	67		
104	68		
105	69		
106	6A	ŀ	
107	6B	, %	
108 109	6C 6D		
110	6E	- > ?	
111	6F	?	
112	70		
113	71		
114	72 70		
115	73		

Table C-2. EBCDIC Character Codes (continued)

	Table C-2. Ebeble Character Codes (commune)					
		EBCDIC Graphic or Control				
Decimal	Hex	Characters	Explanation			
116	74					
117	75					
118	76					
119	77					
120	78 70					
121 122	79 7A	:				
123	7B	#				
124	7C	<i>"</i> @				
125	7D					
126	7E	=				
127	7F	,				
128	80					
129	81	a	1			
130	82	b				
131	83	C				
132	84	d	ļ			
133	85 86	e f				
134 135	86 87					
136	88	g h				
137	89	i				
138	8A		•			
139	8B					
140	8C					
141	8D					
142	8E					
143	8F					
144	90	:				
145	91 92	j k				
146 147	92	, <u>, , , , , , , , , , , , , , , , , , </u>				
148	94	m				
149	95	n				
150	96	0				
151	97	р				
152	98	q				
153	99	r				
154	9A					
155	9B					
156	9C 9D		'			
157 158	9D 9E					
159	9F					
160	ÃO					
161	A1	~				
162	A2	s				
163	A3	t				
164	A4	u				
165	A5	V				
166	A6	w				
167	A7 A8	X V				
168 169	A8 A9	y z				
170	AA					
170	AB					
172	AC					
173	AD					
l	1					

Table C-2. EBCDIC Character Codes (continued)

	· · · · · · · · · · · · · · · · · · ·	EBCDIC	
		Graphic or	
		Control	
Decimal	Hex	Characters	Explanation
174 175	AE AF		
176	B0		
177	B1		
178	B2		
179 180	B3 B4		
181	B5		
182 183	B6		
184	B7 B8		
185	B9		
186 187	BA BB		
188	BC		
189	BD		
190 191	BE BF		
192	C0	{	
193	C1	Α	
194 195	C2 C3	B C	
196	C4	D	
197	C5	E	
198 199	C6 C7	F G	
200	C8	Н	
201 202	C9 CA	l	
203	CB		
204	cc	Ţ	
205 206	CD CE	4	
207	CF	'	
208	D0	}	
209 210	D1 D2	J K	
211	D3	L	
212	D4	M	
213 214	D5 D6	N O	
215	D7	Р	
216 217	D8 D9	Q R	
218	DA	n	
219	DB		
220 221	DC DD		
222	DE		
223	DF		
224 225	E0 E1		
226	E2	s	
227	E3	T	
228 229	E4 E5	U V	
230	E6	W	İ
231	E7	X	

Table C-2. EBCDIC Character Codes (continued)

Decimal	Hex	EBCDIC Graphic or Control Characters	Explanation
232	E8	Υ	
233	E9	Z	
234	EA		
235	EB		
236	EC		
237	ED		
238	EE		
239	EF		
240	F0	0	
241	F1	1	
242	F2	2	
243	F3	3	
244	F4	4	
245	F5	5	
246	F6	6	
247	F7	7	
248	F8	8	
249	F9	9	
250	FA	LVM	
251	FB		
252	FC		
253	FD		
254	FE		
255	FF	EO	

INDEX

Note: Boldfaced page numbers (e.g., 1-5) indicate definitions of terms or other key	Command 1-6 completion status 4-15
information.	failed bit 4-12 , 4-11
2901 timeout bit 4-9	status 4-13 Command registers 2-2 , 3-5 A 3-9
Address	B 3-10
conventions 2-20	C 3-10
cylinder 2-10, 3-45	contents 3-7, A-5
device 2-11	summary A-5
link 2-10, 4-17	Control block (CB) 1-6, 2-5
logical sector offset 3-46	commands 3-42, 4-19
page number list 2-11, 4-17	device address 2-14
sector 2-11	Done bit 4-8
transfer 2-11, 4-18	error status 4-9
Alternate ports bit 2-17, 2-19	extended status 2-14
ASCII character codes C-1	format 2-6
Asynchronous interrupt 2-24, 3-15, 4-15	hard execution error 4-4, 4-16
status register A 3-18	hard execution error bit 4-8
status register B 3-19	interpretation error 4-4
status register C 3-20	link address 2-8, 4-17
	lists 2-5
Begin (002) PIO command 3-40	page number list address 2-8, 4-17
BMC	range error 4-17
error bit 4-10 , 4-21	soft execution error 4-2, 4-16
timeout bit 4-10	soft execution error bit 4-8
BN (Busy/nonzero), see Test conditions	status word 2-13, 4-6 , 4-8
Buffer overflow (Data Late) 4-9	status word (summary) A-14
Bus fault, bit 4-12	status word not 0 4-19
Busy	summary A-10
bit 4-11	termination bit 4-8
flag 2-3 , 2-4	termination error 4-5, 4-16
state 4-1	transfer address 2-8, 4-18
BZ (Busy/zero), see Test conditions	transfer count 2-10, 4-18
DZ (Busy/Zero), see Test conditions	unit number 2-9, 4-17
C pulse 2-4	unreadable 4-18
Cancel List (123) PIO command 3-32	unwritable 4-18
CB, see Control block	Control commands, see PIO commands
CB command 3-1, 3-42	Control function (f), see Pulses
illegal 4-17	Controller 1-3
read 3-43	buffer overflow bit 4-9
summary A-4	detected command checksum error
write 3-47	bit 4-9
CB interpretation error, see Interpretation	fetching range 4-19
error Character codes	in error recovery bit 4-21 retries 2-15
ASCII C-1	
EBCDIC C-4	specifications A-1 states 4-1
Checksum, disk sector header 3-46	siales 4-1
Checksum, disk sector header 3-40	

Controller information block 2-14		Disk subsystem, cont.
get command 3-23		overview 1-1
set command 3-26		programming 1-11
Controller panic, interrupt		software 1-4
(asynchronous) 4-16		Disk subsystems 1-1
Controller registers 2-1		DMA over the BMC, see Mapping options
command 2-2		DMA over the data channel, see Mapping
status 2-2		options
Controller/drive, data bus parity error		DN (Done/nonzero), see Test conditions
bit 4-10		DOA instruction 2-1, 3-1, 3-9
Count codes, see Legal count codes		DOAS instruction 2-4
Current CB		DOB instruction 2-1, 3-1, 3-10
maximum number 2-13		DOBS instruction 2-4
optimization 2-18		DOC instruction 2-1, 3-1, 3-10
Cylinder		DOCS instruction 2-4
address 3-46		Done
number of 2-18		flag 2-3 , 2-4
physical 2-6, 2-10		state 4-1
F-1,1-1-1		Drive 1-2
Data		error bit 4-10
bytes per sector 2-17		interface fault bit 4-9
checksum error bit 4-10		specifications A-1
structures 1-8		_
Data In A, see DIA instruction		Dual port operation release 2-18
Data In B, see DIB instruction		
Data In C, see DIC instruction		reserve 2-18
Data Late, see Buffer overflow		trespass 2-19
Data Out A, see DOA instruction		Dual-port operation 2-18
Data Out B, see DOB instruction		DZ (Done/zero), see Test conditions
Data Out C, see DOC instruction		EDCDIC abancator and a C A
Device		EBCDIC character codes C-4
address 2-8		ECC
type 2-13		correction failed bit 4-8
DIA instruction 2-1, 3-1, 3-12		correction needed bit 4-8
asynchronous interrupt 3-15		detected error bit 4-10
synchronous interrupt 3-17		Ending memory address bit 4-10
Diagnostic, tests 4-21		Ending memory address error bit 4-10, 4-21
Diagnostic Mode (024) PIO command	3-37	Error
DIB instruction 2-1, 3-1, 3-12	J-J7	status 4-9
asynchronous interrupt 3-15		status word 2-9
synchronous interrupt 3-17		Errors 4-1
DIC instruction 2-1, 3-1, 3-12		CB execution 4-2
asynchronous interrupt 3-15		CB interpretation 4-4
synchronous interrupt 3-17		CB termination 4-5
_		PIO command execution 4-6
Disk		PIO illegal command 4-6
configuration 2-17		Extended status 2-20
drives 1-2		Extended status commands, see PIO
error code 2-10		commands
Disk subsystem		Extended status information block 1-7,
commands 1-6		2-20
controller 1-3		
data structures 1-4		Fault, bits 4-12
drives 1-2		First map slot assigned, see Mapping
hardware 1-1		options
interface 1-3		

Flag	status word not 0 4-17
Busy 2-3	Interrupt 4-1
Done 2-3	asynchronous 2-24, 3-15, 4-15
Interrupt on Not Busy 2-4	bit 2-8
	on Not Busy flag 2-4
Get PIO commands 2-12 , 3-22	on soft CB errors bit 2-17
Controller Information (033) 3-23	synchronous 2-24, 3-15, 3-17
Get Extended Status 0 (040) 3-30	timeout bit 4-9
Get Extended Status 1 (041) 3-30	IORST state 4-1
Get Extended Status 2 (042) 3-30	
Get Extended Status 3 (043) 3-31	Legal count codes 2-13
Interface Information (031) 3-24	Link address 2-8
List Status (133) 3-32	illegal 4-17
Mapping Information (027) 3-24	List control commands, see PIO commands
Unit Information (035) 3-25	List is active bit 4-21
Cint information (000) 0 20	Logic fault, bit 4-11
Hard, CB execution error 4-16	Logical
Head	number of blocks 2-17
final CB error address 2-10	sector offset address 3-46
number of 2-18	sector offset address 3-40
	Man alet lead interpunta and Manning
Header, noncompare bit 4-10 Hexadecimal conversion B-1	Map slot load interrupts, see Mapping
	options
Highest unit number 2-13	Mapping
Host/Controller interface 2-1	bit 2-8
711/2 17 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	conventions 2-22
I bit set (asynchronous interrupt) 4-16	Mapping options 2-22
Information block 1-7, 2-12	Get command 3-24
controller 2-14	summary A-19
extended status 1-7, 2-20	Maximum
extended status (summary) A-13	number of current CBs 2-13
interface 2-12	return block size 2-14
summary A-10	Microcode revision number 2-13
unit 2-16	Modified bit 2-17
user option 1-7	
user option (summary) A-12	NIOC instruction 2-4
Initialization commands, see PIO	No errors (asynchronous interrupt) 4-16
commands	No Operation (000) CB command 3-42
Instruction 3-1	No retries bit 2-8
summary A-2	Null interrupt (asynchronous) 4-16
Interface information block 2-12	Number of
get command 3-24	cylinders 2-18
set command 3-26	data bytes per sector 2-17
Internal BMC physical remapping, see	heads 2-18
Mapping options	logical blocks 2-17
Internal diagnostic tests 4-21	map slots assigned 2-22
Interpretation error	sectors per cylinder 2-17
bit 4-8	•
CB range error 4-17	Octal conversion B-1
illegal CB command 4-17	Operation code 2-8
illegal link address 4-17	Optimization 2-18
illegal page number list address 4-17	bit 2-17, 2-19
illegal transfer address 4-18	•
illegal transfer count 4-18	
illegal unit number 4-17	
-	

P pulse 2-4	status (synchronous interrupt
Page number list	summary) A-8
address 2-8, 4-17	Release, see Dual port operation
entries 2-14	Reserve, see Dual port operation
format bit 2-14	Reset (777) PIO command 3-38
mask 2-14	Reset state 4-1
PIO command 3-1, 3-21	Restart (116) PIO command 3-33
arguments 3-1	Retries 2-21, 2-15
code 4-3	Retries performed 2-10
control 3-22	Retry strategy 2-15
execution error 4-6	Return transfer count 2-10
extended status 3-29	
first argument 3-9	S bit, see Interrupt on soft CB errors bit
initialization 3-22	S pulse 2-4
list control 3-31	Sector
second argument 3-10	logical offset address 3-46
sending 3-5	number of data bytes 2-17
state independent 3-37	number of data bytes 2-17 number per cylinder 2-18
status word (summary) A-16	physical address 2-10
status words 4-19	Sending a PIO command 3-5
summary A-4	Servo, data fault bit 4-11
valid while reset 3-40	Set PIO commands 2-12 , 3-22
PIO illegal command error 4-6	Controller Information (032) 3-26
Port operation 2-18	Interface Information (032) 3-26
Port reserve bits 4-11	Mapping Information (026) 3-27
Positioner, fault bit 4-12	Unit Information (034) 3-28
Power, fault bit 4-11	Soft CB execution error 4-16
Power fail	Soft error, S bit set (asynchronous
bit 4-11	interrupt) 4-16
recovery 4-13	Specifications
Power-up state 4-1	controller A-1
Priorities, CB 3-35	drive A-1
Program Load (000) PIO command 3-41	Specify
Programmed I/O command, see PIO	transfer count 2-9
command	user options 2-12
Programming, summaries A-1	Standard return format 3-22
Pulses 2-4	Start List (100) PIO command 3-35
1 uises 2-4	Start List — High Priority (103) PIO
Read CB commands	command 3-36
Read (200) 3-44	State independent commands, see PIO
Read Headers (220) 3-45	commands
Read Raw Data (210) 3-45	States 4-1
Read with Modified Bit Map (242) 3-46	
Read/Verify (201) 3-44	Status registers 2-2 , 3-12 A 3-18
Read/Verify Single Word (205) 3-44	asynchronous interrupt 3-15, 4-15
Read retries 2-15	B 3-19
Read/write, fault bit 4-12	C 3-20
Ready	
bit 4-11	command completion status 4-15
state 4-1	command status 4-13
Recalibrate Disk (400) CB command 3-43	status word (summary) A-14
Registers (400) CB command 3-43	status words 4-13
command (summary) A-5	summary (asynchronous interrupts) A-7
status (asynchronous interrupt	summary (synchronous interrupts) A-8
summary) A-7	synchronous interrupt 3-17, A-9
cammuy, 11,	

Status word 4-1, 4-7	Transfer count
asynchronous interrupt 4-15	final CB error 2-10
Begin command 4-19	illegal 4-18
CB 2-6	return 2-10
CB error 2-6	specify 2-9
CB error status 4-9	Trespass, see Dual port operation
CB status 4-7	Trespass (132) PIO command 3-36
command completion status 4-15	
command status 4-13	Unit information block 2-16
control block (CB) 4-7	get command 3-25
Get and Set commands 4-21	set command 3-26
Get List Status command 4-21	Unit number 4-11
not 0 (asynchronous interrupt) 4-17	CB 2-9
PIO command 4-19	highest 2-13
Program Load command 4-19	illegal 4-17
reset 4-21	unit information block 2-16
status register 4-13	Unit retries 2-15
summaries A-14	Unit status
unit 2-9	change interrupt mask 2-13
Summaries	word 2-9
CB commands A-4	Unit Status (131) PIO command 3-28
command registers A-5	Unreadable CB (asynchronous interrupt
control block A-10	4-18
information blocks A-10	Unwritable CB (asynchronous
instructions A-2	interrupt) 4-18
PIO commands A-4	Upstream BMC loading, see Mapping
programming A-1	options
status registers (asynchronous	Upstream loading with host page tables, see
interrupts) A-7	Mapping options
status registers (synchronous	User option information block 1-7
interrupts) A-8	User options 2-12
status words A-14	
Synchronous interrupt 2-24, 3-15, 3-17	Valid while reset commands, see PIO
status register A 3-18	commands
status register B 3-19	Verify error bit 4-10
status register C 3-20	
Sysgen (025) PIO command 3-38	Waiting for a restart bit 4-21
	Write CB commands
Test	Write (100) 3-47
conditions 2-4	Write Single Word (104) 3-48
internal diagnostics 4-21	Write with Modified Bit Map (142) 3-49
Timeouts have occurred bit 4-21	Write/Verify (101) 3-47
Transfer address 2-8	Write/Verify Single Word (105) 3-48
illegal 4-18	

◆DataGeneral users CFOUP Installation Membership Form

Name		Position			Date	
Company, Organization or Schoo		,				
Address		City		State	Zip	
Telephone: Area Code	No		Ext			
1. Account Category	□ OEM □ End User □ System Ho □ Governme			ode of Operation	□ Batch (C □ Batch (V □ On-Line	
2. Hardware M/600	Qty. Installed C	Qty. On Order	6. Co	mmunications	☐ RSTCP ☐ HASP ☐ RJE80	□ CAM □ 4025 □ Other
C/350, C/330, C/300 S/250, S/230, S/200 S/130 AP/130 CS Series					□ SAM Specify	
N3/D Other NOVA microNOVA				oplication Description	0	
Other (Specify)						
3. Software	□ DOS	□ RDOS □ RTOS □ Other	8. P	urchase	From whom was y purchased? Data General Other	
4. Languages	□ Algol □ DG/L	☐ Assembler ☐ Interactive	9. U	sers Group	Specify	
	□ Cobol □ ECLIPSE Cobol □ Business BASIC □ BASIC	☐ Fortran ☐ RPG II ☐ PL/1 ☐ Other			special interest or Data General Use	

FOLD

FOLD

STAPLE

STAPLE

FOLD

FOLD

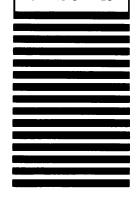


FIRST CLASS PERMIT NO. 26 SOUTHBORO, MA. 01772

Postage will be paid by addressee:



ATTN: Users Group Coordinator 4400 Computer Drive Westboro, MA 01581 NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES



◆DataGeneral

Engineering Publications Comment Form

Please help us improve our future publications by answering the questions below. Use the space provided for your comments.

Title:	
	Document No.

Yes No)	O You (can,cannot) find things easily.	Other:
	Is this manual easy to read?	O Language (is,is not) appropriate.	
		O Technical terms (are, are not) defined as needed.	
		O Learning to use the equipment	O To instruct a class.
	In what ways do you find this manual useful?	O As a reference	Other:
		As an introduction to the product	
	Do the illustration but to a	O Visuals (are, are not) well designed.	
	Do the illustrations help you?	O Labels and captions (are, are not) cle	ear.
		O Other:	
- -	Does the manual tell you all you need to know	?	
	What additional information would you like?		
		-	
	ls the information accurate?		
	(If not please specify with page number and paragraph.)	1982) 2016; 2018; 2018;	
		기계 (2) 기계 (4분) 기계 (4분)	
		auf Kari Carpoliti U No u	
	 Albani, de la periode da la libraria de la travalla de la libraria. Albani, de la libraria de la periode de la libraria del libraria de la libraria de la libraria del libraria de la libraria del libraria de la libraria de la libraria del libraria		
Name: _		Tul	
Company	ny:	Division:	
	s:		
		Telephone:	Date:

FOLD FOLD

STAPLE STAPLE

FOLD FOLD



BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 26 SOUTHBORO, MA. 01772

Postage will be paid by addressee



ATTN: ENGINEERING PUBLICATIONS

2706 Montopolis Drive Austin, TX 78741 NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

