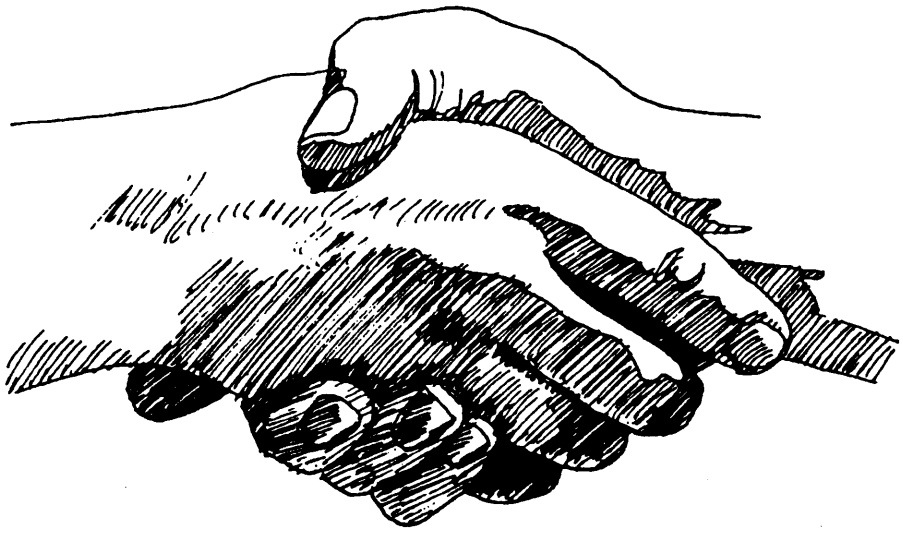
Xylogics

Model 753
User's Manual



Your Partner For Performance

166-753-001
Revision B
May 21, 1988

753 Revision Level History

Revision	Description
A (3/1/88)	Initial release
A1 (3/15/88)	Corrected board dimensions in Sections 1.0 and 1.3.
B (5/21/88)	Added Section 2.5 (Jumper Block JG); all subsequent subsections increment by one. Added Command Optimization (COP) to Byte 0A (Bit 6) of the Controller Parameters IOPB. Added definitions of Unit/Command Look Ahead (UCLA) and Elevator Seeks to Section 1. Added discussion of UCLA to Section 9.3.3 (Read Ahead). Added Section 8.8 (Command Optimization); all subsequent subsections increment by one. Updated style of tables and charts. Corrected pin numbering in Section 10.2. Removed references to Black Hole Transfers, and variable sector size.

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Table of Contents

List of Figures and Tables

Section 1: Specifications

Section 1: Specifications

1.0 Introduction

The Xylogics Model 753 disk controller connects up to four SMD-E disk drives to VMEbus™ systems using VME 6U backplanes, without requiring a daughter board or additional hardware.

1.1 Using This Manual

This manual provides two software reference cards that display the IOPB structure and codes (see inserts). Section 2 describes how to install and test the 753, Section 3 describes the 753 registers, Section 4 describes the IOPBs, and Section 5 describes the commands. Section 6 describes error processing, Section 7 is a programming tutorial, Section 8 explains the 753's special functions, Section 9 describes the theory of operation, and Section 10 includes drive interface information.

1.1.1 Abbreviations

This manual uses the following mnemonics:

AIO	Add IOPB
AIOP	AIO Pending
AM	Address Modifier
ASR	Auto-seek Retry
AUD	Auto-update
CHEN	Chain Enable
COP	Command Optimization
CRIO	Clear Remove IOPB
CRBS	Clear Register Busy
CTYP	Controller Type
DFLT	Drive Fault
DMA	Direct Memory Access
DRA	Disable Read Ahead
DRDY	Drive Ready
EC32	32-Bit ECC
ECC	Error Correction Code
ECCM	Error Correction Mode
EDT	Enable DMA Timeout
ERRS	Error Summary

VMEbus is a trademark of the VMEbus International Trade Association.

1.1.1 Abbreviations (continued)

FERR	Fatal Error
FIFO	First In/First Out Buffer
FIXD	Fixed/Removable Media
H	Notation for Hexadecimal Values
ICS	IOPB Checksum
IOPB	Input/Output Parameter Block
LSI	Large Scale Integration
MBS	Megabytes per Second
MM	Maintenance Mode
NPRM	Non-privileged Request Mode
ONCL	On-cylinder
OVS	Overlap Seek
PNUM	PROM Number
RBC	Retry Before Correction
RBS	Register Busy Semaphore
RIO	Remove IOPB
RMM	Register Maintenance Mode
ROR	Release On Request
SGM	Scatter/Gather Mode
SKER	Seek Error
TDT	Throttle Dead Time
TMOD	Transfer Mode
UCLA	Unit/Command Look Ahead
WRPT	Write-protect

1.2 Design Reliability

The following Xylogics features minimize the likelihood of product failure:

- Design for worst case voltage and temperature.
- Extensive evaluation testing.
- Low parts count through extensive use of custom LSI.
- All components burned-in.
- One card; resides in backplane or expansion chassis.
- Power-cycling under thermal stress during test.

1.3 Physical

Packaging -- The 753 completely resides on one printed circuit board.

Dimensions -- The 753 is a 2 by 2 Eurocard that measures 9.2-inches high by 6.3-inches deep (233.35 mm by 160 mm). It is identical in form-factor to the standard VME dual high-dual wide printed circuit board.

Shipping Weight -- 3 pounds (1.4 kg).

Front Panel -- The 753 front panel has connectors for four SMD-E/H-SMD disk drives.

Connectors -- The 753 uses a 60-pin high-density connector for the *A* cable, and four 50-pin high-density connectors for the *B* cables.

1.4 Environmental

The 753 environmental requirements are 0 through 55°C, with a maximum relative humidity of 90% (without condensation). Air flow across the board must maintain a maximum temperature differential of 7°C to prevent hot spots.

1.5 Electrical

Power -- The 753 uses 5 amperes at +5 volts DC (VDC), and 1 ampere at -12 VDC. The -5 volts for the differential transceivers is derived on-board.

Tolerance -- Voltages must be within plus or minus 5% (4.75 to 5.25; -11.4 to -12.6).

Grounding -- Common earth ground must be established between the disk drives and the CPU chassis, backplane, and expansion cabinets.

Mean Time Before Failure (MTBF) -- 25,000 hours at 55°C (estimated).

1.6 System-related Specifications

Data Buffering -- The 753 has a FIFO buffer that is 128K-bytes long. Data can be put into one end of the FIFO and simultaneously removed at the other end; there are no delays associated with filling and emptying the FIFO.

Read Ahead -- The 753 completes a given read and then reads ahead until its buffer is full; the controller satisfies future contiguous reads with data from the FIFO.

Format -- The 753 formats a specified number of tracks as per the Format command. Use the Read/Write Track Headers commands to incorporate custom interleaving schemes. Standard interleaving is 1:1; 2:1 to 16:1 interleaving is software programmable.

Media Defects -- The 753 has several methods for remapping bad blocks. One method leaves spare sectors on each track that can slip bad sectors with Read/Write Track Headers commands. An alternate method has the spare sectors on the last part of the maximum head. The 753 also remaps entire tracks. This reduces the total number of spare sectors required with minimal affect on system performance.

Read Defect Map -- The 753 can read the manufacturer's defect information directly from the unformatted disk.

Status LEDs -- The 753 implements two status LEDs. L1 indicates the controller is active; L2 indicates the on-board diagnostics did not complete successfully, or an error occurred.

On-board Diagnostics -- The 753 runs an extensive on-board diagnostic routine upon power-up or a bus reset. If an error occurs during this test, the 753 posts the failure in a special error register.

Scatter/Gather -- The 753 supports scatter/gather on Read and Write commands. The controller can gather data from up to 32 different memory locations and transfer it to the disk for use in a Write command; it can scatter the data out from the disk drive to the appropriate memory locations with a Read command. To execute a scatter/gather software issues a normal Read or Write command along with a DMA list that contains a memory address and the number of words to transfer to/from that location. The smallest granularity of scatter/gather is a 16-bit word.

Error Detection and Correction -- The 753 supports a 48-bit data ECC with a redundant header check; it optionally supports a 32-bit ECC on the header and data. Software controls automatic detection and correction.

1.6 System-related Specifications (continued)

The 48-bit ECC detects an error burst up to 28-bits long, and corrects error burst up to 14-bits long. The 32-bit ECC detects an error burst up to 22-bits long, and corrects error bursts up to 11-bits long, assuring data integrity.

Implied Seek Capability -- Data transfer instructions contain an implied seek. Data transfers cross sector, head, and cylinder boundaries as required (spiral read/write).

Overlap Seeks -- When overlap seeking is enabled, the 753 may have both drives simultaneously seeking to the appropriate cylinders.

Elevator Seeks -- When elevator seeking is enabled, the 753 reorders commands in ascending and then descending cylinder order, providing the best throughput from the disk system.

Diagnostic Support -- A comprehensive set of stand-alone diagnostics written in the programming language *C* is available.

1.7 Disk Drive-related Specifications

Physical Drive Interface -- The 753 supports the Extended SMD Interface ([SMD-E]; see the Control Data Corporation [CDC] Specification 64712402, Revision A, and Fujitsu B03P-4760-0101A).

Interface Data Rate -- The 753 currently supports a maximum disk data rate of 2.4 megabytes per second (MBS). The 753 supports this data rate at a 1:1 interleave factor. This allows continuous data transfers, crossing sector and head boundaries with no loss of disk revolutions (assuming the controller has enough bus time to transfer the data).

Mixed Data Rates -- The 753 can mix drives with different data rates (i.e., 1.2 MBS, mixed with 1.8 MBS or mixed with 2.4 MBS drives).

Number of Disk Drives -- The 753 supports up to four SMD-E disk drives.

Disk Sector Format -- The 753 sector format includes a header field separated from a data field by a splice area.

Header Format -- The header contains sector, head, cylinder address, and header ECC or a redundant header.

Cabling -- The 753 requires high density connectors at the controller end and standard flat cable connectors at the drive end. Contact Xylogics for further information.

1.8 VMEbus-related Specifications

VME Compliance Number -- IEEE P1014/D1.0.

Transfer Mode -- Direct Memory Access (DMA).

DMA Throttle Control -- Each time the 753 becomes bus master, it executes DMA transfers to or from host memory up to the maximum throttle limit or the number of words/longwords available in the FIFO buffer.

Dyna-throttle -- During a Read command, each time firmware executes the DMA scheduler, the 753 calculates the amount of data currently in the FIFO and DMAs from one to seven sectors to host memory. Dyna-throttle does not override the normal throttle and throttle dead time features that tune system bus activity.

DMA Data Transfer Rate -- The 753 has a maximum transfer rate of 18 MBS based on 30 nanoseconds (ns) memory response time (assuming longword mode transfers). Typically, the 753 transfers data at a rate of up to 10 MBS based on 200 ns memory response time (assuming longword mode transfers).

DMA Dead Time -- The 753 supports a programmable throttle dead time between throttle bursts. This dead time not only prevents the 753 from taking over the bus, but allows bus access time for other DMA devices.

Data Transfer Limit -- From 1 to 65,535 sectors with a single IOPB.

Bus Compatibility -- The 753 is compatible with the standard VMEbus (Rev. C.1).

Addressing Capability -- The 753 supports master A32 and slave A16 as per the VMEbus Specification. As a slave, the 753 responds to address modifier 02DH, and optionally to 29H (software programmable).

Data Width -- The 753 supports D16 and D32 as per the VMEbus Specification.

Release on Request -- The 753 maintains bus mastership, but releases the bus at the request of other peripheral devices (software programmable).

1.8 VMEbus-related Specifications (continued)

Release When Done -- The 753 releases the bus after each bus access (normal operation).

Bus Request Levels -- The 753 supports four bus request levels (jumper selectable).

Interrupt Priority -- Software programmable interrupt level and vector.

Controller Input/Output Parameter Block (IOPB) Length -- 30 bytes.

Controller Registers -- Seven 8-bit I/O Registers; byte or word addressable. Only eight bits respond during word access.

1.9 Software-related Specifications

Software Interface -- Host software can pass IOPBs to the 753 regardless of the controller's *Busy* status.

1.9.1 Software Interface

The software interface includes seven byte-wide registers. Four of these bytes comprise the VME Address Register, the fifth byte is the Address Modifier Register, the sixth byte is the Control and Status Register (CSR), and the last byte is the Fatal Error Register (the 753 returns the fatal error codes in this register). The CSR includes two bits that are very important to IOPB processing: *Add IOPB* (AIO) and *Remove IOPB* (RIO).

The IOPB is a block of command and status information. It includes the disk address, the bus address, and the type of operation. The software driver sets up the IOPB in memory, sends the IOPB address to the VME Address Registers, and sets AIO. After receiving the IOPB address, the 753 clears AIO, executes the IOPB and, upon completion or error, updates the IOPB status and sets RIO. The VME Address Registers point to the completed IOPB. The software driver reads the address and then clears RIO.

Software may add IOPBs to the queue, providing AIO is clear, by writing the IOPB address to the address registers and setting AIO (regardless of the 753's *Busy* status).

1.9.2 Software Drivers

Xylogics provides its customers with UNIX™ drivers for a variety of VME-based systems. Currently, Xylogics provides a full device driver, boot ROM, and diagnostic support for Sun Microsystems'™ 3/100 and 3/200 series machines. Xylogics also provides UNIX System V drivers for a variety of machines. A format utility for these drivers enables defect management and feature tuning.

1.9.3 Software Compatibility

The 753 is software compatible with the 752 (earlier version Xylogics VME disk controller).

1.10 Programmable Features

- Software controlled interrupt or polled operations.
- Software programmable DMA parameters.
- Software programmable drive size parameters.
- Software controlled transfer retry/correction.

UNIX is a trademark of AT&T Bell Laboratories.

Sun Microsystems is a trademark of Sun Microsytems, Inc.

Section 2: Installing and Testing the Controller

Section 2: Installing and Testing the 753

2.0 Introduction

Section 2 describes how to unpack, configure, install, and test your 753 controller.

2.1 Unpacking and Inspection

2.1.1 Inspect the Shipping Carton and the Controller

Inspect the carton for possible shipping damage. If you determine there is damage, do not unpack the unit. Notify Xylogics and the freight carrier immediately. If no damage is visible, carefully unpack the 753. Save the carton and other packing material for possible later use.

Inspect the 753 for any loose parts; make sure they are all firmly seated in their sockets. If any parts need reinsertion, observe proper orientation.

2.1.2 Contents

The 753 is a single printed circuit board. Optional items include a manual, cables, software on a floppy disk, 1/4-inch magnetic tape cartridge, or 1/2-inch tape reel. If any items are missing or damaged, please contact Xylogics at one of the following telephone numbers.

United States (Burlington, MA): (617) 272-8140;
United Kingdom (Milton Keynes): 44-908-569444.

2.1.3 Handling Precautions

Observing proper handling precautions minimizes the risk of damaging the 753 with electrostatic discharge. When transporting the 753, use an antistatic bag, antistatic bin, or the original shipping carton and packing material. Personnel handling the controller should observe proper grounding methods including, but not limited to, wrist bands, heel straps, and antistatic mats.

2.2 Configuring the 753

You can configure the 753 with several jumper options. The following subsections describe these options.

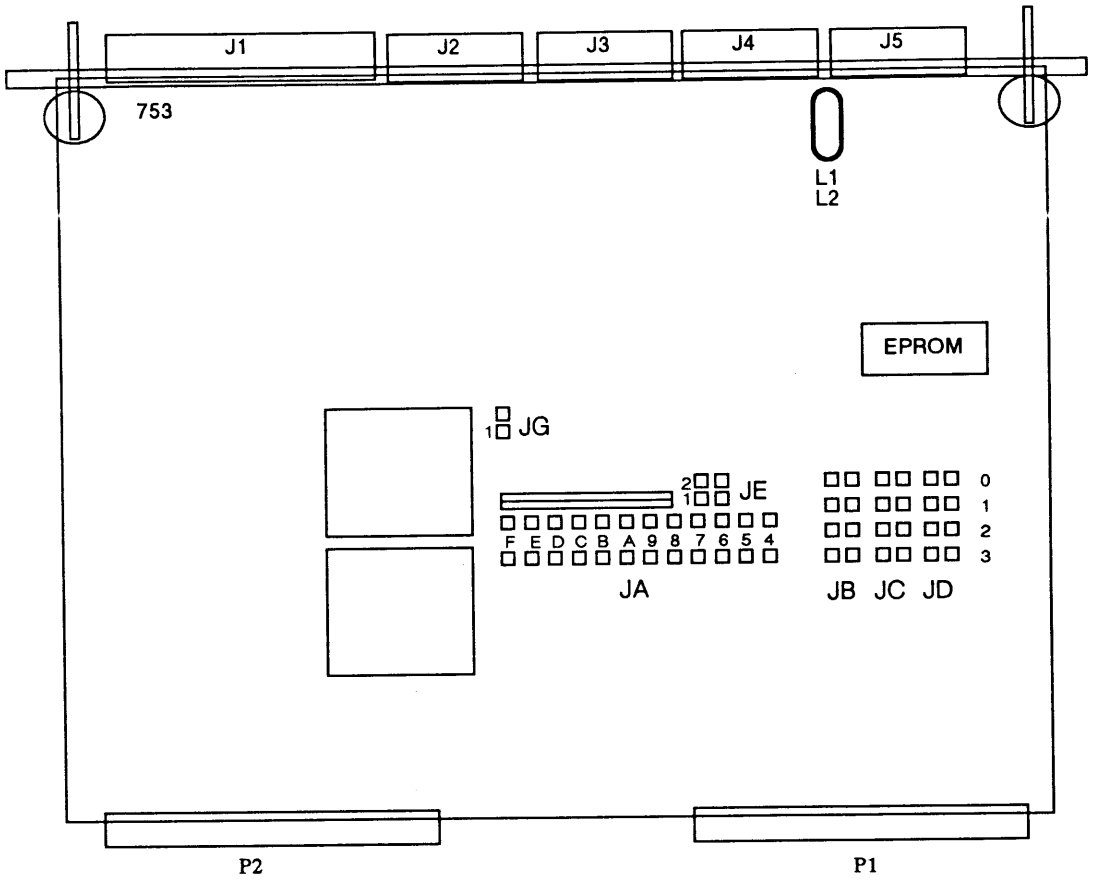


Figure 2-1. Jumper Locations

2.2.1 Base Address Selection

Jumper block JA controls the base address. Table 2–1 shows how to set the jumpers for commonly used base addresses. Inserting a jumper makes the 753 respond to a zero on that address line; removing a jumper makes the 753 respond to a one. Connect the jumper between similar pin numbers on each block. The 753 uses bits 1 through 3 to determine which register is being accessed. The 753 is an A16 slave, and responds to address modifier 02DH, and optionally to 029H.

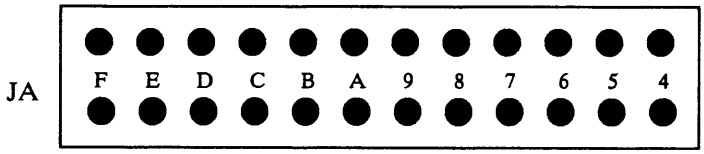


Figure 2–2. Base Address Jumper Block

Table 2–1. Base Address Selection

Screen Label:	F	E	D	C	B	A	9	8	7	6	5	4
Address:												
0100	I	I	I	I	I	I	I	O	I	I	I	I
0800	I	I	I	I	O	I	I	I	I	I	I	I
EE40*	O	O	O	I	O	O	O	I	I	O	I	I
EE80	O	O	O	I	O	O	O	I	O	I	I	I

O = Out; I = In;

*Standard Factory Configuration

2.2.2 Bus Request and Bus Grant Lines

The 753 uses the Bus Request and Bus Grant lines to become bus master. In VMEbus arbitration, there are four Bus Request/Grant levels: 0 through 3. The 753 drives one Bus Request line according to the jumper scheme you choose. The arbiter responds by driving the associated Bus Grant In line (i.e., *BG0IN* responds to *BR0*). If the 753 receives a Bus Grant on any of the Bus Grant In lines (*BG0IN* through *BG3IN*), and is not requesting the bus on that line, it passes the grant to the appropriate Bus Grant Out line: *BG0OUT* through *BG3OUT*.

Select a request level by jumpering one Bus Request (*BR0* through *BR3*), one Bus Grant In, and one Bus Grant Out line to match the selected request level. Jumper the remaining Bus Grant In/Out lines so that the incoming signal passes through the board (i.e., jumper *BGxIN* to *BGxOUT*, where *x* represents the remaining grant levels).

For example, Figure 2–3 shows the jumpering scheme for Level 3 (Figure 2–3A shows the jumper blocks as they actually appear on the board; Figure 2–3B is labeled for this example): jumper JB4 to JB8; then jumper JC4 to JC8, and JD4 to JD8. Jumper the remaining grant levels from JC5 to JD1, JC6 to JD2, and JC7 to JD3. Factory configuration: Bus Request Level 3.

> *Certain processors (i.e., Sun Microsystems) only support Bus Request Level 3.*

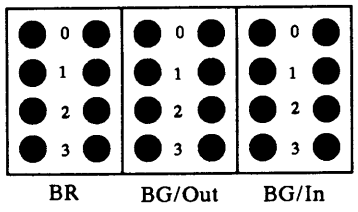


Figure 2–3A. Actual Board Layout

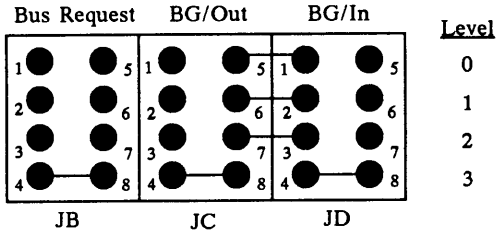


Figure 2–3B. Factory Configuration

Figure 2–3. Jumpering Bus Request and Bus Grant Levels

2.2.3 Parallel Arbitration

If you are using the 753 in parallel arbitration, and the Bus Grant Out lines must be isolated from the next slot's Bus Grant In lines, remove all jumpers between JC 5-8 and JD 1-4 (see Figure 2-3).

2.3 Maintenance Mode Lockout Jumper

Installing jumper JE 1-2 gives you unrestricted use of the maintenance mode.

When jumper JE 1-2 is removed, you may only execute the diagnostic portion of the maintenance mode. The non-diagnostic portion of the maintenance mode is proprietary to Xylogics and subject to change without notice.

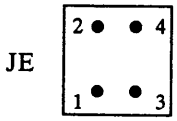


Figure 2-4. Jumper Block JE

2.4 Self Test Disable

When jumper JE 3-4 is installed, the 753 does not execute the self test on power-up.

2.5 Jumper Block JG

Jumper JG is proprietary to Xylogics. It must be *in* at all times.

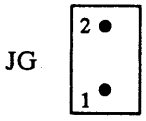


Figure 2-5. Jumper Block JG

2.6 PROMs and PALs

Table 2-2. PROM/PAL Part Numbers and Locations

Location	Part Number	Type
G15	180-002-203	EPROM
L14	181-001-049	PAL
L15	181-001-050	PAL
D5	181-001-051	PAL
E5	181-001-052	PAL

2.7 Light Emitting Diodes

The 753 has two light emitting diodes (LEDs). L1 (BSY) is the Busy LED. It is located closest to the printed circuit board; when lit, the 753 is active. L2 (ERR) is the Error LED; it straddles L1. During power-up, L2 lights for a moment and then goes off. If L2 remains on, a fatal error occurred.

2.8 Board Labels/Revision Control

All Xylogics controllers use various revision control labels. The revision level is important when discussing configuration issues. Please familiarize yourself with your board's revision before contacting us.

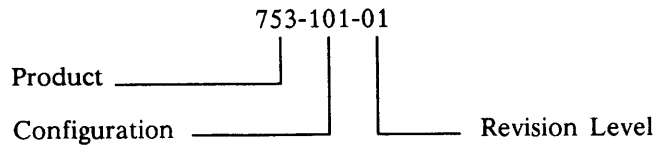


Figure 2-6. Sample Part Number

2.9 Preparing The Computer System for Installation

The backplane of your system must provide a VMEbus slot for the 753. The slot must be capable of handling a bus master, and the power source must handle the power consumption of the entire system, including the 753.

2.9.1 Backplane Jumpers

Remove any jumpers that short, or cause the Interrupt Acknowledge *(IACK IN/OUT)* and DMA Grants *(BG 0-3 IN/OUT)* to bypass the slot in which you are installing the 753.

2.9.2 Card Cage Slot

The card cage must have a slot at the proper DMA priority available for the 753. The 753 uses DMA to transfer data and IOPBs. Placement of the 753 in the DMA priority chain may be critical. The amount of bus bandwidth it uses will be high at times; this may affect other boards in the system. Likewise, other boards may not allow enough time for the 753 to DMA enough data to keep up with the disk; consider this when choosing a slot. If the 753 does not get a high enough priority, then its DMA falls behind the disk requirements, and it has to wait until the next revolution before continuing the transfer. If the 753 priority is high, it gets enough DMA time, but other boards having smaller buffers may not receive enough data within the allotted DMA time. The priorities must be balanced for your system to work properly.

2.9.3 Power Considerations

The 753 affects the power consumption of the entire computer system. The 753 uses +5 volts for logic and -5 volts, derived from the -12 volt supply, for the SMD interface's differential drivers/receivers. Make sure the power supplies can handle the entire power load. Readjust the voltages *after* plugging in the 753. A power supply that is just adequate may cause intermittent and unusual problems as it generates noise from occasional overcurrent protection.

Limits: +5 volts (4.75 to 5.25 volts) at 5 amps;
-12 volts (-11.4 to -12.6 volts) at 1 amp.

2.10 Preparing the Disk Drive for Installation

Follow the manufacturer's instructions for unpacking and inspecting the disk drive.

Configure the drive for use with the 753. This entails setting up the unit select, number of sectors per track, and ensuring the *A* cable provides the sector and index pulses. See your drive manual for more specific configuration information.

2.10.1 Drive Unit Select

A plug on the front of the drive, or switches on one of the drive's internal circuit cards, usually selects the drive unit number. The 753 accesses drives with unit numbers ranging from 0 through 7. Set the first drive to Unit 0.

2.10.2 Number of Sectors Per Track

Switches on one of the drive's internal circuit cards usually select the number of sectors per track. The 753 standard format uses a minimum of 88 bytes of overhead per sector. This is a nominal number derived from the factory-set defaults (see Section 8.2).

If you are using sector slip, the number of sectors available to the program is the total number of physical sectors on the drive less the spares (see Section 8.3).

Most disk drives have a runt sector (a very small sector at the end of the disk). The 753 requires that you format all sectors except the runt. The minimum runt size is six bytes.

2.10.3 Sector and Index Pulses

Both the *A* (Control) and *B* (Radial) cables can carry the sector and index pulses. Disk vendors usually provide drives with sector and index on the *A* cable. The 753 requires the *A* cable to carry sector and index.

2.10.4 Tags 4 and 5

Some disk drives use the spare lines (see Section 10) for maintenance functions (Tag 4). Other disk drives use the spare lines for the *Extended Cylinder* bit (bit 10). The 753 supports both options; configure the drive for its intended use.

2.10.5 Extended Cylinder Addressing

There are two methods for addressing cylinders beyond 1023. Xylogics supports the method that uses the spare lines on the *A* cable as the cylinder address (bit 10). The 753 does not support the alternate method of using the upper bits of the common interface bus and Tag 2 (Head Tag).

2.11 Initial Tests

This section relies upon your familiarity with your system's monitor and diagnostics.

2.11.1 Power-up and Self Test

The 753 initiates a self test upon power-up. The Error LED (L2) remains lit during the self test, and then goes off. If L2 remains on after the 753 passes the self test, the board is not functioning properly; the Fatal Error Register may indicate the nature of the problem. When L2 is on, *SYSFAIL* is asserted on the VMEbus. Contact Xylogics for further assistance. *Check the power supply voltages and ensure they are within limits (4.75 to 5.25 volts and -11.4 to -12.6 volts).*

2.11.2 Drive Ready

Spin the drive up and wait for *drive ready*. Issue a Read Drive Parameters command. The Drive Status byte indicates the drive status at execution time. If you cannot get the proper status, and DRDY is clear, check the drive cable connections and try again. If you are still unable to get the status, check the -12V supply on the bus. If the problem persists, check the disk drive with an off-line tester.

2.12 Diagnostics

When you run the diagnostics:

- Format the disk with either a diagnostic or format program.
- Run a full pass of your diagnostic or determine that the system is working properly.
- Cable and test any additional drives.

Section 3:
The Controller Registers

Section 3: The 753 Registers

3.0 Introduction

The 753 programming interface uses seven, one-byte long, I/O registers. The registers have one function when read, and another when written. The bus address jumpers define the register set's base address. Table 3-1 lists the registers along with the address offset from the base address. The 753 responds to either bytes or 16-bit words; when it responds to words, only 8 bits are valid.

Table 3-1. Register Offsets

Register	Offset
IOPB Address Byte 0 (Least Significant Byte)	1
IOPB Address Byte 1	3
IOPB Address Byte 2	5
IOPB Address Byte 3 (Most Significant Byte)	7
IOPB Address Modifier	9
Control and Status Register	B
Fatal Error Register	D

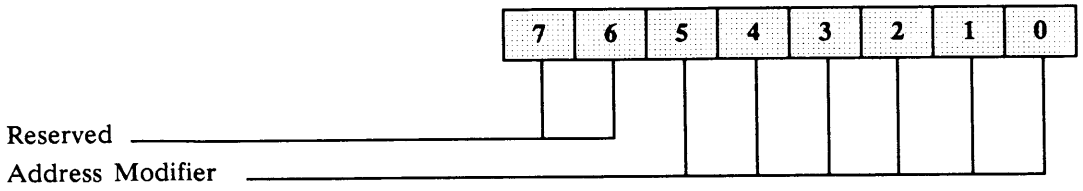
3.1 IOPB Address Registers

The first four registers define the 32-bit address of an IOPB (chain). When you write these registers, the 753 interprets the address of the IOPB (chain) ready for execution. When you read them, and the *Remove IOPB* (RIO) bit is set, the registers point to the IOPB (chain) that the 753 just completed.

The *Add IOPB* (AIO) and *Remove IOPB* (RIO) bits in the Control and Status Register define the protocol for reading and writing this address register (see Section 3.3).

3.2 IOPB Address Modifier

This register defines the IOPB address modifier. Address modifiers are used for such purposes as memory mapping, privilege levels, and addressing range (see the VMEbus Specification). Section 3.3 defines the protocol for reading and writing this register.

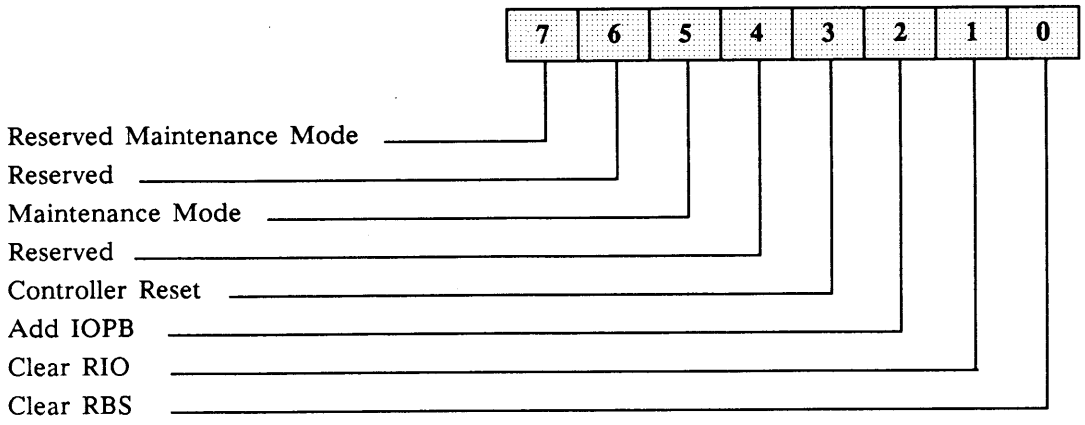


Bit	Mnemonic	Description
7-6		**Reserved.**
5-0	AM	**Address Modifier:** Most systems use the standard AM code of 3D (see the VMEbus Specification).

3.3 Control and Status Register

When written, this register provides the host with control of the 753 operation; when read, it provides the host with 753 status information. Section 3.3.1 defines the bits in this register when written; Section 3.3.2 defines the bits when read.

3.3.1 Control Register (Write)

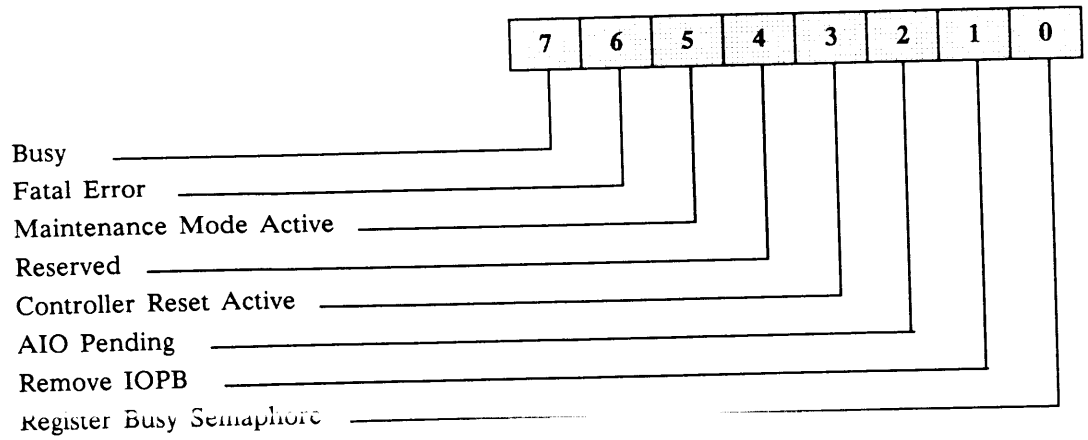


3.3.1 Control Register (Write) (continued)

Bit	Mnemonic	Description
7	RMM	**Register Maintenance Mode:** When RMM and MM are set, the 753 echoes the values previously written in all the registers, except the CSR.
6		**Reserved.**
5	MM	**Maintenance Mode:** Setting MM and AIO places the 753 in maintenance mode. Maintenance mode supports a different register protocol; use it as a diagnostic tool (see Section 8).
4		**Reserved.**
3	CRST	**Controller Reset:** CRST signals the 753 microprocessor to perform a *soft* reset; it deselects all the drives (and releases dual port), stops the DMA and disk sequencers (potentially during sector transfers), and cancels any IOPBs in the queue. After completing the *Controller Reset*, the 753 clears the CSR. CRST does not initiate a power-up self test.
2	AIO	**Add IOPB:** When set, the 753 executes the IOPB (chain) at the address pointed to by the IOPB Address and Address Modifier Registers. As soon as the host sets AIO, the 753 sets *AIO Pending* (AIOP) in the Status Register, indicating the 753 has received AIO, but has not yet processed the new chain's address. The 753 clears AIOP after internally storing the new (chain) address. The 753 can store up to 31 IOPB addresses in this manner. Reasserting AIO if AIOP is set violates the register protocol.
1	CRIO	**Clear RIO:** The host sets CRIO to clear RIO in the Status Register (SR). Typically, the host sets CRIO after reading the address of a completed IOPB (chain) from the IOPB Address and Modifier Registers. Clearing RIO enables the 753 to update the IOPB Address and Modifier Registers with the address and address modifier of a newly completed IOPB (chain). Clearing RIO, if it is not set in the SR, violates the register protocol.
0	CRBS	**Clear RBS:** The host sets CRBS to clear RBS in the Status Register. Clearing RBS releases the registers for use by another host (see Section 8.7.2). CRBS is only relevant in a multiprocessor environment.

A Controller Reset takes up to one second to complete.

3.3.2 Status Register (Read)



Bit	Mnemonic	Description
7	BUSY	**Busy:** When set, the 753 is executing IOPBs. The 753 sets *Busy* after acknowledging the first IOPB address by clearing AIOP; it clears *Busy* after completing all the IOPBs with no new ones pending (within 500 microseconds of the host clearing RIO on the last IOPB). The 753 redefines this bit in maintenance mode (see Section 8.6).
6	FERR	**Fatal Error:** When set, the 753 detected a fatal hardware error. A fatal error asserts *SYSFAIL*. Setting CRST clears FERR. The Fatal Error Register holds more specific information. The 753 sets FERR under the following conditions: (1) Maintenance Mode Test Failure; (2) Power-up Self Test Failure; (3) IOPB Checksum Miscompare; (4) IOPB DMA Fatal; (5) IOPB Address Alignment Error; (6) Firmware Error; (7) Illegal Maintenance Mode Test Number; and (8) *ACFAIL* Asserted.
5	MMA	**Maintenance Mode Active:** When set, the 753 is in maintenance mode (see Section 8.6).

3.3.2 Status Register (Read) (continued)

Bit	Mnemonic	Description
4		**Reserved.**
3	RSTA	**Controller Reset Active:** The host set CRST in the Control Register, and the 753 is resetting itself.
2	AIOP	**AIO Pending:** When set, the host set AIO in the Control Register, but the 753 has not yet acknowledged its receipt. When clear, the host can reassert AIO.
1	RIO	**Remove IOPB:** The 753 sets RIO after completing an IOPB (chain) and placing the address in the IOPB Address and Address Modifier Registers. After the host reads the address and modifier, it must clear RIO by setting CRIO in the Control Register.
0	RBS	**Register Busy Semaphore:** When set, multiple processors can share register access without colliding (see Section 8.7.2). RBS is only relevant in a multiprocessor environment.

3.4 Fatal Error Register

If a fatal error occurs, the 753 returns the appropriate completion code in this register. Table 3–2 lists the fatal error codes; Section 6.4 describes them.

3.4 Fatal Error Register (continued)

Table 3–2. Fatal Error Codes

Code	Description
E0	Reserved
E1	IRAM Self Test Failure
E2	Reserved
E3	Maintenance Test 3 Failure (DSKCEL RAM)
E4	Maintenance Test 4 Failure (Header Shift Register)
E5	Maintenance Test 5 Failure (VMEDMA Registers)
E6	Maintenance Test 6 Failure (REGCEL Chip)
E7	Reserved
E8	Maintenance Test 8 Failure (Disk FIFO)
E9-EF	Reserved
F0	IOPB Checksum Miscompare
F1	IOPB DMA Fatal
F2	IOPB Address Alignment Error
F3	Firmware Error
F5	Illegal Maintenance Mode Test Number
F6	*ACFAIL* Asserted

Section 4: IOPB Description

Section 4: IOPB Description

4.0 Introduction

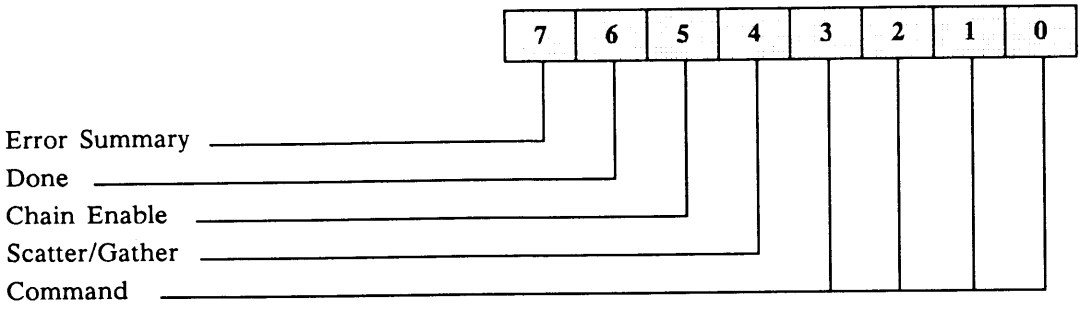
The Input/Output Parameter Block (IOPB) passes messages between the 753 and host software. Software passes the type of transfer, the disk and data addresses, and the count to the 753. The 753 returns the transfer status and, if AUD is set or an error occurs, the ending addresses upon command completion. This section begins with the standard IOPB for most data transfers and follows with variations of the IOPB.

4.1 Standard IOPB

The 753 uses the standard IOPB for data transfer and some general purpose commands.

	7	6	5	4	3	2	1	0
00	ERRS	DONE	CHEN	SGM	Command Code			
01	Completion Code							
02	0	SR	CSE	WRPT	DFLT	SKER	ONCL	DRDY
03	Internal Status							
04	Subfunction Code							
05	FIXD	0				Unit		
06	Link List Length					Interrupt Level		
07	Interrupt Vector							
08	Count High							
09	Count Low							
0A	Cylinder High							
0B	Cylinder Low							
0C	Head							
0D	Sector							
0E	0	Data or Link Address Modifier						
0F	0	Next IOPB Address Modifier						
10	Data or Link Address High							
11	Data or Link Address							
12	Data or Link Address							
13	Data or Link Address Low							
14	Next IOPB Address High							
15	Next IOPB Address							
16	Next IOPB Address							
17	Next IOPB Address Low							
18	IOPB Checksum High							
19	IOPB Checksum Low							

4.1.1 IOPB Byte 0 (Command)



Bit	Mnemonic	Description
7	ERRS	**Error Summary:** ERRS is only valid if *Done* is set. When set, a hard or soft error occurred during IOPB processing. When clear, the 753 successfully completed the IOPB.

Clear Done and ERRS before executing an IOPB.

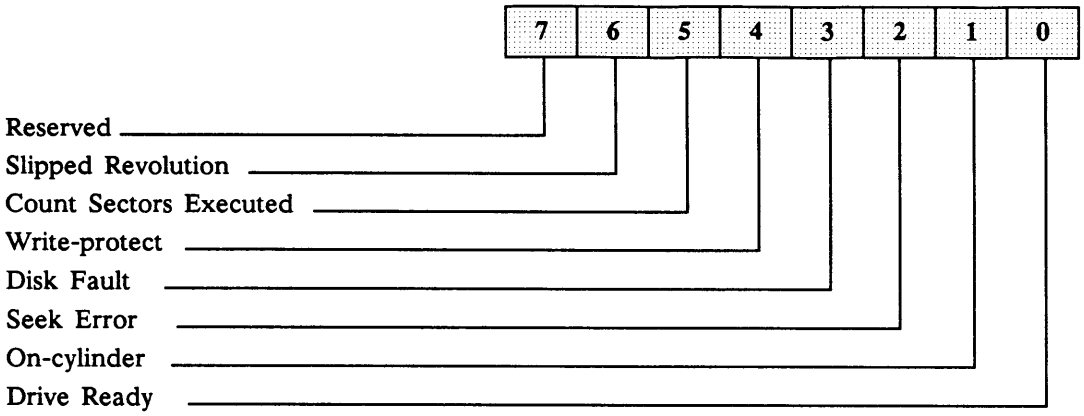
6	DONE	**Done:** When set, the IOPB is complete; if chained, host software may remove the IOPB and reuse it.
5	CHEN	**Chain Enable:** When set, the Next IOPB Address Modifier and Next IOPB Address point to the next chained IOPB. When clear, this IOPB is not chained. The 753 always returns one IOPB at a time.
4	SGM	**Scatter/Gather Mode:** When set, the IOPB is either a scatter read or a gather write transfer; a linked list describes the number of 16-bit words and to what address the 753 transfers each section of the data. The link address modifier and the link address specify the link list location. When clear, this IOPB specifies the data transfer address; the data is transferred to/from contiguous memory. SGM is only valid for standard read and write operations.
3-0	COMM	**Command:** See Table 4-2.

4.1.2 IOPB Byte 1 (Status Byte 1)

After the 753 executes the IOPB, it sets *Done* and posts a completion code in this byte. Completion codes are only valid if *Done* is set. A code of 0x indicates a successful completion; any other value indicates an error occurred (see Section 6).

4.1.3 IOPB Byte 2 (Status Byte 2)

IOPB Byte 2 is the Disk Status byte; it is only valid if *Done* is set. The 753 reads Byte 2 (excluding bits 5 and 6) from the drive that this IOPB selects.



Bit	Mnemonic	Description
7		**Reserved.**
6	SR	**Slipped Revolution:** The 753 sets SR if its DMA cannot keep up with the disk; it waits until the sector comes around on the next revolution.
5	CSE	**Count Sectors Executed:** The 753 sets CSE if the current sector count is invalid and it has to recount the sectors.
4	WRPT	**Write-protect:** The 753 sets WRPT if the selected drive is write-protected.
3	DFLT	**Disk Fault:** The 753 sets DFLT if a fault exists in the selected drive.

4.1.3 IOPB Byte 2 (Status Byte 2) (continued)

Bit	Mnemonic	Description
2	SKER	**Seek Error:** The 753 sets SKER if software specifies a cylinder higher than the drive maximum, or a seek does not complete within 500 ms.
1	ONCL	**On-cylinder:** The 753 sets ONCL when the selected drive is *on-cylinder*.
0	DRDY	**Drive Ready:** The 753 sets DRDY when the selected drive is *ready*.

4.1.4 IOPB Byte 3 (Status Byte 3)

IOPB Byte 3 is reserved. It reflects the 753's internal status.

4.1.5 IOPB Byte 4 (Subfunction)

IOPB Byte 4 is the Subfunction byte. Subfunction codes follow a convention that indicates whether the code is generic to all VME controllers, generic to a group of controllers, or specific to a particular controller (see Table 4–1).

The 753 combines standard command codes with subfunction codes to define the required operation. Table 4–2 lists the 753 command and subfunction codes.

Table 4–1. Subfunction Code Classes

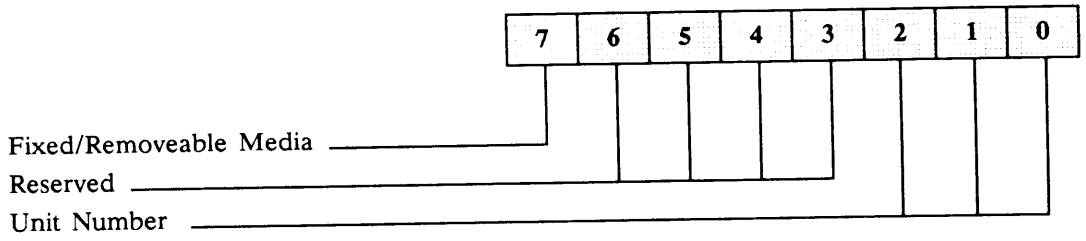
Codes (Hex)	Class
00-1F	Generic to All
20-3F	Generic Tape
40-5F	772-specific
60-7F	Reserved
80-9F	Generic Disk
A0-AF	75x-, and 7053-specific
B0-BF	71x-specific
C0-FF	Reserved

4.1.5 IOPB Byte 4 (Subfunction) (continued)

Table 4-2. Command and Subfunction Codes

Code	Command	Subfunction	Description
0	No Operation	00	No Operation
1	Write	00	Normal Write
2	Read	00	Normal Read
3	Seek	00	Report Current Address
		01	Seek and Report Current Address
		02	Start Seek and Report Comp Imm
4	Drive Reset	00	Drive Reset
5	Write Parameters	00	Write Controller Parameters
		80	Write Drive Parameters
		81	Write Format Parameters
6	Read Parameters	00	Read Controller Parameters
		80	Read Drive Parameters
		81	Read Format Parameters
		A0	Read Drive Status Extended
7	Extended Write	80	Write Track Headers
		81	Write Format
		82	Write Header, Header Verify, Data, and Data ECC
		A0	Write Defect Map
		A1	Write Defect Map Extended
8	Extended Read	80	Read Track Headers
		81	Verify Data
		82	Read Header, Header Verify, Data, and Data ECC
		A0	Read Defect Map
		A1	Read Defect Map Extended
9	Diagnostics	00	Self Test
A-F	Reserved		

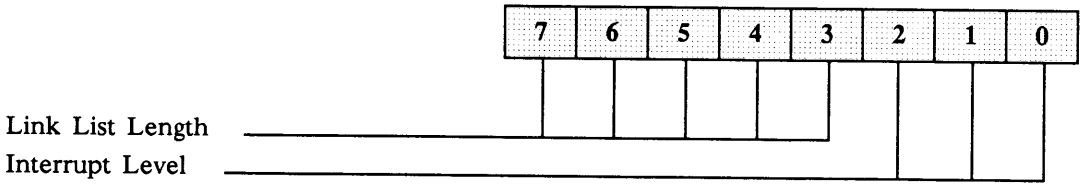
4.1.6 IOPB Byte 5 (Unit)



Bit	Mnemonic	Description
7	FIXD	**Fixed/Removable Media:** When set, the 753 is accessing the fixed media of a disk drive. When clear, the 753 is accessing the removable media of a drive. FIXD allows you to treat a fixed/removable drive as two separate disk drives (see Section 8.13).
6-3		**Reserved.**
2-0	UNIT	**Unit Number:** This value specifies the unit number (from 0 to 7) of the drive to which the transfer is directed.

Depending on the command, Bytes 6 through 13 have different definitions (see Sections 4.2 through 4.4).

4.1.7 IOPB Byte 6 (Interrupt Level)



Bit	Mnemonic	Description
7-3	LLL	**Link List Length:** These bits specify the length, in elements, of a linked list for Scatter/Gather commands. Each element refers to an 8-byte block in the linked list (see Table 8–2).
2-0	INTL	**Interrupt Level:** The 753 uses INTL as the VMEbus hardware interrupt level after completing the IOPB. The 753 will not interrupt if bits 0 through 2 are clear.

4.1.8 IOPB Byte 7 (Interrupt Vector)

IOPB Byte 7 determines the interrupt vector that the 753 uses upon command completion. This byte is not valid if the interrupt level is zero.

4.1.9 IOPB Bytes 8 and 9 (Count)

IOPB Byte 8 is Count High; Byte 9 is Count Low. These bytes specify how many sectors the 753 transfers in a given data transfer IOPB. They also specify how many tracks the 753 formats in a given Format command.

4.1.10 IOPB Bytes A and B (Cylinder)

IOPB Byte A is Cylinder High; Byte B is Cylinder Low. These bytes specify the starting cylinder address for a transfer.

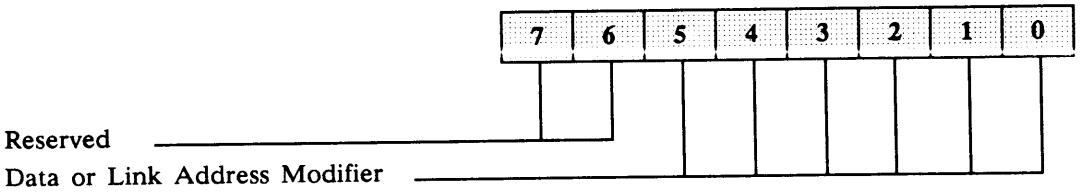
4.1.11 IOPB Byte C (Head)

IOPB Byte C specifies the starting head number for a transfer.

4.1.12 IOPB Byte D (Sector)

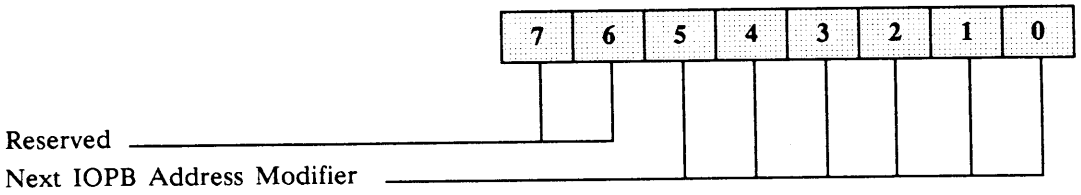
IOPB Byte D specifies the starting sector number for a transfer.

4.1.13 IOPB Byte E (Data or Link Address Modifier)



Bit	Description
7-6	**Reserved.**
5-0	**Data or Link Address Modifier:** If SGM is set, this field specifies the link list address modifier; if SGM is clear, this field specifies the data address modifier.

4.1.14 IOPB Byte F (Next IOPB Address Modifier)



4.1.14 IOPB Byte F (Next IOPB Address Modifier) (continued)

Bit	Description
7-6	**Reserved.**
5-0	**Next IOPB Address Modifier:** If CHEN is set, this field, along with the Next IOPB Address, point to the next IOPB in the chain. The 753 ignores this byte if CHEN is set.

4.1.15 IOPB Bytes 10 through 13 (DMA Data Address)

IOPB Byte 10 is DMA Data Address High; Byte 13 is DMA Data Address Low. These bytes comprise the data or link list address pointers. The 753 uses these bytes with the data or link list address modifier to point to the data or linked list address. If SGM is set, this address points to the linked list; if SGM is clear, this address points to the data address. The link list address must be on a 16-bit word boundary.

4.1.16 IOPB Bytes 14 through 17 (Next IOPB Address)

IOPB Byte 14 is Next IOPB Address High; Byte 17 is Next IOPB Address Low. These bytes comprise the Next IOPB Address pointers. The 753 uses these bytes with the Next IOPB Address modifier to point to the next IOPB in the chain if CHEN is set in Byte 0. The Next IOPB Address must be on a 16-bit word boundary.

4.1.17 IOPB Bytes 18 and 19 (IOPB Checksum)

Byte 18 is IOPB Checksum High; Byte 19 is IOPB Checksum Low. The 753 calculates the checksum by adding the IOPB bytes and comparing the value against the value software calculated and posted in these bytes (see Section 8.12).

4.1.18 IOPB Bytes 1A and 1B (ECC Pattern Word)

IOPB Byte 1A is ECC Pattern Word High; Byte 1B is ECC Pattern Word Low. These bytes are required for ECC Mode 0 and may be required for Mode 2 (see Section 6.4).

4.1.19 IOPB Bytes 1C and 1D (ECC Offset Word)

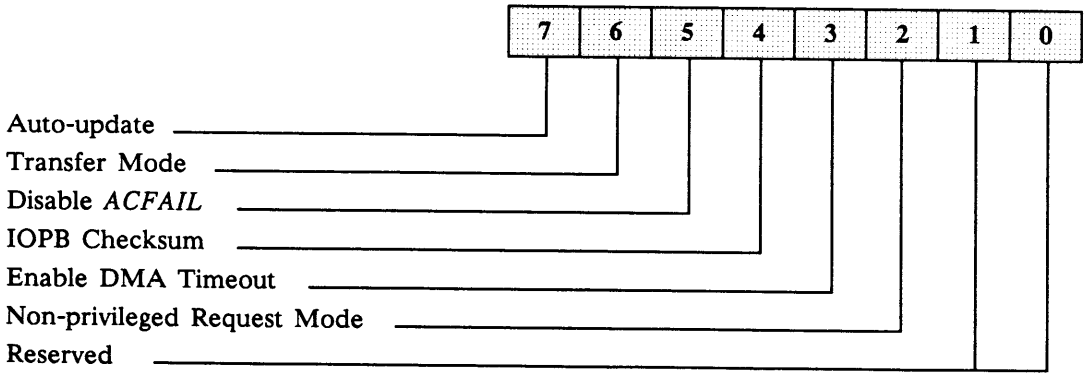
IOPB Byte 1C is ECC Offset Word High; Byte 1D is ECC Offset Word Low. These bytes are required for ECC Mode 0 and may be required for Mode 2 (see Section 6.4).

4.2 Controller Parameters IOPB

The 753 sets and reads various controller parameters with this IOPB. The 753 uses the standard IOPB, but redefines bits in Bytes 8 through E and 10 through 13.

	7	6	5	4	3	2	1	0
00	ERRS	DONE	CHEN	SGM	Command Code			
01	Completion Code							
02	0	SR	CSE	WRPT	DFLT	SKER	ONCL	DRDY
03	Internal Status							
04	Subfunction Code							
05	0					Unit		
06	0					Interrupt Level		
07	Interrupt Vector							
08	AUD	TMOD	DACF	ICS	EDT	NPRM	0	
09	TDT		0	ROR	0			DRA
0A	OVS	COP	0	ASR	0	RBC	ECCM	
0B	Throttle							
0C	EPROM Release Level							
0D	0							
0E	Controller Type							
0F	0	Next IOPB Address Modifier						
10	EPROM Part Number High							
11	EPROM Part Number Low							
12	EPROM Revision Level							
13	0							
14	Next IOPB Address High							
15	Next IOPB Address							
16	Next IOPB Address							
17	Next IOPB Address Low							
18	IOPB Checksum High							
19	IOPB Checksum Low							

4.2.1 IOPB Byte 8 (Controller Parameters A)



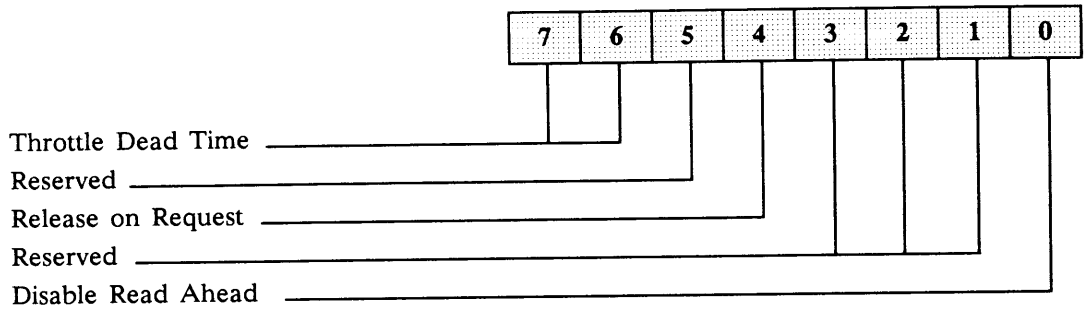
Bit	Mnemonic	Description
7	AUD	**Auto-update:** When set, the 753 updates the IOPB to the transfer's ending parameters; it updates the disk address, the sector count, and the data address after completing the transfer or detecting an error. When clear, the 753 only updates the IOPB if an error occurs. The update includes the sector in error, the correct remaining sector count, and the proper data address.
6	TMOD	**Transfer Mode:** When set, the 753 executes data transfers in longword mode. When clear, it executes transfers in word mode. IOPB transfers are always in word mode. If a transfer starts on an improper address boundary, the 753 transfers a byte and/or a word, as necessary to align boundaries, and continues the transfer in the selected mode. The 753 may end the transfer with a byte and/or a word.
5	DACF	**Disable ACFAIL:** When set, the 753 ignores the bus signal *ACFAIL*.
4	ICS	**IOPB Checksum:** When set, the 753 reads the IOPB and compares the checksum it generated during the read with the checksum the software driver appended to the IOPB. If AUD is set, the 753 also updates the Checksum bytes in any IOPB. Clearing ICS disables this feature (see Section 8.12).

> *Since this feature adds at least 100 microseconds to each transfer, it affects the 753's performance.*

4.2.1 IOPB Byte 8 (Controller Parameters A) (continued)

Bit	Mnemonic	Description
3	EDT	**Enable DMA Timeout:** When set, the 753 enables an on-board DMA bus error timer. When clear, the 753 relies on the external VMEbus transfer timer. The timeout period is one millisecond.
2	NPRM	**Non-privileged Request Mode:** When set, the 753 responds to address modifiers 02DH and 29H. When clear, the 753 only responds to 02DH (see the VMEbus Specification).
1-0		**Reserved.**

4.2.2 IOPB Byte 9 (Controller Parameters B)

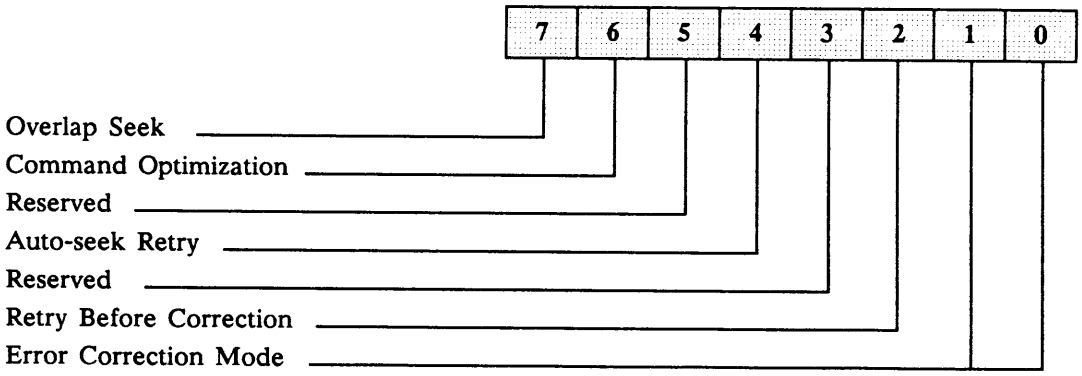


Bit	Mnemonic	Description
7-6	TDT	**Throttle Dead Time:** TDT selects one of four minimum time periods that determines the time the 753 remains off the bus between throttle bursts (see Section 8.11).
5		**Reserved.**

4.2.2 IOPB Byte 9 (Controller Parameters B) (continued)

Bit	Mnemonic	Description
4	ROR	**Release on Request:** When set, the 753 releases the bus at the request of other bus masters; otherwise, it continues with the next throttle burst. The 753 monitors the bus request lines and releases *Bus Busy* only if another bus request is pending; it completes the specified throttle burst before releasing the bus due to a pending request. When clear, the 753 releases the bus at the end of each throttle burst and rearbitrates if more data transfers are pending (see Section 8.15).
3-1		**Reserved.**
0	DRA	**Disable Read Ahead:** When set, the 753 disables its read ahead feature. When clear, the 753 satisfies all subsequent contiguous reads with data from the read ahead buffer (if possible).

4.2.3 IOPB Byte A (Controller Parameters C)



Bit	Mnemonic	Description
7	OVS	**Overlap Seek:** When set, the 753 initiates overlap seeking if more than one drive is present. When clear, the 753 does not initiate overlap seeking.

4.2.3 IOPB Byte A (Controller Parameters C) (continued)

Bit	Mnemonic	Description
6	COP	**Command Optimization:** When set, the 753 initiates elevator seeking and optimizes IOPB processing, maximizing disk access.
5		**Reserved.**
4	ASR	**Auto-seek Retry:** When set, the 753 clears the drive, seeks to the requested cylinder, and retries the transfer up to two times on any of the following errors: *seek, header error/cylinder, header error/head, drive not on-cylinder,* and *drive faulted.*
3		**Reserved.**
2	RBC	**Retry Before Correction:** When set, if an ECC error occurs, the 753 retries the operation once without calculating the error syndrome. If an error occurs on the second try, the 753 reverts to the specified error correction mode.
1-0	ECCM	**Error Correction Mode:** There are three error correction modes. Mode 0 stops a transfer and provides the driver with the error's offset and pattern. The driver performs the actual correction. Mode 1 flags an error and continues the transfer. Mode 2 performs the correction in host memory, flags a soft error, and continues the transfer.

4.2.4 IOPB Byte B (Controller Parameters D)

Bits 0 through 7 are the *Throttle* (THRO) bits. The throttle is the maximum number of transfers allowed each time the 753 becomes bus master. The throttle value determines the maximum DMA burst length for both data and IOPB DMA transfers. This byte allows a throttle setting from 1 to 256 (see Table 4-3).

4.2.4 IOPB Byte B (Controller Parameters D) (continued)

Table 4-3. Throttle Values

Value	Weight
0	256
1	1
2	2
3	3
:	:
255	255

4.2.5 IOPB Byte C (EPROM Release Level)

The 753 returns the EPROM release level on a Read Controller Parameters command.

4.2.6 IOPB Byte E (Controller Type)

IOPB Byte E is the Controller Type byte. Xylogics assigns each VME controller a unique controller type code.

Table 4-4. Controller Type Codes

Controller	Code (H)	Type
712	12	(ESDI Disk)
714	14	(ESDI Disk)
772	72	(Pertec Tape)
752	52	(SMD Disk)
753	54	(SMD Disk)

4.2.7 IOPB Bytes 10 and 11 (EPROM Part Number)

The 753 returns a portion of the EPROM part number on a Read Controller Parameters command. The four nibbles in these two bytes refer to the part number's last four digits. For example, if the part number is 180-002-173, Byte 10 is 21H and Byte 11 is 73H.

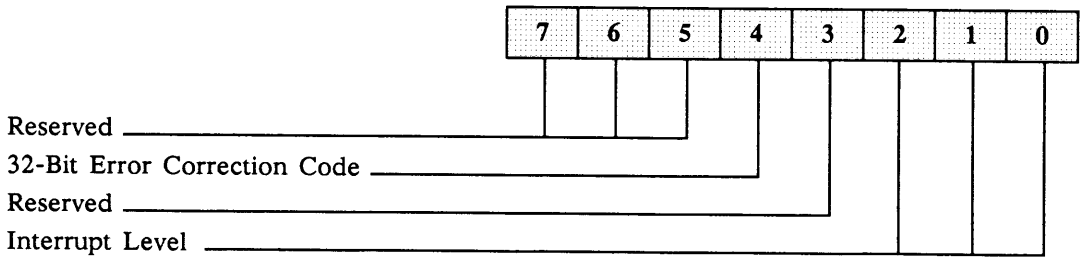
4.2.8 IOPB Byte 12 (EPROM Revision Level)

This byte contains the revision level of the EPROM plugged into the board.

4.3 Drive Parameters IOPB

	7	6	5	4	3	2	1	0
00	ERRS	DONE	CHEN	0	Command Code			
01	Completion Code							
02	0	SR	CSE	WRPT	DFLT	SKER	ONCL	DRDY
03	Internal Status							
04	Subfunction Code							
05	FIXD	0				Unit		
06	0			EC32	0	Interrupt Level		
07	Interrupt Vector							
08	Max Sector L/H							
09	Head Offset							
0A	Max Cylinder High							
0B	Max Cylinder Low							
0C	Max Head							
0D	Max Sector							
0E	Sectors Per Track (Rd. Dr. Prm.)							
0F	0	Next IOPB Address Modifier						
10	0							
11	0							
12	0							
13	0							
14	Next IOPB Address High							
15	Next IOPB Address							
16	Next IOPB Address							
17	Next IOPB Address Low							
18	IOPB Checksum High							
19	IOPB Checksum Low							

4.3.1 IOPB Byte 6 (Drive Parameters)



Bit	Mnemonic	Description
7-5		**Reserved.**
4	EC32	**32-Bit ECC:** When set, the 753 uses a 32-bit ECC on the header and data. When clear, it uses a redundant header and a 48-bit data ECC.
3		**Reserved.**
2-0	INTL	**Interrupt Level:** See Section 4.1.7.

4.3.2 IOPB Byte 8 (Max Sector/Last Head)

IOPB Byte 8 specifies the last head's maximum sector value for use in cylinder sparing. This value is zero-based. Byte 0DH must equal Byte 08H if cylinder sparing is not used (see Section 8.3.2).

4.3.3 IOPB Byte 9 (Head Offset)

IOPB Byte 9 specifies the drive's head offset value. Use zero for non-fixed/removable drives (see Section 8.13.1).

4.3.4 IOPB Bytes A and B (Max Cylinder)

IOPB Byte A is Max Cylinder High; Byte B is Max Cylinder Low. These bytes specify the drive's max cylinder value. This value is zero-based, i.e., the max cylinder on an 823-cylinder drive is 822.

4.3.5 IOPB Byte C (Max Head)

IOPB Byte C specifies the drive's max head value (zero-based).

4.3.6 IOPB Byte D (Max Sector)

IOPB Byte D specifies the drive's max sector value (zero-based; see Section 8.3).

4.3.7 IOPB Byte E (Sectors Per Track)

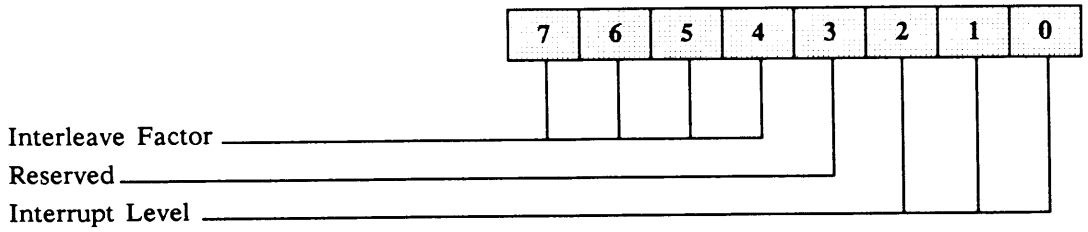
IOPB Byte E returns the number of sectors per track during a Read Drive Parameters command. The 753 determines this figure by counting the sector pulses from the drive. This value is the actual number of sectors; it is not zero-based and does not include a runt sector.

The Format command determines how many sectors to format using this count. Normal Read and Write commands use this count to limit the number of header compares before a *header not found* error occurs.

4.4 Format Parameters IOPB

	7	6	5	4	3	2	1	0
00	ERRS	DONE	CHEN	SGM	Command Code			
01	Completion Code							
02	0	SR	CSE	WRPT	DFLT	SKER	ONCL	DRDY
03	Internal Status							
04	Subfunction Code							
05	FIXD	0				Unit		
06	Interleave				0	Interrupt Level		
07	Interrupt Vector							
08	Field 1							
09	Field 2							
0A	Field 3							
0B	Field 4							
0C	Bytes Per Sector High							
0D	Bytes Per Sector Low							
0E	0							
0F	0	Next IOPB Address Modifier						
10	Field 6							
11	Field 7							
12	0							
13	0							
14	Next IOPB Address High							
15	Next IOPB Address							
16	Next IOPB Address							
17	Next IOPB Address Low							
18	IOPB Checksum High							
19	IOPB Checksum Low							

4.4.1 IOPB Byte 6 (Interleave)



Bit	Mnemonic	Description
7-4	INTF	**Interleave Factor:** The 753 uses INTF during format operations. For 1:1 interleaving, the interleave factor is zero. The interleave factor for other ratios is (n+1), where n is the interleave factor.

Table 4-5 Interleave Factors

Interleave Factor	Ratio
0	1:1
1	2:1
2	3:1
:	:
F	16:1

3		**Reserved.**
2-0	INTL	**Interrupt Level:** See Section 4.1.7.

4.4.2 IOPB Byte 8 (Field 1)

Field 1 is the number of bytes from one byte after the index or sector pulse to when the 753 enables *Read Gate* for headers. This value must be greater than or equal to one. Xylogics recommends one.

> *The combined value of Fields 1 and 2 must not exceed 255.*

4.4.3 IOPB Byte 9 (Field 2)

Field 2 is the number of bytes from when the 753 enables *Read Gate* to when it starts looking for the Header Sync byte. Field 2 must equal a non-zero value. Xylogics recommends 0AH. *See the note below Section 4.4.2.*

4.4.4 IOPB Byte A (Field 3)

Field 3 is the number of bytes from the sector pulse to the Header Sync byte. This value must be greater than or equal to the sum of Field 1 plus Field 2. Xylogics recommends 1BH.

4.4.5 IOPB Byte B (Field 4)

Field 4 is the number of bytes between the Header ECC and the Data Sync byte. This value must be larger than 11. Xylogics recommends 14H.

> *Field 4 is actually four bits longer than this byte specifies. The 753 uses the extra four bits when testing for a successful header compare and header verify.*

4.4.6 IOPB Bytes C and D (Bytes Per Sector High/Low)

IOPB Byte C specifies Bytes Per Sector High; Byte D specifies Bytes Per Sector Low. The 753 supports only 512-byte (200H) sectors.

4.4.7 IOPB Byte 10 (Field 6)

Field 6 is the number of bytes between enabling *Read Gate* and searching for the Data Sync byte. This value must be larger than nine. Xylogics recommends 0AH.

4.4.8 IOPB Byte 11 (Field 7)

Field 7 is the number of bytes *Write Gate* remains on after the Data ECC. This value must be greater than or equal to one. Xylogics recommends three.

Section 5: Commands

Section 5: Commands

5.0 Introduction

Each command begins a new page. A highlighted IOPB diagram follows each command description. The highlights indicate which fields the 753 absolutely requires for command execution, which fields are optional, and which fields it returns after execution.

Each IOPB is 30-bytes long. Reserving all 30 bytes for each IOPB maintains IOPB integrity. Generally, all commands use Bytes 0 through 19H. Bytes 1AH through 1DH are reserved.

5.0.1 Setting Up the Command

Each IOPB diagram indicates the bytes or fields that must be set for each operation. Certain parameters are essential; others are optional. All commands require that the command, unit, and interrupt level fields contain valid information. The interrupt vector field must be valid if the interrupt level is not zero.

5.0.2 Completing the Command

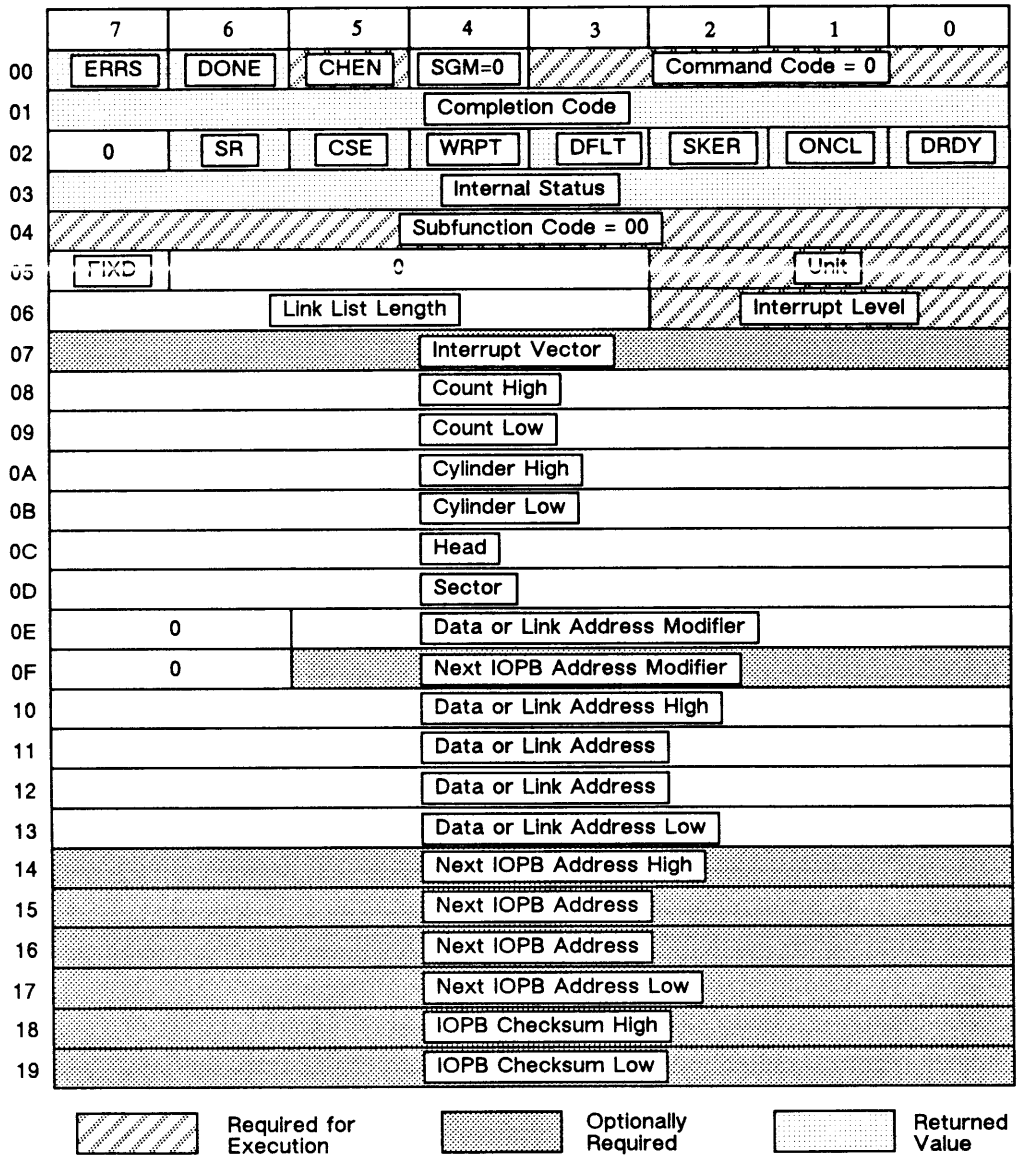
After the 753 completes the command, it updates IOPB Bytes 0 through 3 with *Error Summary* (ERRS), *Done*, a completion code, and an internal status. The 753 only updates the entire IOPB if *Auto-update* (AUD) is set, an error occurs, or if read parameters or read extended status commands are executed. If AUD is set, and no errors occur, the 753 sets *Done*, posts a completion code of zero in Byte 1, and disk drive status information in Byte 2. For any command that DMAs data to or from memory, the 753 updates the data and disk addresses to point to the transfer's last address plus one.

Table 5-1. Command Completion

Status	Action
AUD Clear/No Error Occurs	753 updates Bytes 0-3 with ERRS, *Done*, a completion code, and internal status
AUD Set/No Error Occurs	753 updates the entire IOPB
AUD Clear/Error Occurs	753 updates the entire IOPB
AUD Clear/A Read Parameters or Read Extended Status Command is Executed	753 updates the entire IOPB

5.1 No Operation

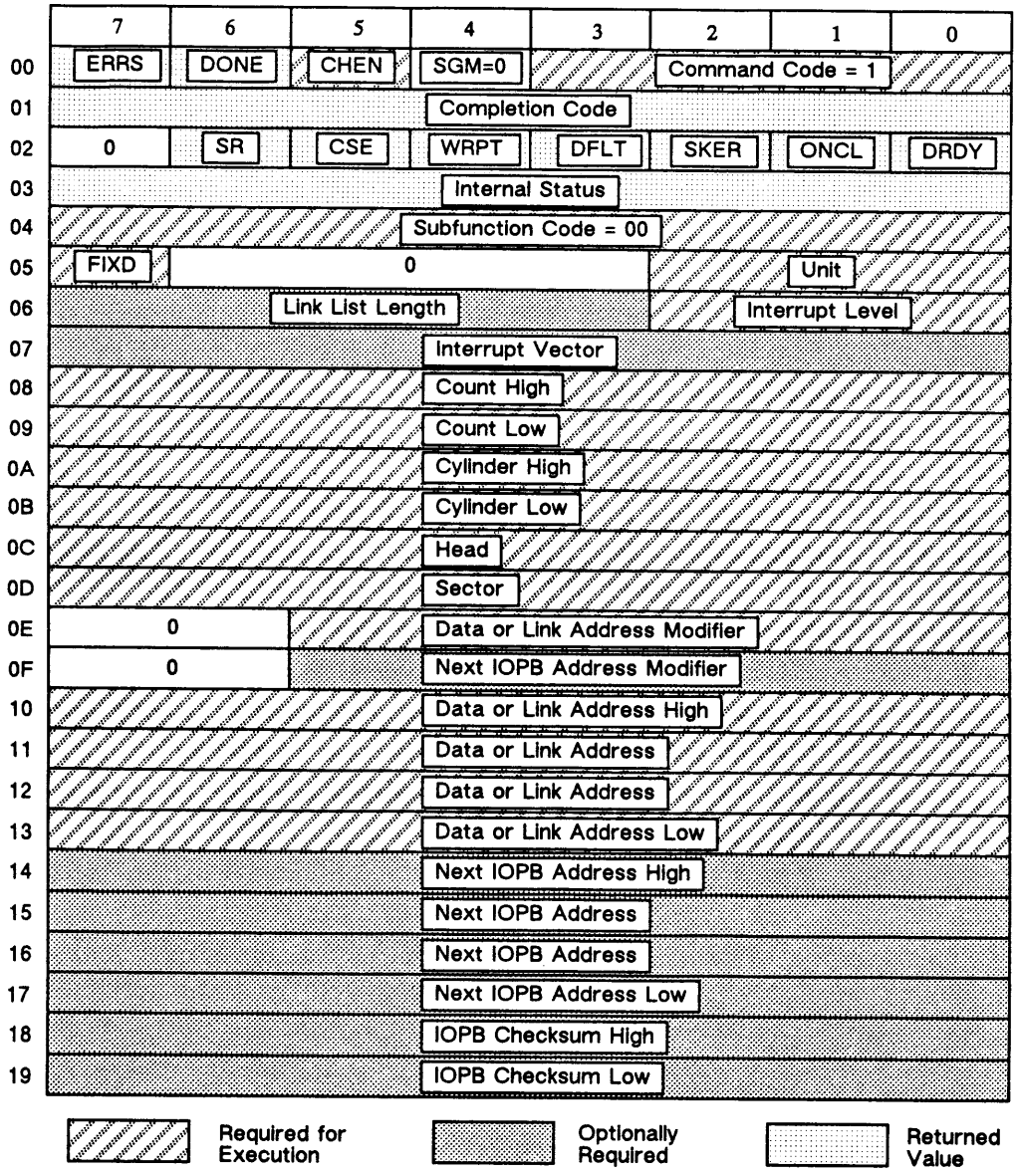
The No Operation command is a diagnostic tool; the 753 reads the IOPB and marks it *Done*.



5.2 Write Data

After reading and decoding the IOPB, the 753 positions the disk drive heads at the target cylinder, reads in the data from host memory as per the IOPB, and writes the data contiguously to the specified disk address and subsequent sequential sectors.

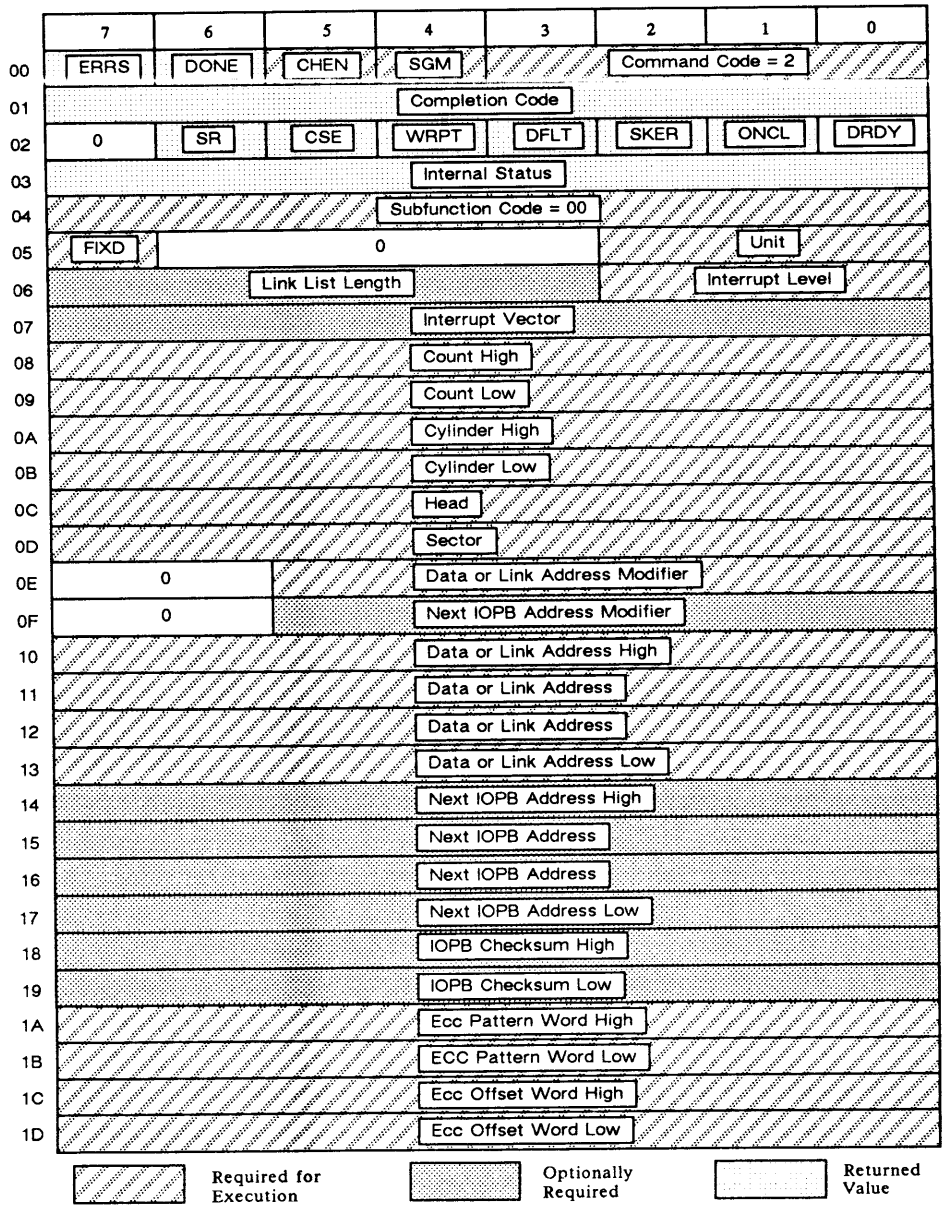
Write Data has two IOPB formats: normal and scatter/gather. A normal IOPB writes one contiguous block of host memory to the disk. A gather write IOPB places up to 32 different blocks of host memory in contiguous sectors on the disk (see Section 8.10).



5.3 Read Data

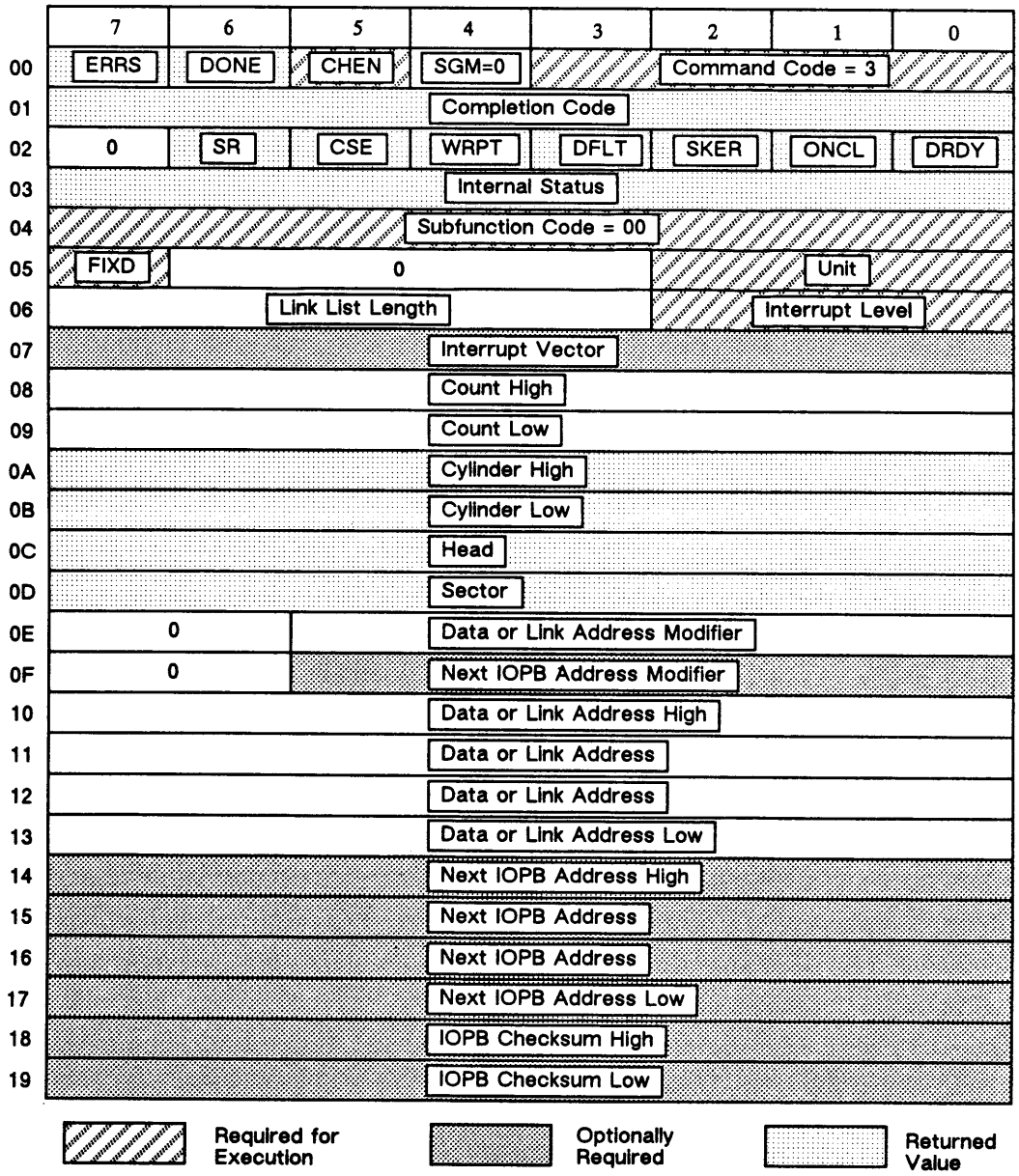
After reading and decoding the IOPB, the 753 positions the disk drive heads at the target cylinder, reads the disk data as per the IOPB, and writes the data in host memory. The 753 reads data beyond the requested sector count until the buffer fills or a seek is requested.

Read Data has two IOPB formats: normal and scatter/gather. A normal IOPB places the disk data in one contiguous block of host memory. A scatter read IOPB places the disk data in up to 32 different blocks of host memory (see Section 8.9).



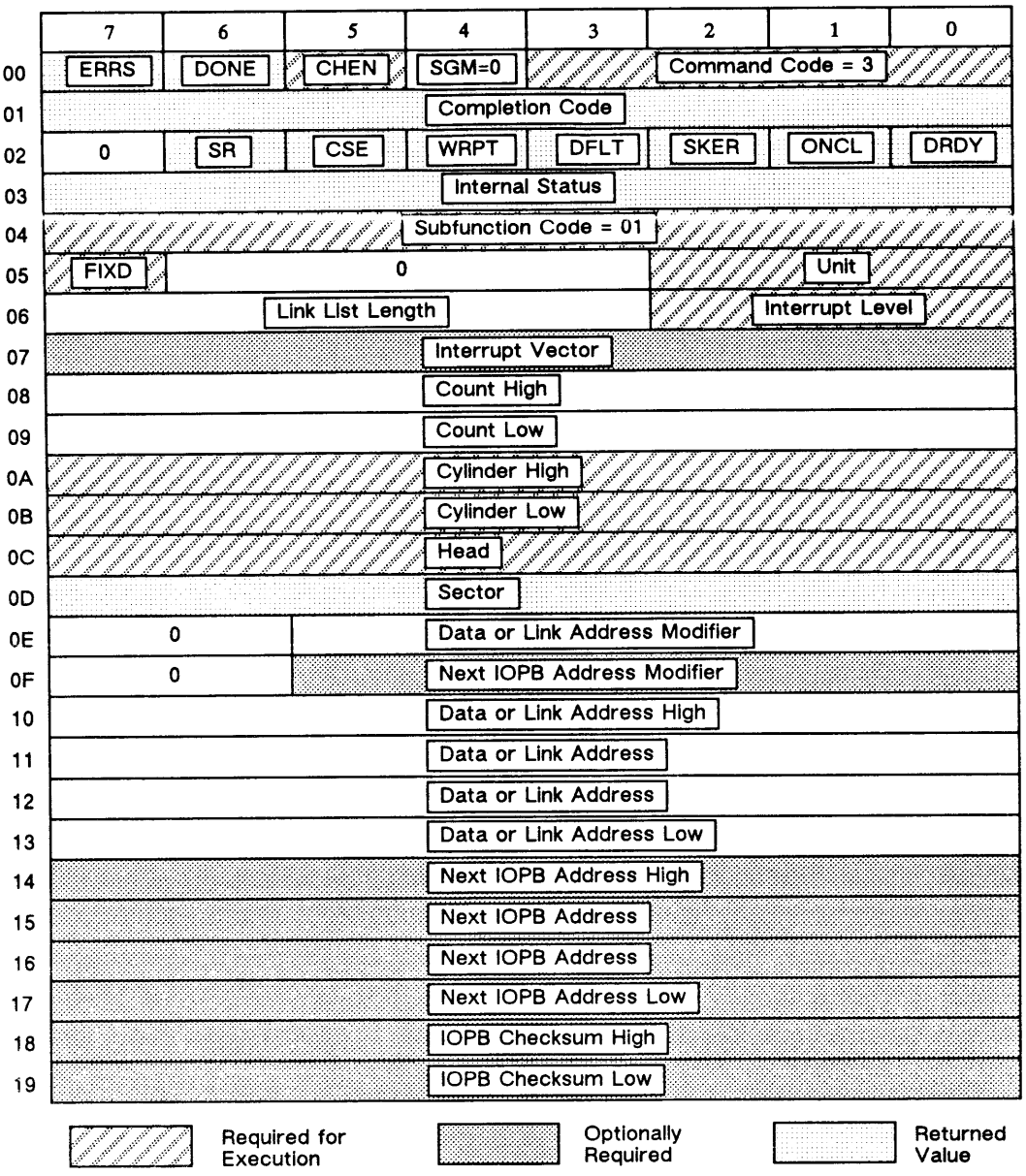
5.4 Report Current Address

The 753 selects the disk drive, reads the first good header field, and returns the address to the host via the IOPB; it updates the IOPB regardless of AUD's status.



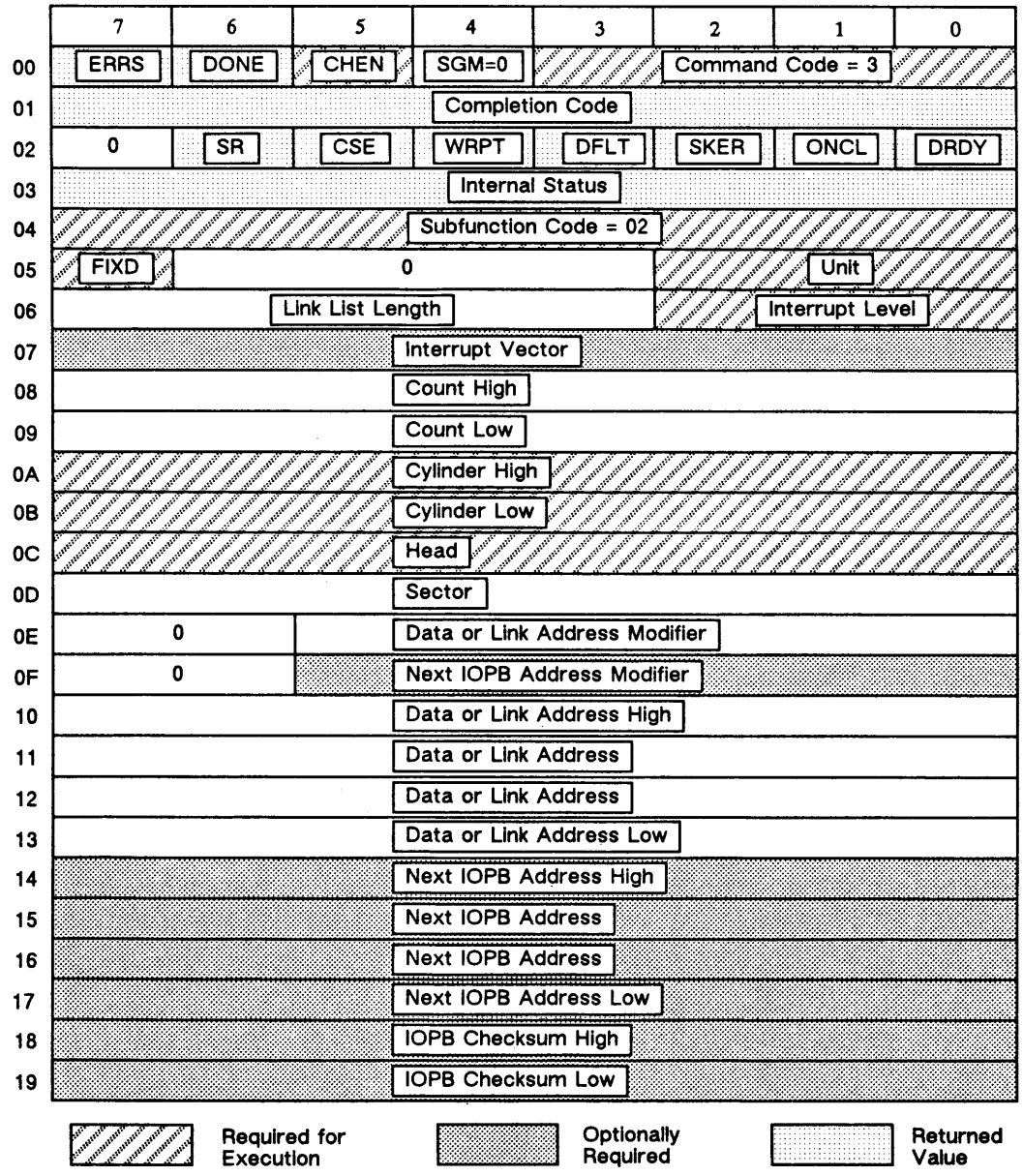
5.5 Seek and Report Current Address

The 753 issues a seek to the selected disk drive for the target cylinder. After the drive completes the seek, the 753 reads the first good header field it encounters and reports it to the host via the completed IOPB. The 753 updates the IOPB regardless of AUD's status.



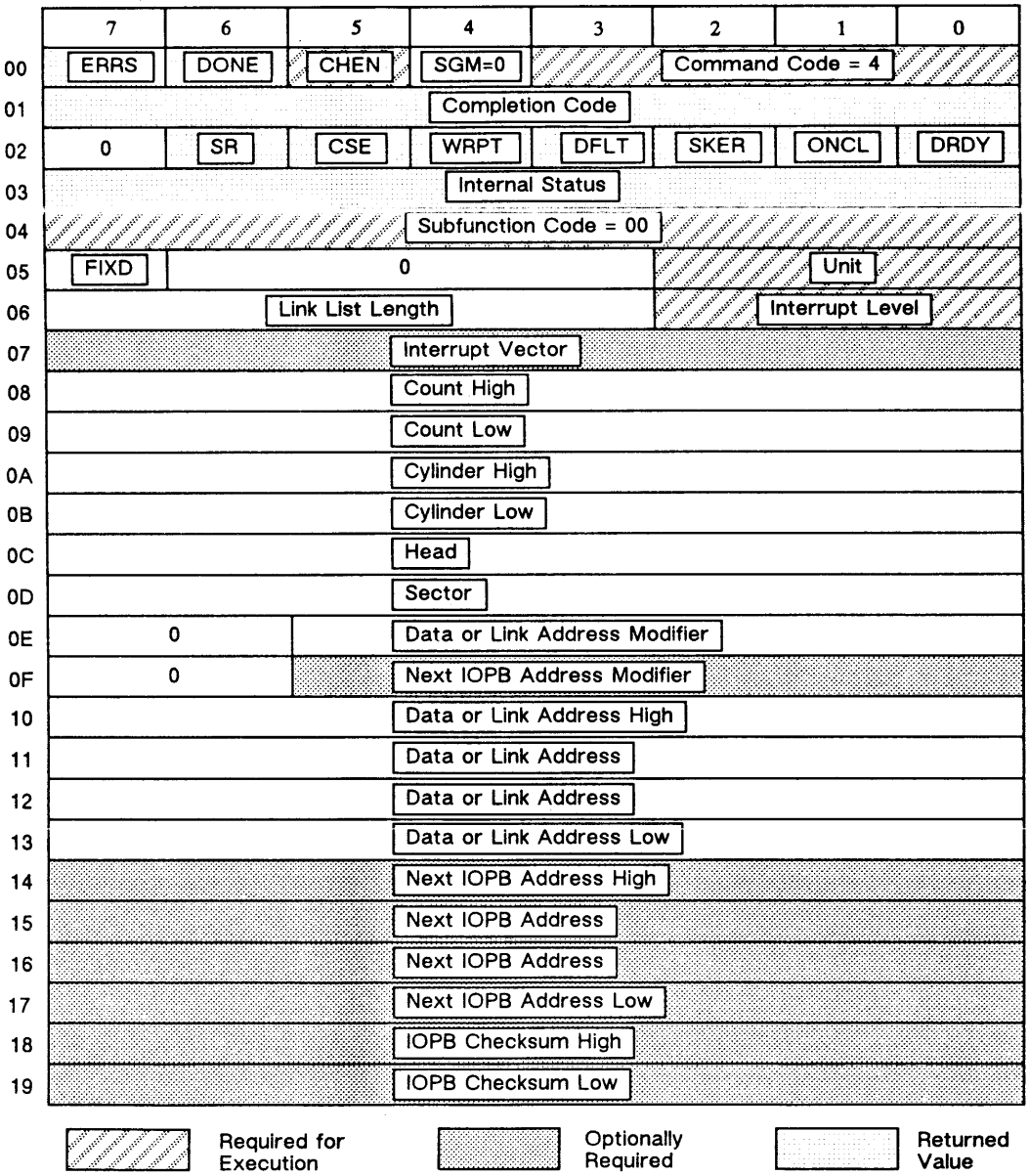
5.6 Start Seek and Report Completion Immediately

The 753 issues a seek to the selected disk drive for the target cylinder, and reports a completion to the host without waiting for the drive to complete the seek.



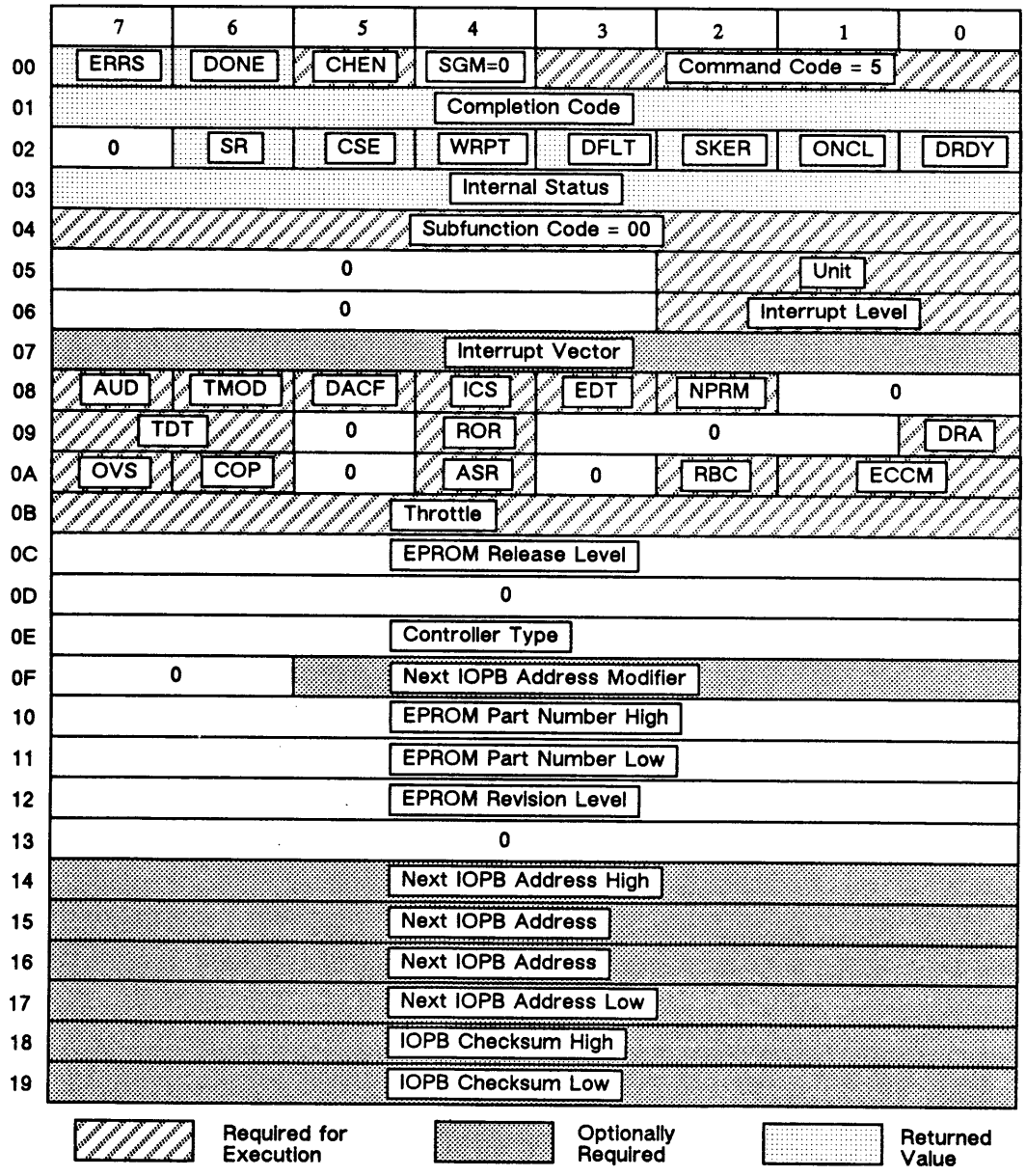
5.7 Drive Reset

The 753 resets the disk drive. First it issues a fault clear and then a recalibrate. The IOPB is complete when the recalibrate completes or times out on drives that are *ready*. The 753 does not wait for the recalibrate to complete on drives that are not *ready*.



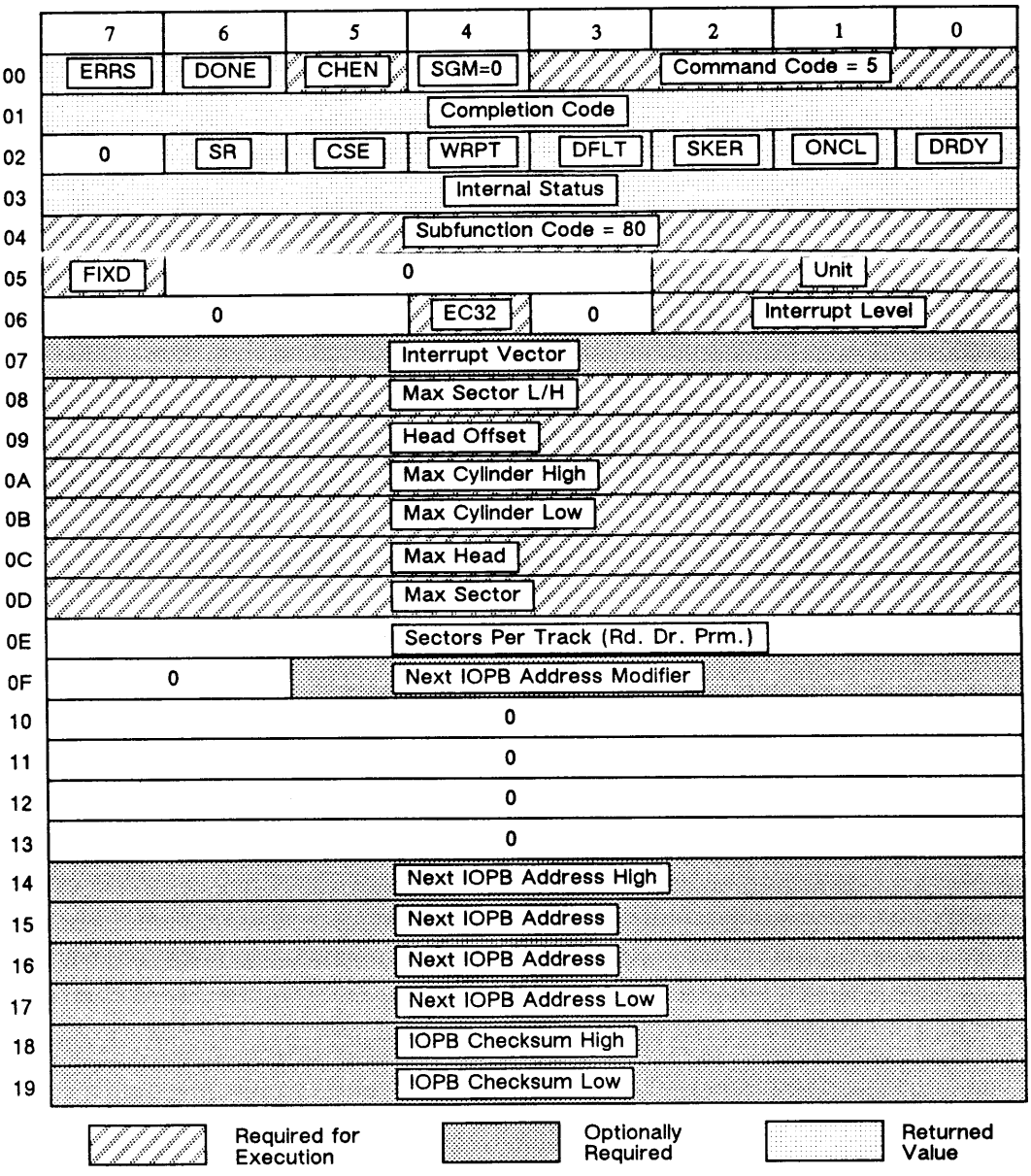
5.8 Write Controller Parameters

This command initializes the 753 with its operational parameters. The 753 assumes no default parameters (see Section 4.2).



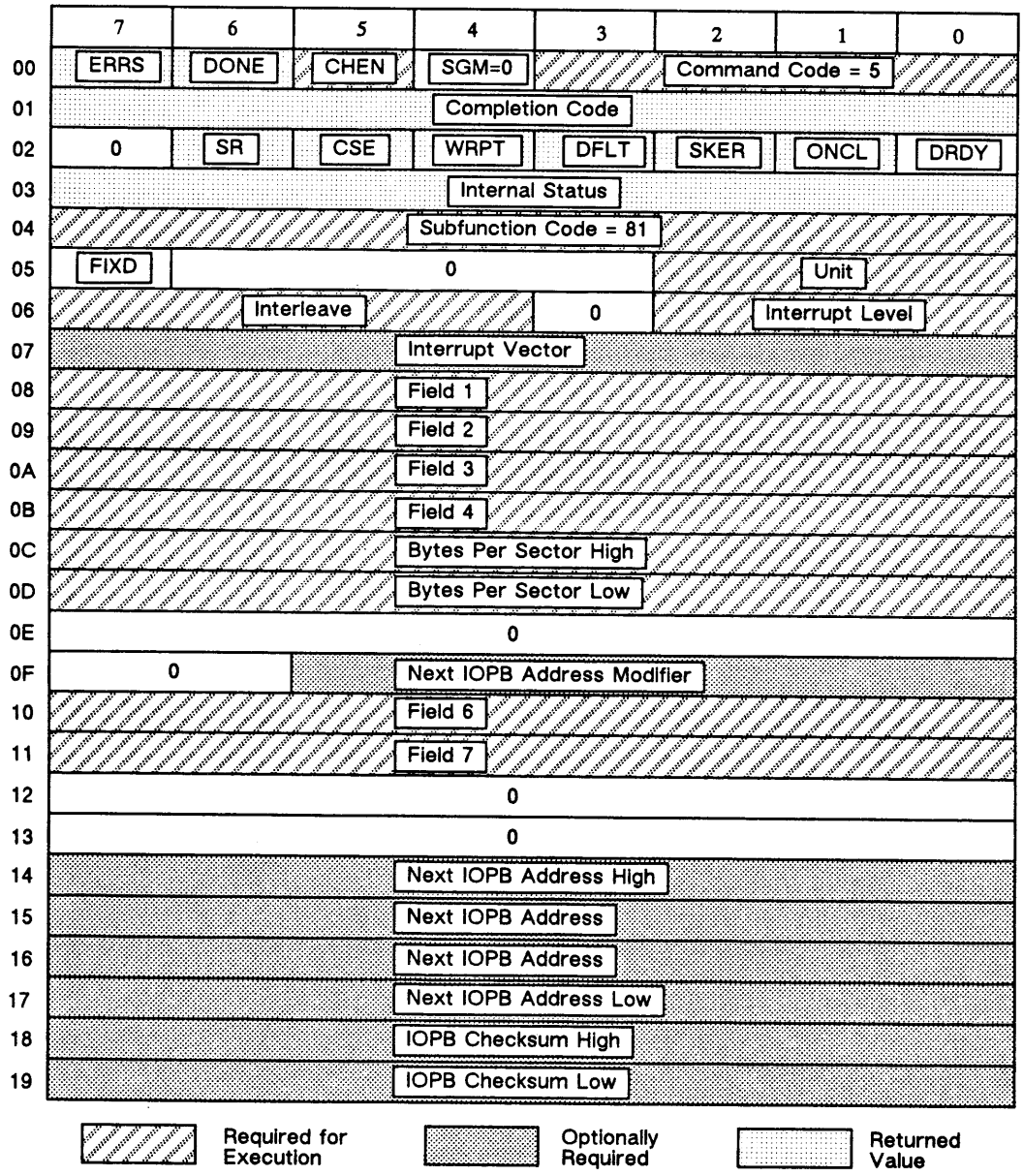
5.9 Write Drive Parameters

This command specifies the disk drive's physical characteristics. The 753 assumes no default parameters (see Section 4.3).



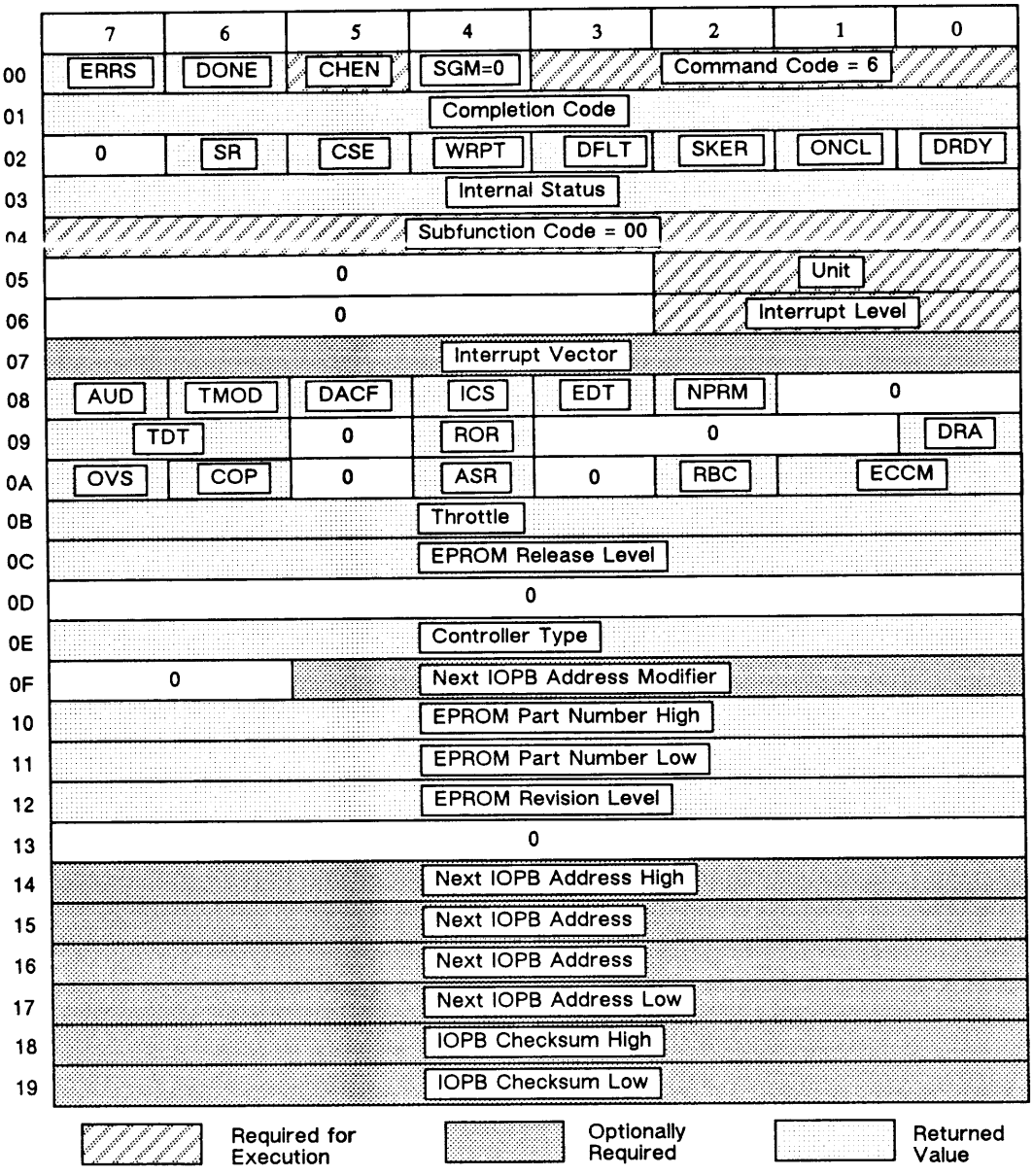
5.10 Write Format Parameters

This command specifies the drive's media format. The 753 only supports 512 bytes per sector (see Section 4.4).



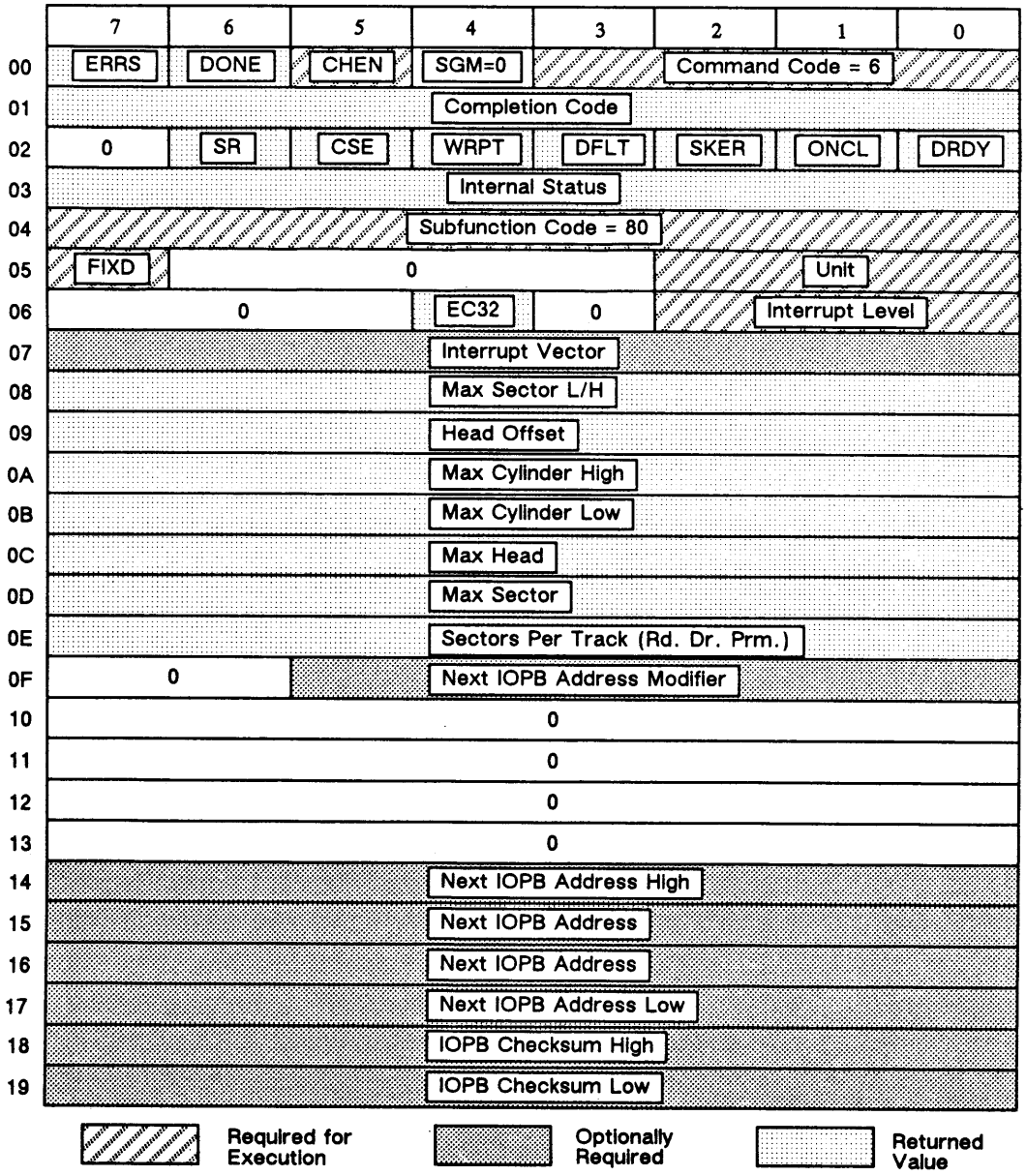
5.11 Read Controller Parameters

The 753 returns its current operational parameters to the host via the IOPB. It updates the IOPB regardless of AUD's status (see Section 4.2).



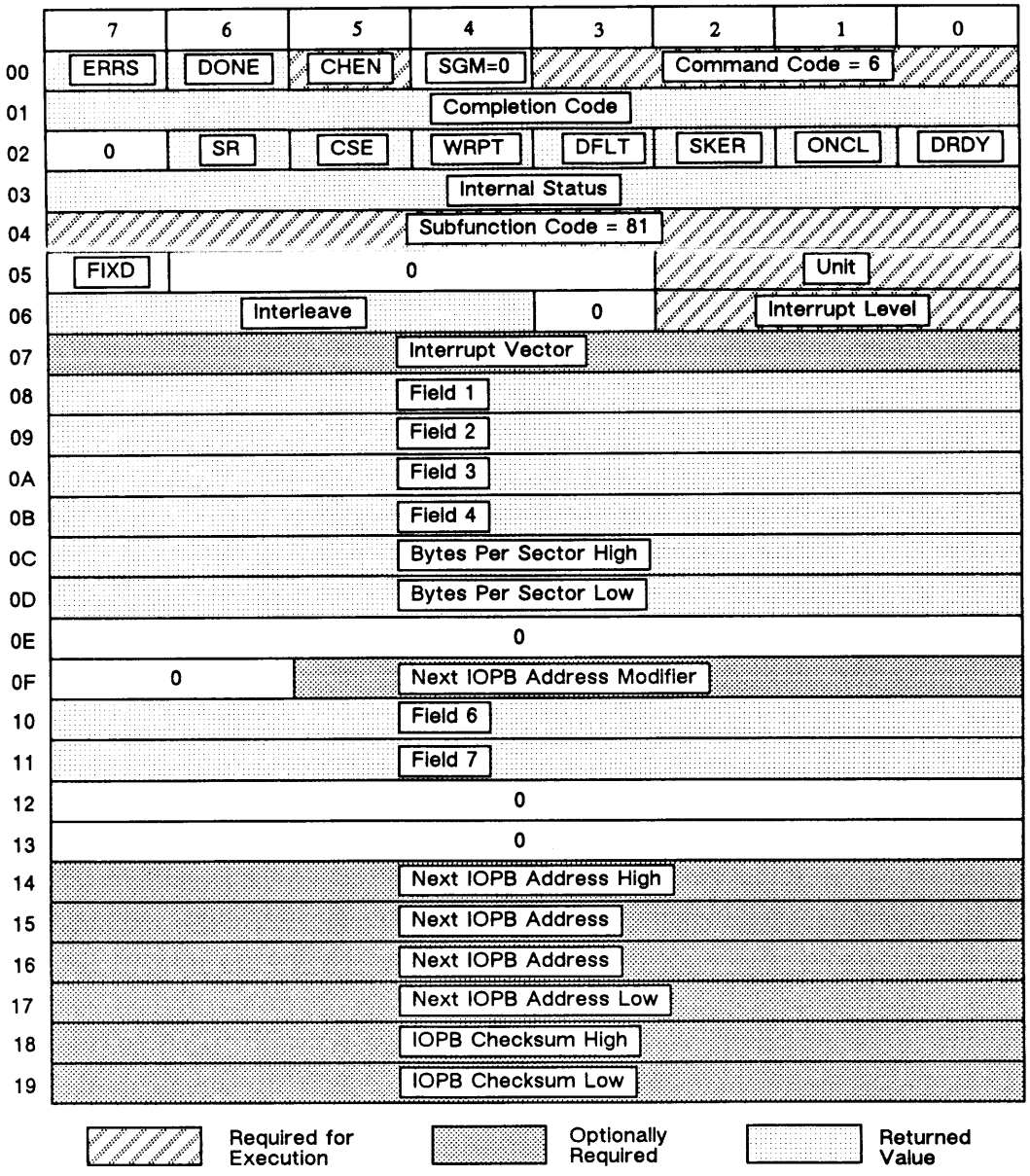
5.12 Read Drive Parameters

The 753 returns the values programmed for each drive's physical characteristics to the host via the IOPB. It returns the specified drive's actual number of sectors per track in Byte 0EH. The 753 updates the IOPB regardless of AUD's status (see Section 4.3).



5.13 Read Format Parameters

The 753 returns the selected disk drive's format parameters to the host via the IOPB. It updates the IOPB regardless of AUD's status (see Section 4.4).



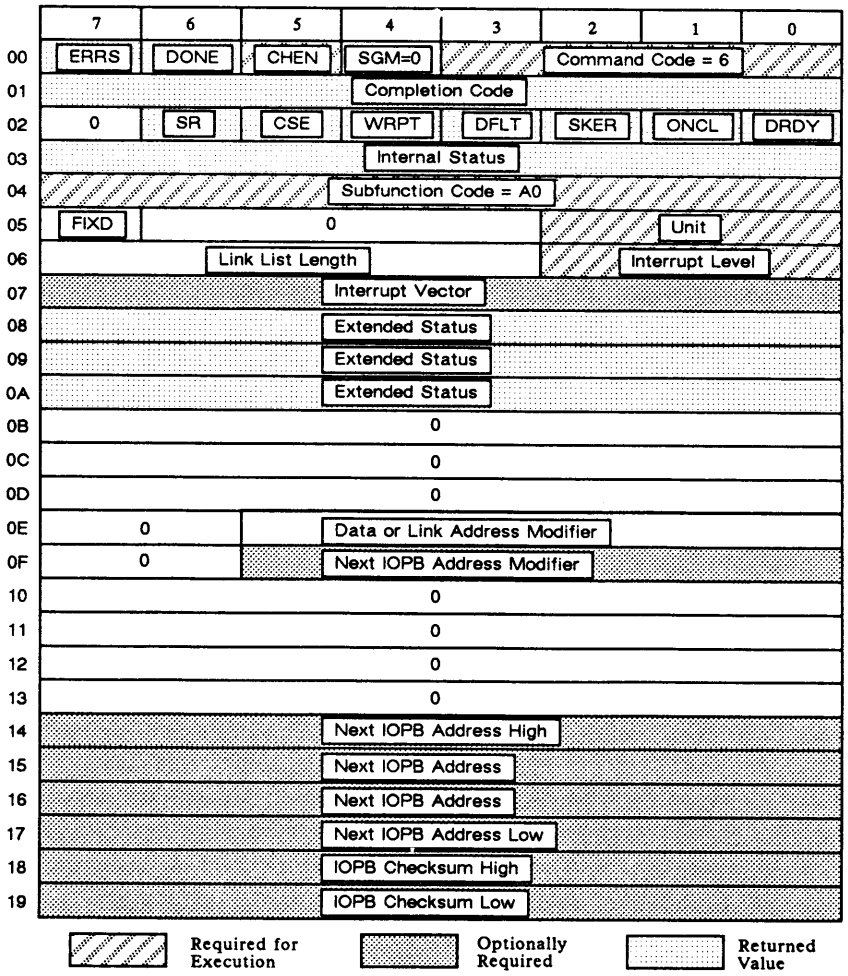
5.14 Read Drive Status Extended

The 753 reads the SMD drive interface's extended status. It returns the drive's status in IOPB Bytes 8 through A regardless of AUD's status (see Table 5-2). The drive-specific bytes follow the same bit alignment as the standard Status byte.

Table 5-2. Extended Drive Status

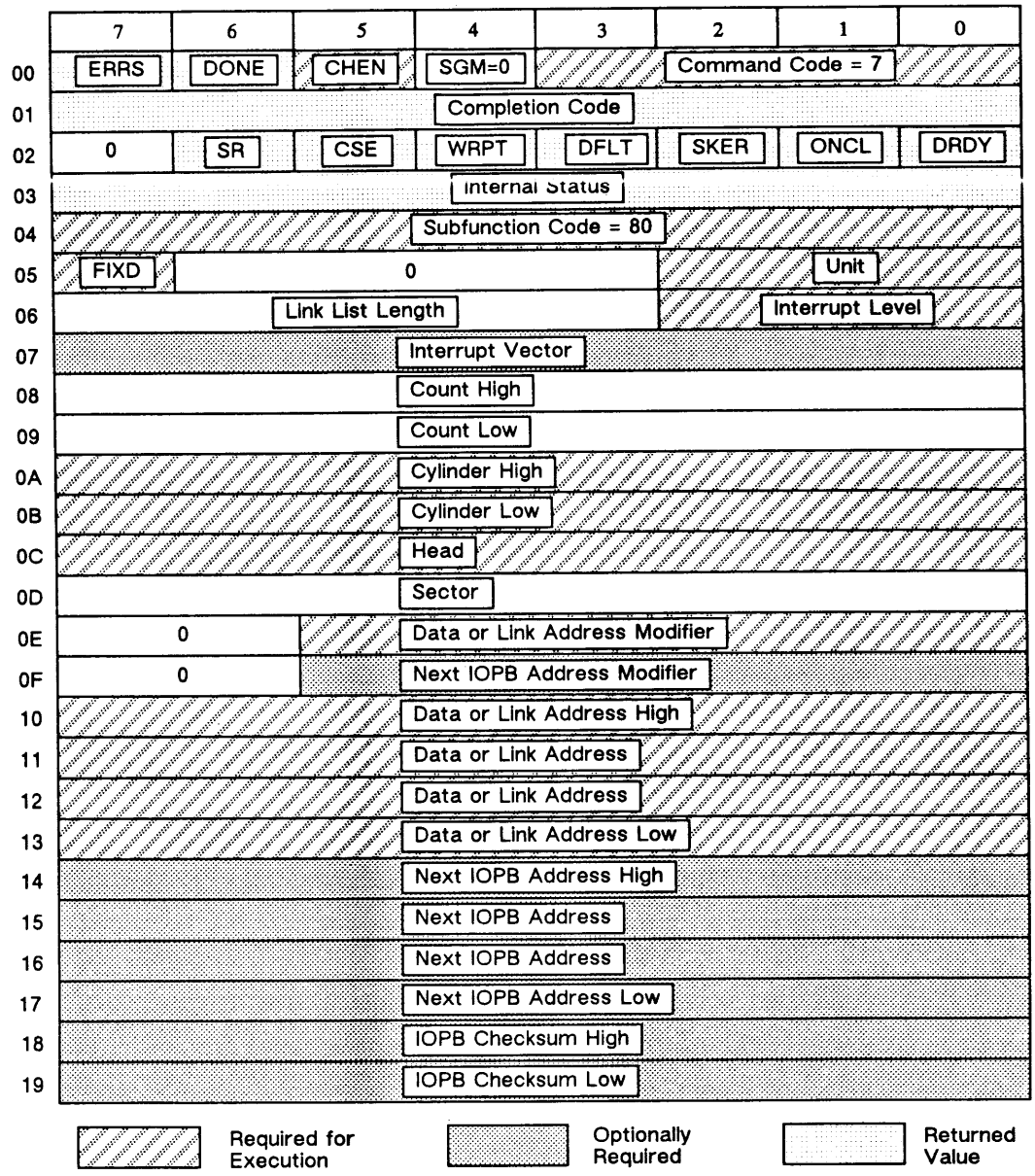
Byte	Tag 4	Tag 5	Bit: 7	6	5	4	3	2	1	0
02	0	0	SECT	INDX	ADM	WRPT	DFLT	SKER	ONCL	DRDY
08	1	0	*	*	*	*	*	*	*	*
09	0	1	*	*	*	*	*	*	*	*
0A	1	1	*	*	*	*	*	*	*	*

*Drive-Specific



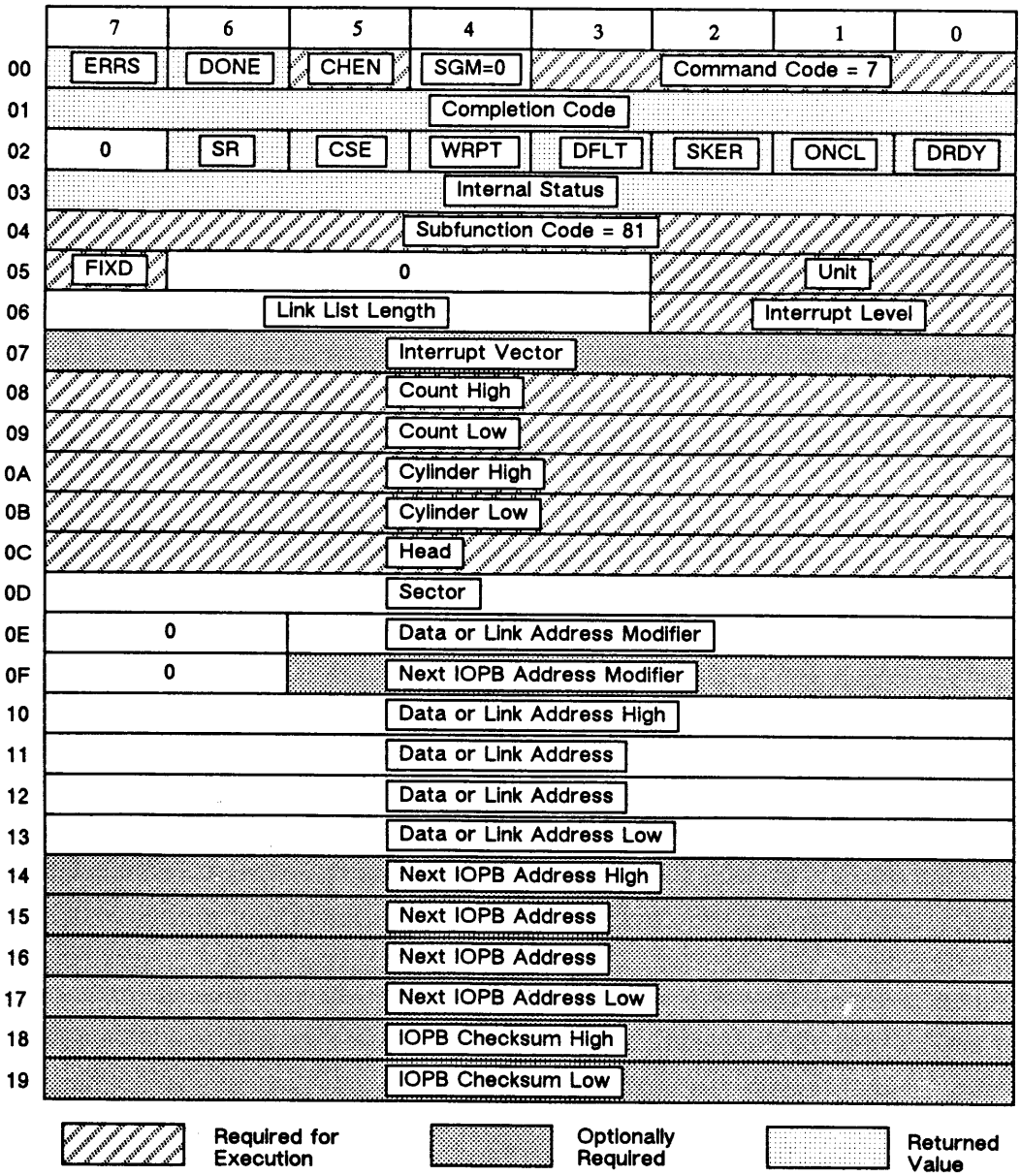
5.15 Write Track Headers

The host writes the sector header fields on a track (only one track per IOPB). The 753 takes the data for the header fields from host memory: four bytes per header; one header for each sector on the track. The data fields are not preserved. It places the data on the track starting from index. Section 8.3 defines the data format in each header.



5.16 Write Format

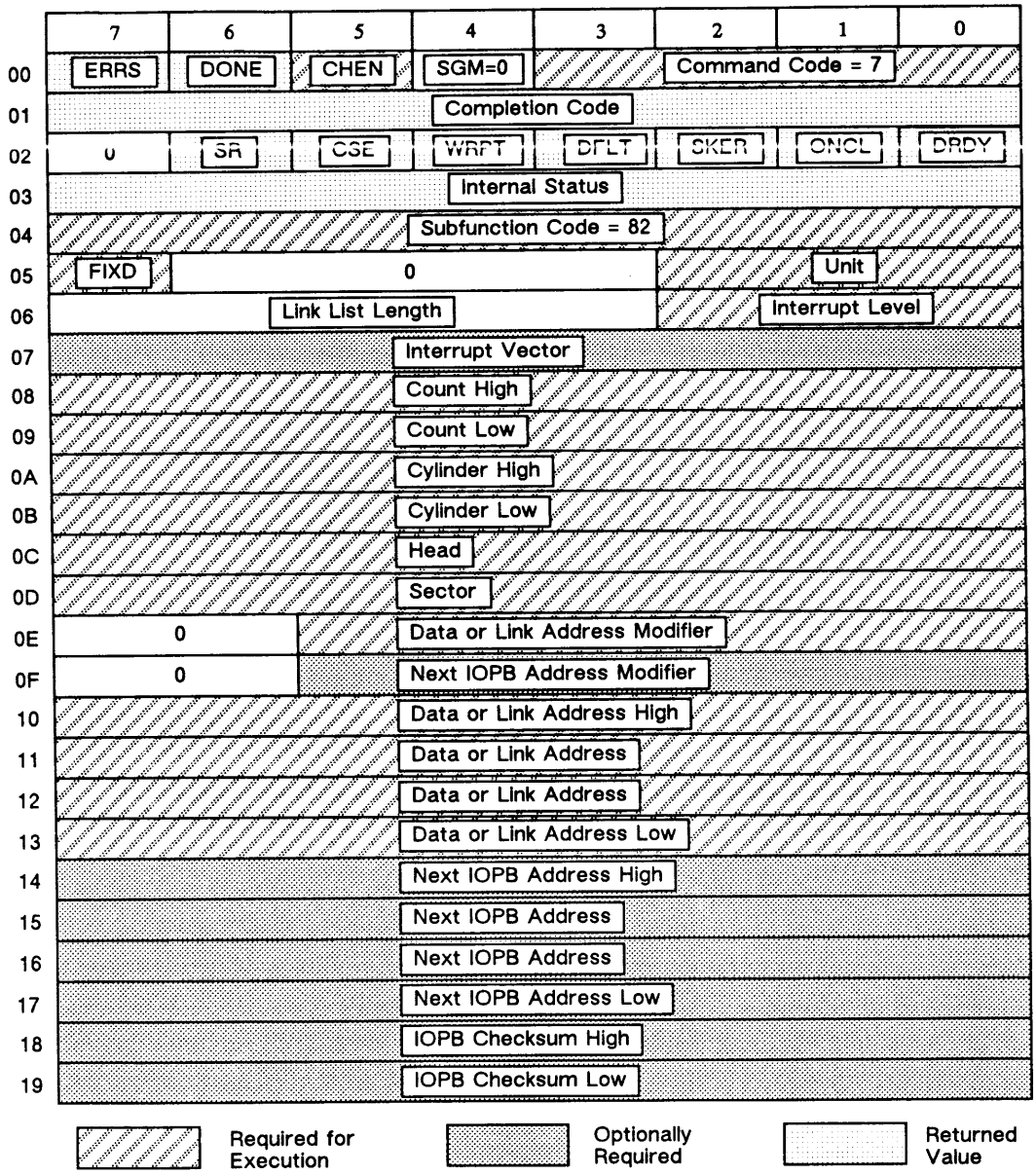
The 753 formats the drive, writing the header of all sectors with the appropriate sector ID. The data field contains zeros and a valid ECC. The Count bytes in this command refer to the number of tracks to be formatted (see Section 8.2).



5.17 Write Header, Header Verify, Data, and Data ECC

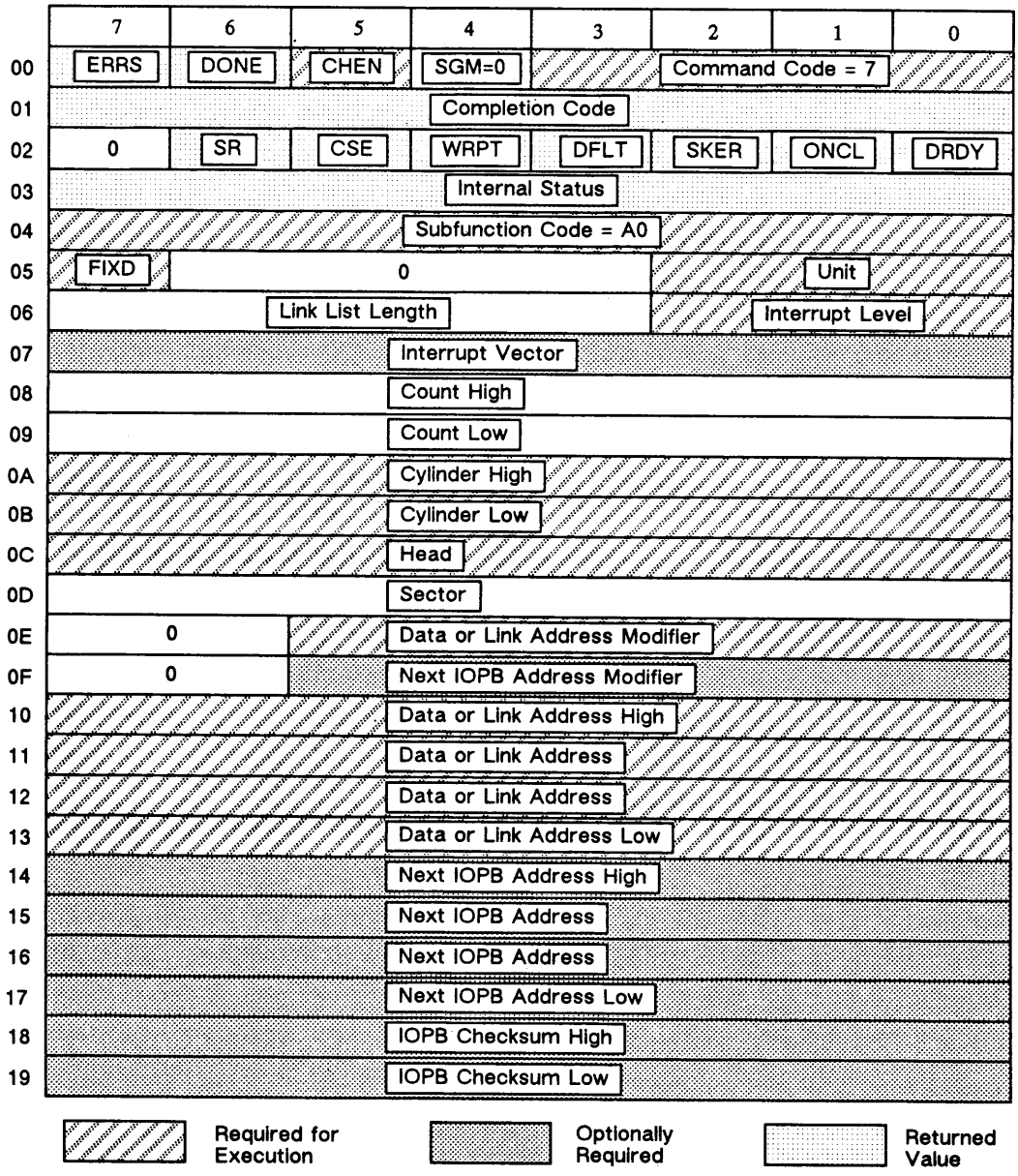
The 753 writes a sector header, header verify, data, and data ECC. There are always four bytes in the header, but the other fields vary according to the initial 753 set-up. The 753 does not cross head or cylinder boundaries while executing this command.

The host must calculate the error correction codes since the 753 does not calculate any ECC fields for this command (see Section 8.14).



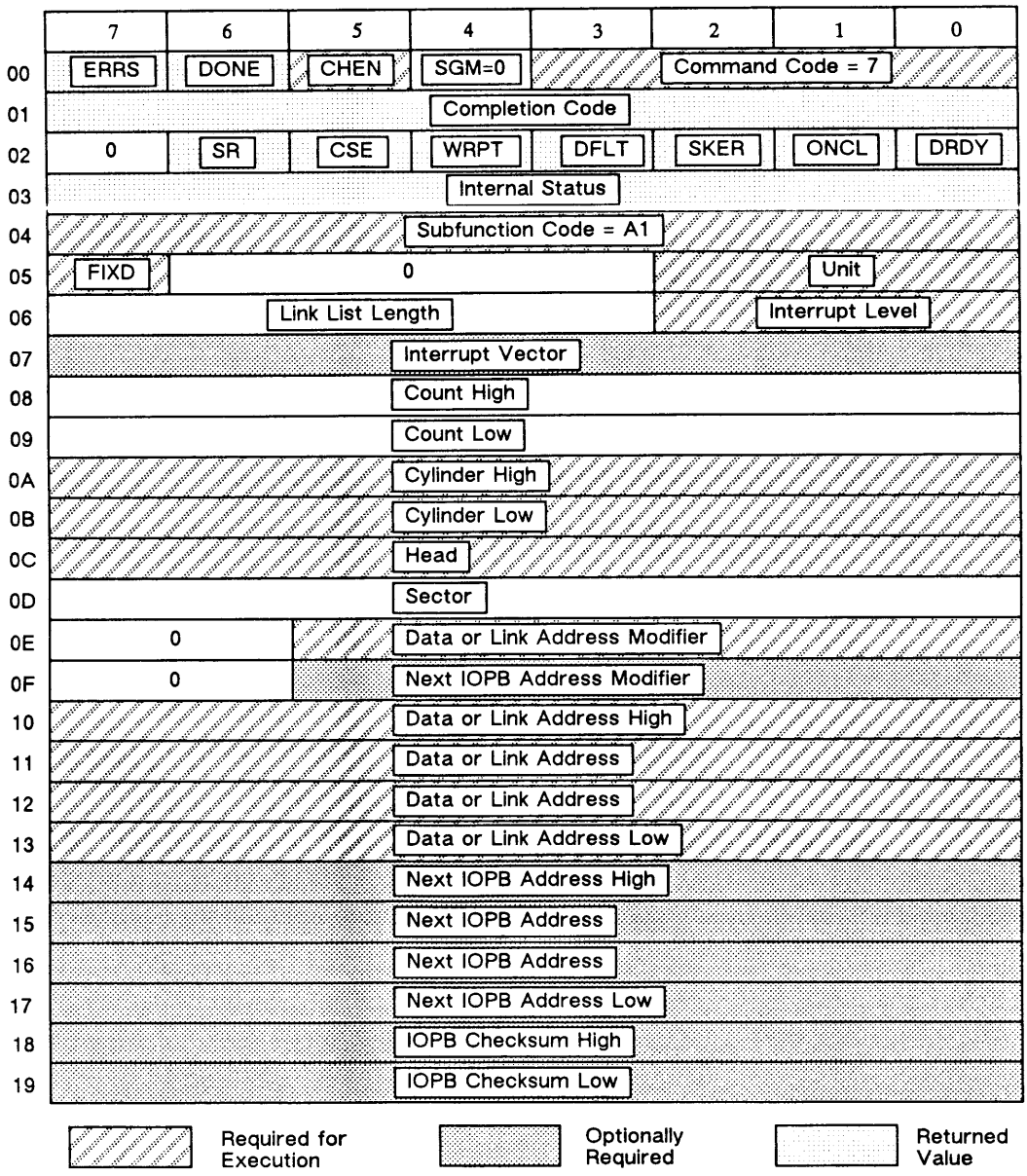
5.18 Write Defect Map

Write Defect Map is a useful maintenance command for debugging software. The 753 uses data from host memory and writes a manufacturer's defect map to the disk (see Section 8.1).



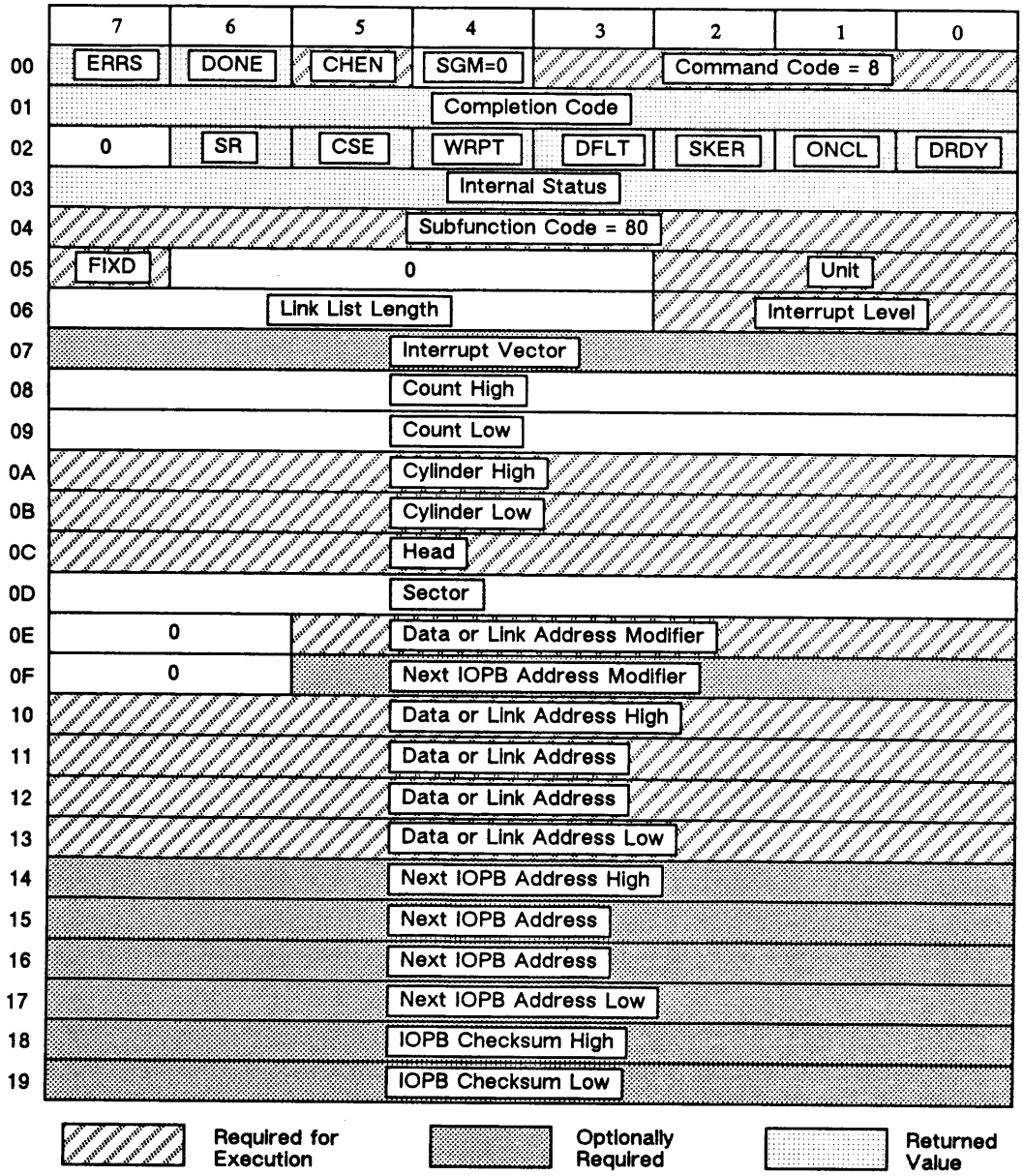
5.19 Write Defect Map Extended

Write Defect Map Extended is a useful maintenance command for debugging software. The 753 uses data from host memory and writes a manufacturer's defect map to an alternate location on the track, avoiding media defects by offsetting the defect map from index.



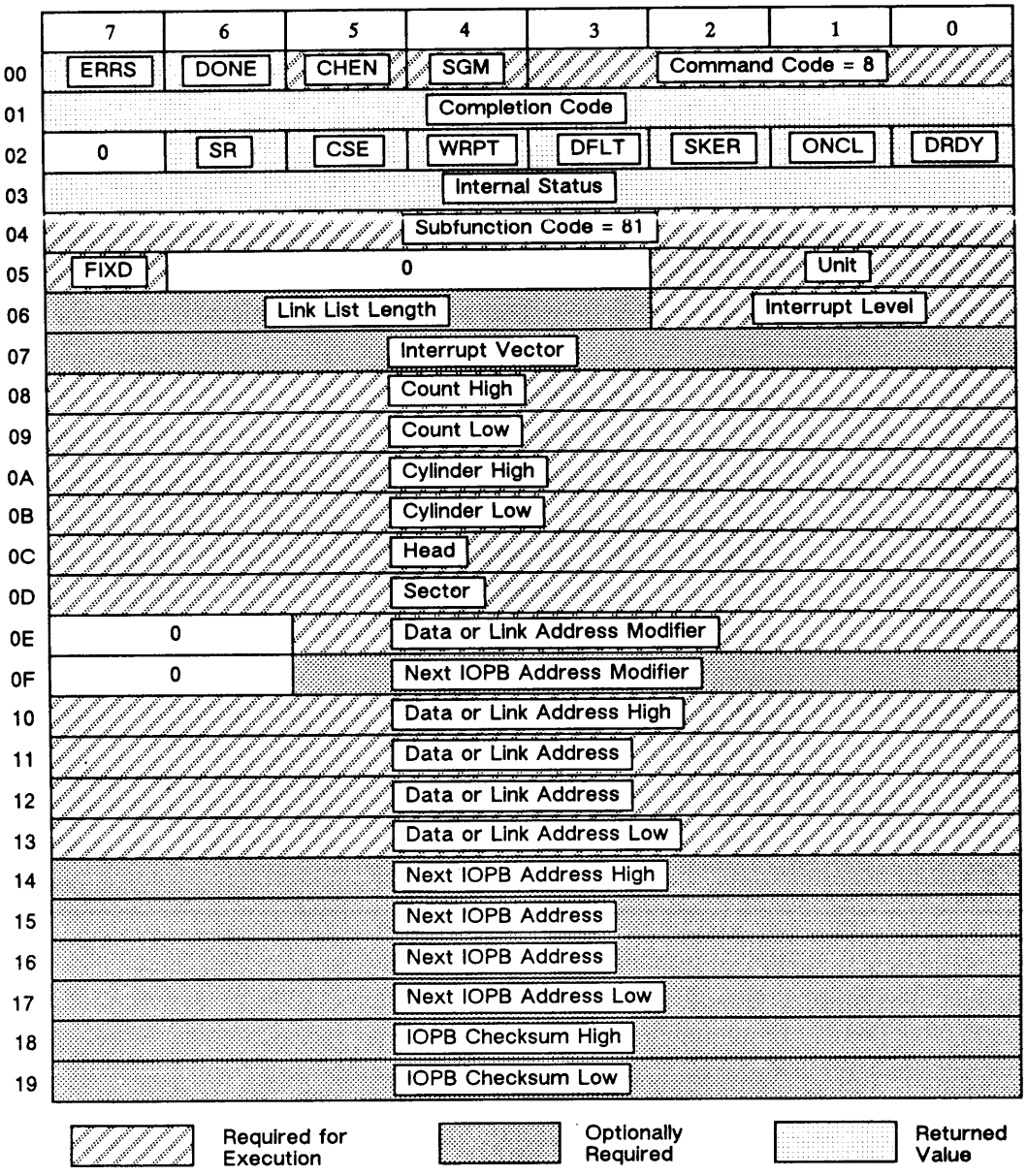
5.20 Read Track Headers

The host reads the sector header fields on a track. The 753 places the data from the header fields in host memory: four bytes per header; one header for each sector on the track. Section 8.3 defines the data format in each header.



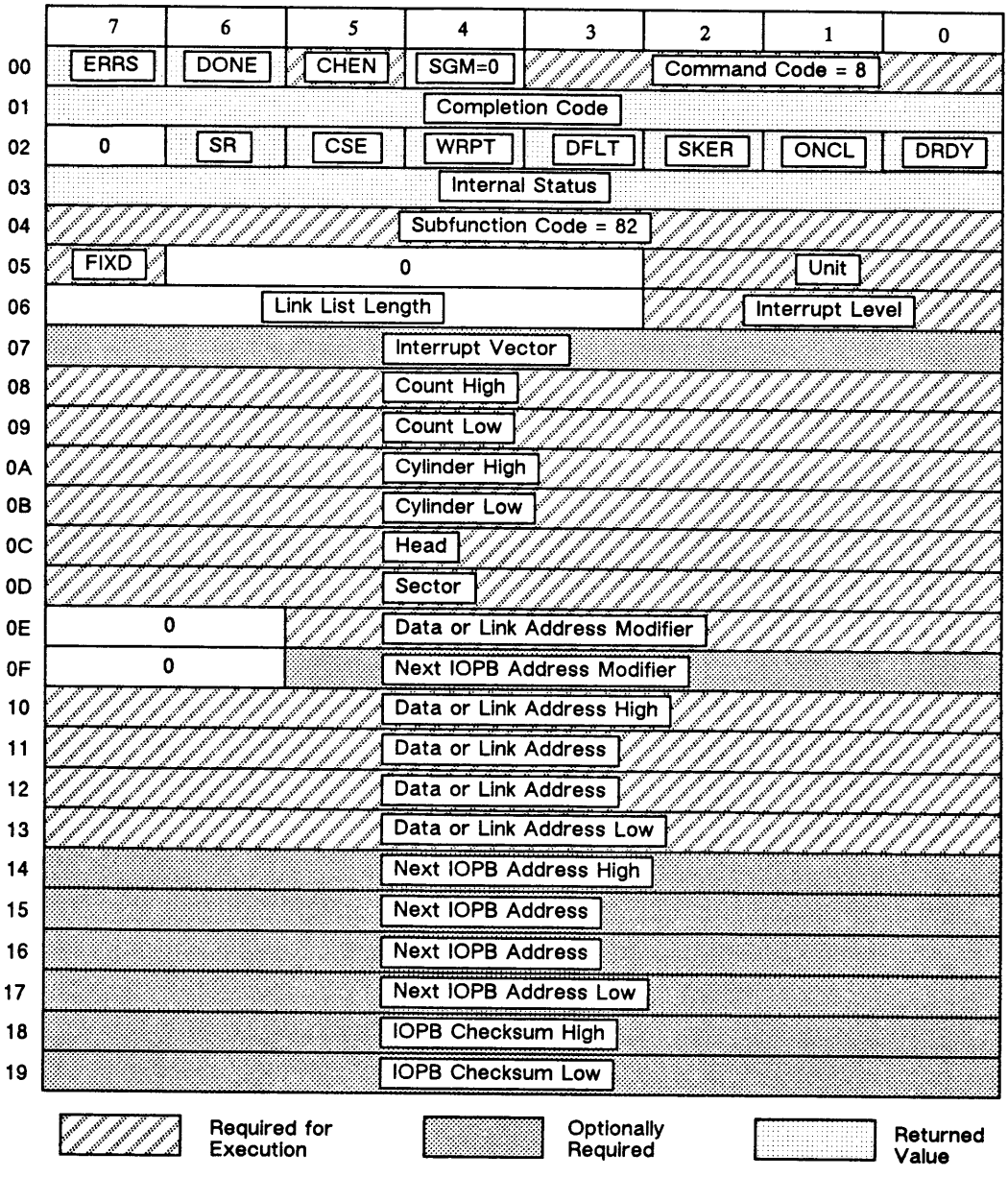
5.21 Verify Data

The 753 verifies the data on the disk: it simultaneously reads the data from the host and the disk, and compares them on a bit-by-bit basis. The granularity of the mismatch reporting is one sector. The ending data address does not indicate a mismatch error's location.



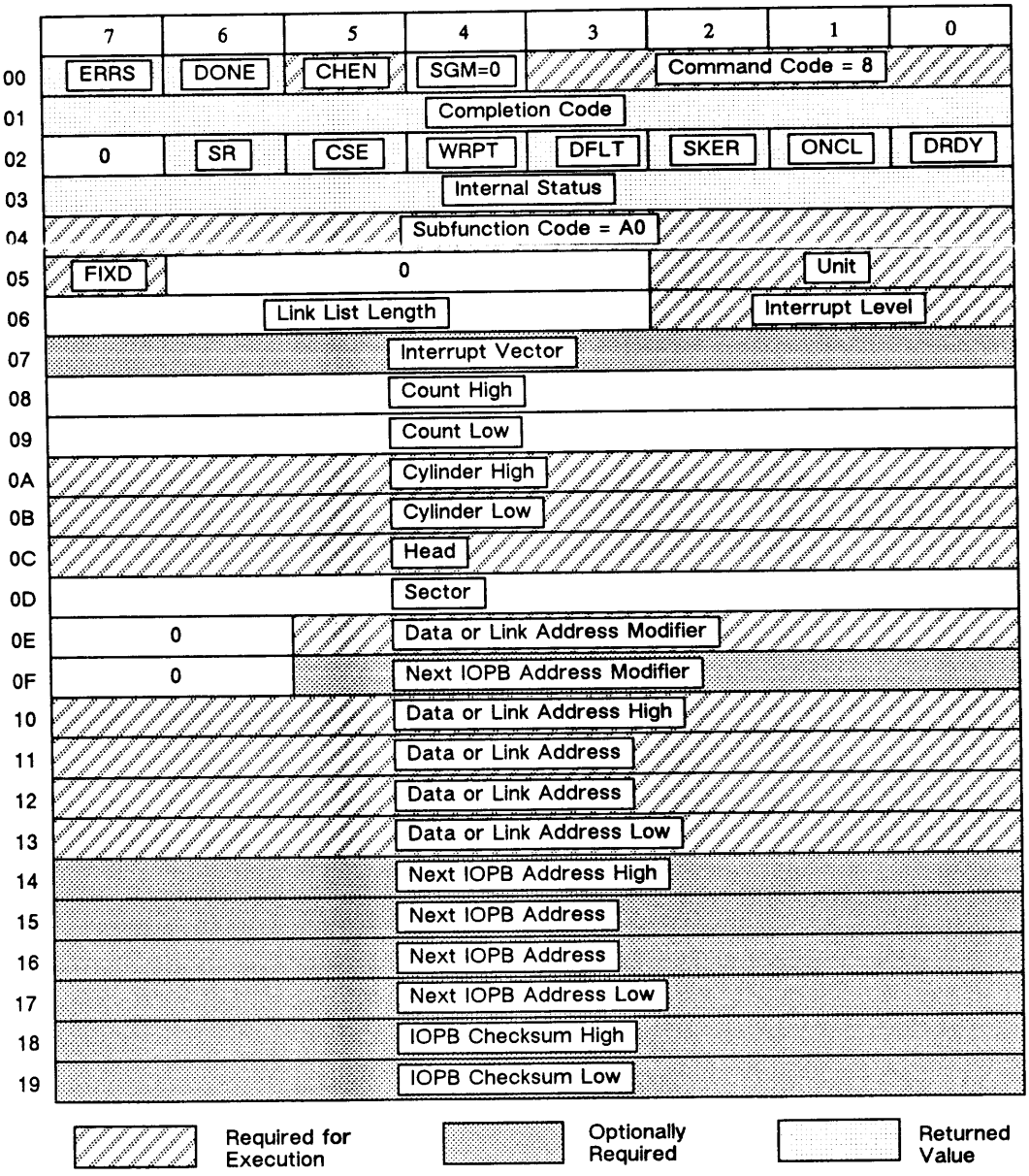
5.22 Read Header, Header Verify, Data, and Data ECC

The 753 reads a sector header, header verify, data, and data ECC. The header always has four bytes, but the other fields vary according to the initial 753 set-up. The 753 reads the physical sectors regardless of the interleave factor; it does not cross head or cylinder boundaries (see Section 8.14).



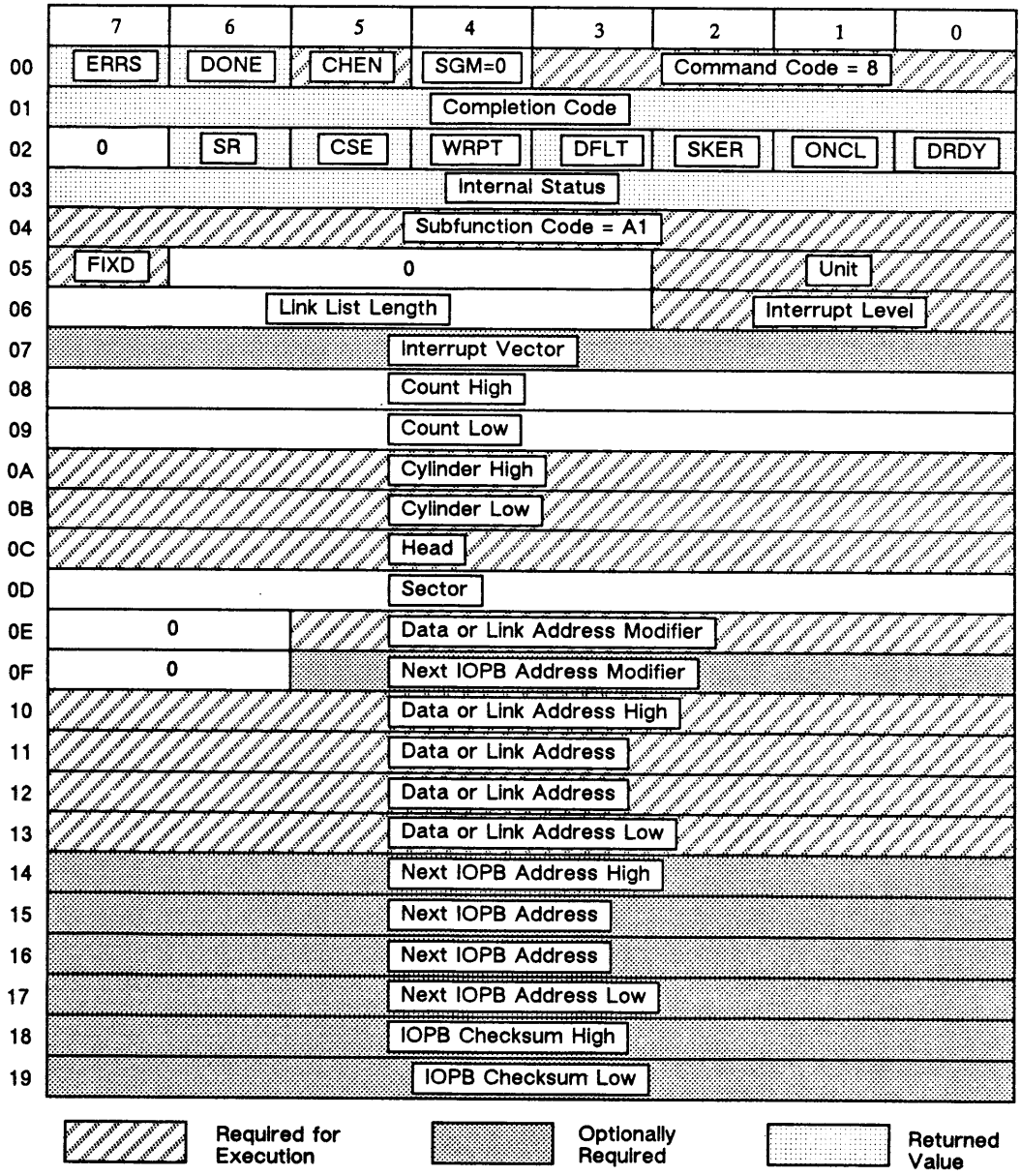
5.23 Read Defect Map

The 753 reads the manufacturer's defect map and returns the data to memory in the correct bit order (see Section 8.1).



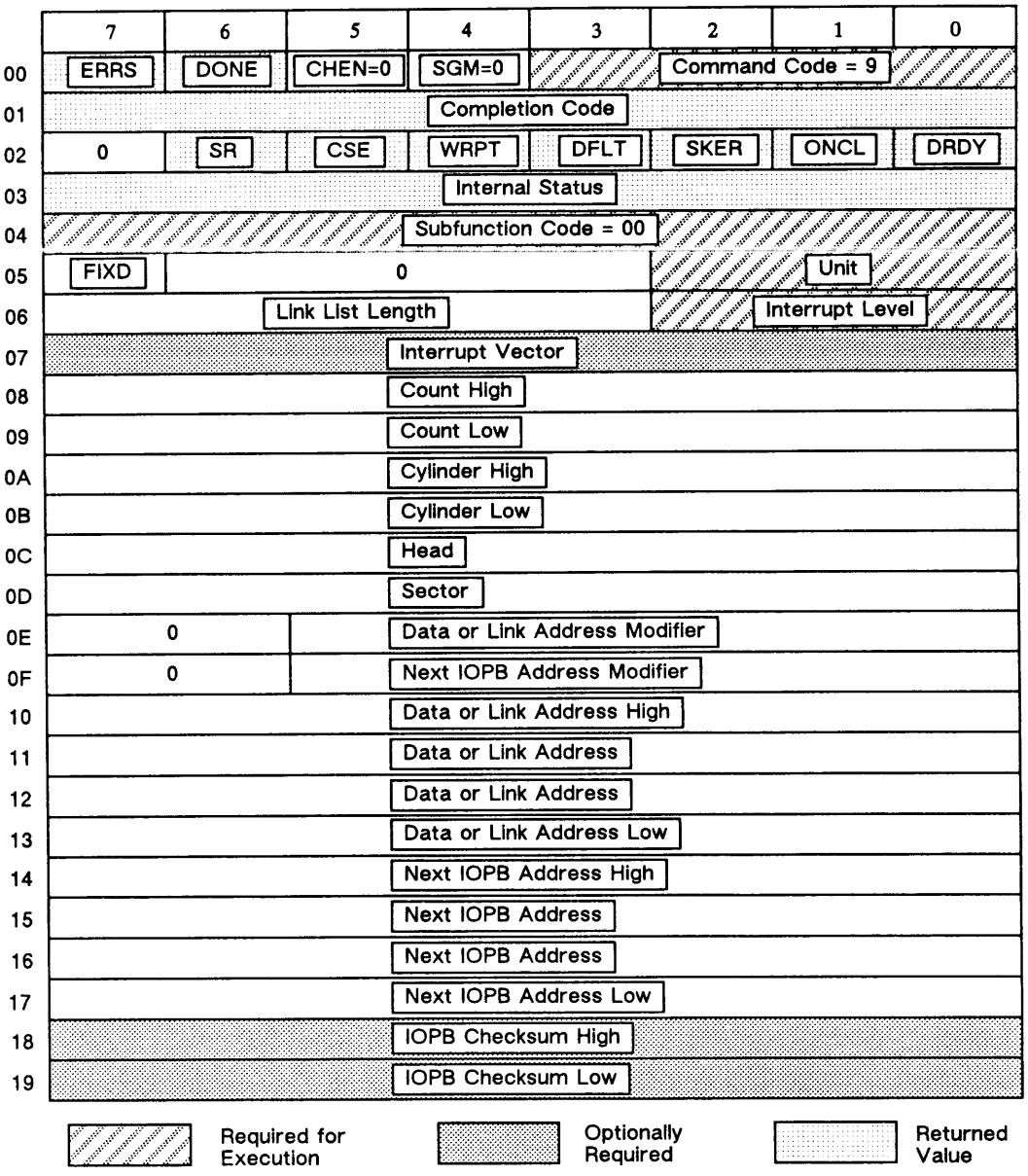
5.24 Read Defect Map Extended

The 753 bypasses any media defects and reads the manufacturer's defect map from the alternate location on the track (see Section 8.1).



5.25 Diagnostics

The 753 executes the on-board self test diagnostics. Do not chain this IOPB; it cannot be used with others in the command queue.



Section 6: Error Processing

Section 6: Error Processing

6.0 Introduction

The *Error Summary* (ERRS) bit, *Fatal Error* (FERR) bit, Fatal Error Register, and completion code represent the 753's status after executing a command. FERR indicates the transfer failed and the 753 requires a *Controller Reset* before continuing. ERRS only affects the specific IOPB and may be tested instead of checking the completion code; the 753 does not require a *Controller Reset* before continuing. The completion code informs software that the 753 successfully completed a command, failed to complete a command, or encountered and corrected a problem with one of its internal recovery procedures.

6.1 The Completion Code

The 753 posts a completion code in IOPB Byte 1. A completion code is only valid if *Done* is set. Table 6-2 lists the completion codes. All codes not listed in this table are reserved. Table 6-3 lists the returned values for the various codes. The following subsections describe these codes and any required corrective action.

6.1.1 Completion Code Convention

Completion codes follow a convention that recommends the action required by either the software driver or manual intervention. The byte's upper nibble is the recovery code, and the lower nibble is the actual error code (see Table 6-1).

Table 6-1. Recovery Codes

Code	Procedure
0	No Action/Status Only
1	Non-retryable Programming Error
3	Successfully Recovered Soft Error
4	Hard Error/Retry
6	Hard Error/Reset and Retry
7	Fatal Hardware Error
8	Miscellaneous Error
9	Requires Manual Intervention

6.1.1 Completion Code Convention (continued)

Table 6-2. Summary of Completion Codes

Action	Code (Hex)	Description
No Action/Status Only	00	Successful Completion
Non-retryable Programming Errors	10	Illegal Cylinder Address
	11	Illegal Head Address
	12	Illegal Sector Address
	13	Count Zero
	14	Unimplemented Command
	15-1B	Illegal Field Lengths 1-7
	1C	Illegal Scatter/Gather Length
	1D	Not Enough Sectors Per Track
	1E	Next IOPB Alignment Error
	1F	Scatter/Gather Address Align.
	20	Scatter/Gather With Auto-ECC
Successfully Recovered Soft Errors	30	Soft ECC Corrected
	31	ECC Ignored
	32	Auto-seek Retry Recovered
	33	Soft Retry Recovered
Hard Errors/Retry	40	Hard Data ECC
	41	Header Not Found
	42	Drive Not Ready
	43	Operation Timeout
	44	VMEDMA Timeout
	45	Disk Sequencer Error
	48	Header ECC Error
	49	Read Verify
	4A	Fatal VMEDMA Error
	4B	VMEbus Error
Hard Errors/ Reset/Retry	60	Drive Faulted
	61	Header Error/Cylinder
	62	Header Error/Head
	63	Drive Not On-cylinder
	64	Seek Error
Fatal Hardware Errors	70	Illegal Sector Size
Miscellaneous Errors	80	Soft ECC
Requires Manual Intervention	90	Write-protect Error

6.1.1 Completion Code Convention (continued)

Table 6-3. Returned Values for Completion Codes

Code	Name	Type	Count	Disk Addr	Data Addr
10	Illegal Cylinder Address	Prog	N/C	N/C	N/C
11	Illegal Head Address	Prog	N/C	N/C	N/C
12	Illegal Sector Address	Prog	N/C	N/C	N/C
13	Count Zero	Prog	N/C	N/C	N/C
14	Unimplemented Command	Prog	N/C	N/C	N/C
15-1B	Illegal Field Lengths 1-7	Prog	N/C	N/C	N/C
1C	Illegal Scatter/Gather Length	Prog	N/C	N/C	N/C
1D	Not Enough Sectors Per Track	Prog	N/C	N/C	N/C
1E	Next IOPB Alignment Error	Prog	N/C	N/C	N/C
1F	Scatter/Gather Address Align.	Prog	N/C	N/C	N/C
20	Scatter/Gather With Auto-ECC	Prog	Err	Err	xx
30	Soft ECC Corrected	Soft	Std	Std	Std
31	ECC Ignored	Soft	Std	Std	Std
32	Auto-seek Retry Recovered	Soft	Std	Std	Std
33	Soft Retry Recovered	Soft	Std	Std	Std
40	Hard Data ECC	Retry	Err	Err	Err
41	Header Not Found	Retry	Err	Err	Err
42	Drive Not Ready	Retry	xx	xx	xx
43	Operation Timeout	Retry	xx	xx	xx
44	VMEDMA Timeout	Retry	xx	xx	Err
45	Disk Sequencer Error	Retry	Err	Err	Err
48	Header ECC Error	Retry	Err	Err	Err
49	Read Verify	Retry	Err	Err	Err
4A	Fatal VMEDMA Error	Hdwe	xx	xx	Err
4B	VMEbus Error	Hdwe	xx	xx	Err
60	Drive Faulted	Reset	xx	xx	xx
61	Header Error/Cylinder	Reset	Err	Err	Err
62	Header Error/Head	Reset	Err	Err	Err
63	Drive Not On-cylinder	Retry	N/C	N/C	N/C
64	Seek Error	Retry	Err	Err	Err
70	Illegal Sector Size	Hdwe	N/C	N/C	N/C
80	Soft ECC	Misc	Err	Err	Err
90	Write-protect Error	Manual	N/C	N/C	N/C

Key:

N/C	No change from original IOPB
Err	IOPB bytes reflect error (i.e., DMA address = address of failure)
Std	IOPB bytes change based on normal Auto-update
Retry	Retryable IOPB
xx	Unpredictable output

6.1.2 Completion Code Descriptions

6.1.2.1 No Action/Status Only

The following completion code usually requires no action; the 753 returns it for status only.

Code(H)	Description
00	*Successful Completion* – Not an error; indicates the IOPB is complete.

6.1.2.2 Non-retryable Programming Errors

This group of errors is usually encountered while debugging drivers; they should not occur in a normal operating system environment.

Code(H)	Description
10	*Illegal Cylinder Address* – Host software specified a cylinder address greater than the maximum cylinder number specified in the last Set Drive Parameters command for this drive.
11	*Illegal Head Address* – Host software specified a head address greater than the maximum head address specified in the last Set Drive Parameters command for this drive.
12	*Illegal Sector Address* – Host software specified a sector address greater than the maximum sector number specified in the last Set Drive Parameters command for this drive.
13	*Count Zero* – Host software issued the 753 an IOPB that required a count, but the count was zero. Read, Write, and Format commands require a valid count.
14	*Unimplemented Command* – This error occurs on all reserved commands.
15	*Illegal Field Length 1* – See Section 8.2.2.1.
16	*Illegal Field Length 2* – See Section 8.2.2.2.
17	*Illegal Field Length 3* – See Section 8.2.2.3.
18	*Illegal Field Length 4* – See Section 8.2.2.4.
1A	*Illegal Field Length 6* – See Section 8.2.2.5.
1B	*Illegal Field Length 7* – See Section 8.2.2.6.

6.1.2.2 Non-retryable Programming Errors (continued)

Code(H)	Description
1C	*Illegal Scatter/Gather Length* – The linked list specified a number of words to transfer that does not agree with the amount of data contained in the requested number of sectors for transfer.
1D	*Not Enough Sectors Per Track* – The format routine could not format because too few sectors were available on the track.
1E	*Next IOPB Alignment Error* – The Next IOPB address did not start on a 16-bit boundary; the 753 does not execute the Next IOPB.
1F	*Scatter/Gather Address Alignment Error* – A scatter/gather address did not start on a 16–bit boundary.
20	*Scatter/Gather With Auto-ECC Error* – A scatter/gather ended with a soft ECC error. Due to scatter/gather boundaries, the 753 did not automatically correct the error, but reverted to ECC Mode 0.

6.1.2.3 Successfully Recovered Soft Errors

This group of errors reflects status only. If some errors recur often, the operating system should try to map out the sectors involved. Recurring errors degrade performance. The 753 does not update the IOPBs for successfully recovered soft errors.

Code(H)	Description
30	*Soft ECC Corrected* – During a disk read in ECC Mode 2, the 753 detected and corrected one or more ECC errors.
31	*ECC Error Ignored* – During a disk read in ECC Mode 1, the 753 detected but ignored an ECC error and continued the transfer.
32	*Auto–seek Retry Recovered* – The 753 successfully completed the transfer but, during the transfer, it recovered from an error by resetting the drive.
33	*Soft Retry Recovered* – The 753 encountered an error while executing this command. An RBC retry was successful.

6.1.2.4 Hard Errors - Retry

These errors indicate the transfer failed; retry the operation. If several retries fail, you must intervene or the operating system may crash.

Code(H)	Description
40	*Hard Data ECC Error* - The 753 detected a hard data ECC error in the data field during a Read command. Retry the previous Read operation.
41	*Header Not Found* - The 753 cannot find the requested sector; it searches for one disk revolution plus one sector to locate the header (see Section 9.3.2).
42	*Drive Not Ready* - The selected drive is not ready, but not faulted; issue a Drive Reset. Possible causes include: o Drive not up-to-speed. o Drive hardware error. o Bad or improperly connected cable(s). o No drive of the specified unit number is connected to the 753.
43	*Operation Timeout* - The 753 did not complete the IOPB within the two second timeout period.
44	*VMEDMA Timeout* - The DMA controller did not complete its task within the allotted time limit. Possibly, memory did not respond in time.
45	*Disk Sequencer Error* - The disk sequencer did not complete its task within the allotted time limit. The 753 cannot send or receive the appropriate signals from the selected drive. Some causes include: improper or defective cabling, or an unformatted drive.
48	*Header ECC Error* - The 753 found a header, but the ECC did not match.
49	*Read Verify Error* - The disk data did not match the data from memory.
4A	*Fatal VMEDMA Error* - The VMEDMA stopped for no apparent reason. Neither the count or the address overflowed, and there was no bus error.
4B	*VMEbus Error* - The VME *BERR** signal was asserted while the 753 was bus master (see the VMEbus Specification).

6.1.2.5 Hard Errors - Reset/Retry

This group of errors indicate the transfer failed. Software should issue a Drive Reset to the drive in use before retrying the operation.

Code(H)	Description
60	*Drive Faulted* - The selected drive is faulted; issue a Drive Reset. If the fault persists, you must intervene.
61	*Header Error/Cylinder* - The cylinder address did not match during a sector search. Check the cylinder address and retry the operation.
62	*Header Error/Head* - The head address did not match during a sector search.
63	*Drive Not On-cylinder* - At some point during the transfer, the 753 expected the drive to be *on-cylinder*, and it was not.
64	*Seek Error* - The disk drive reported a *seek* error.

6.1.2.6 Fatal Hardware Errors

These errors indicate the hardware failed. Manual intervention or a *Controller Reset* may be the only recovery approaches that work.

Code(H)	Description
70	*Illegal Sector Size* - The disk drive's sector size is not large enough to hold the header, data, and specified field lengths.

6.1.2.7 Miscellaneous Errors

Code(H)	Description
80	*Soft ECC Error* - During a disk read in ECC Mode 0, the 753 detected a soft error in the current sector's data field. Software must perform the final correction (see Section 6.3).

6.1.2.8 Requires Manual Intervention

The write-protect error requires that you manually remove the write-protection.

Code(H)	Description
90	*Write-protect Error* – A command that writes to the disk (e.g., Write, Format, Write Track Headers) is issued, but the drive is write-protected.

6.2 Soft Error Completion Codes

The 753 updates the IOPB with the last error it encounters; it may overwrite previous soft errors with a new soft or hard error status. The controller does not update the IOPBs for successfully recovered soft errors (see section 6.1.2.3).

6.3 Error Correction Code

Most error correction code (ECC) algorithms require retrying the operation at least once before attempting the correction. When RBC is set, the 753 automatically retries the operation once before applying the correction algorithm.

6.3.1 Error Correction Code – Mode 0

When using Mode 0, the following procedure corrects a soft ECC error. The 753 provides a pattern and offset for the correction process.

1. Reserve 32 bits of storage for the shifted ECC pattern, and initialize them to zero. Take the ECC pattern word from the IOPB and put it in the lowest 16 bits of the storage.
2. Get the offset from the IOPB and decrement by one. This makes the count zero-based instead of one-based.
3. Shift the pattern the number of count bits left, deriving the count from the offset's three low order bits.

6.3.1 Error Correction Code - Mode 0 (continued)

4. Divide the bit address by eight by performing three logical shifts to the right. The result is the word offset into the bad sector. Adding this offset to the starting memory address of the sector in error creates a pointer to the first word for correction.

5. Exclusive-OR the three memory bytes at the pointer and the two pattern words generated in Step 1.

6.3.2 Error Correction Code - Mode 1

The 753 does not correct any detected errors in Mode 1. After completing the operation, it posts a completion code indicating that at least one ECC error occurred during the transfer.

6.3.3 Error Correction Code - Mode 2

The 753 automatically corrects a soft ECC error in Mode 2. The 753 determines the pattern and offset, completes the DMA, and fetches the data in error from host memory; then it corrects the data, and returns it to memory.

6.4 Fatal Error Codes

If a fatal error occurs, the 753 sets FERR in the Status Register and posts the error code in the Fatal Error Register. The following error codes appear only in the Fatal Error Register. The only way to clear a fatal error is by issuing a *Controller Reset* (CRST).

Code(H)	Description
E0	*Reserved.*
E1	*IRAM Self Test Failure* - The 753 tests the IRAM with an incrementing data pattern, and then tests it with a decrementing pattern. An error indicates a bad IRAM.
E2	*Reserved.*

6.4 Fatal Error Codes (continued)

Code(H)	Description
E3	*Maintenance Test 3 Failure* – The 753 tests the Writable Control Store in the DSKCEL with an incrementing data pattern then tests it with a decrementing pattern. An error indicates a bad DSKCEL.
E4	*Maintenance Test 4 Failure* – The 753 shifts a pattern of zeros and ones through the Header Shift Register (HSR). An error indicates a bad HSR.
E5	*Maintenance Test 5 Failure* – The 753 writes, and then reads, the VMEDMA Registers. An error indicates a bad VMEDMA.
E6	*Maintenance Test 6 Failure* – There is a problem with the REGCEL chip.
E7	*Reserved.*
E8	*Maintenance Test 8 Failure* – The 753 fills the FIFO with sequential data and then reads it. An error indicates a problem with the DSKCEL or FIFO.
F0	*IOPB Checksum Miscompare* – The generated checksum did not match the appended checksum and ICS is set (see Section 8.12).
F1	*IOPB DMA Fatal* – The 753 did not complete the DMA within the allotted timeout period. The memory could be defective or not present; the 753 may not have been able to become bus master.
F2	*IOPB Address Alignment Error* – The IOPB address did not start on a 16-bit boundary. Change the IOPB address and retry the operation.
F3	*Firmware Error* – Flag settings or counter values are inconsistent with the firmware routines; the IOPB cannot DMA the appropriate error status. The 753's state is indeterminate; you must issue a *Controller Reset*.
F5	*Illegal Maintenance Mode Test Number* – The command is invalid, or the maintenance mode jumper is not in.
F6	*ACFAIL Asserted* – The VMEbus signal *ACFAIL* is asserted, causing the 753 to stop. Correct the problem asserting *ACFAIL* and then reset the 753.

Section 7:
A Programming Tutorial

Section 7: A Programming Tutorial

7.0 Introduction

This tutorial programming procedure begins with a single NOP command and progresses to normal Read and Write commands. Each section builds on the previous section's information. In the sent/returned portion of each sample IOPB, the x represents an indeterminate value that depends on the external conditions.

7.1 No Operation (NOP)

The NOP command familiarizes you with the 753 programming interface (see Figure 7-1).

7.1.1 Allocating Memory for an IOPB

First, allocate space in host memory for the IOPB. This allocation is a function of the operating system or the program running. Next, set up the IOPB for a simple NOP.

7.1.2 Point the 753 to the IOPB

The IOPB is now in host memory. Point the 753 to the IOPB by loading the IOPB address and address modifier into the appropriate 753 registers. The 753 looks for the IOPB at the physical address to which the registers point.

> *Make sure the address compensates for any memory mapping that may be done between virtual and physical addressing in your system.*

7.1.3 Executing the NOP IOPB

The 753 now points to the IOPB in host memory. Writing AIO in the CSR directs the 753 to process the IOPB.

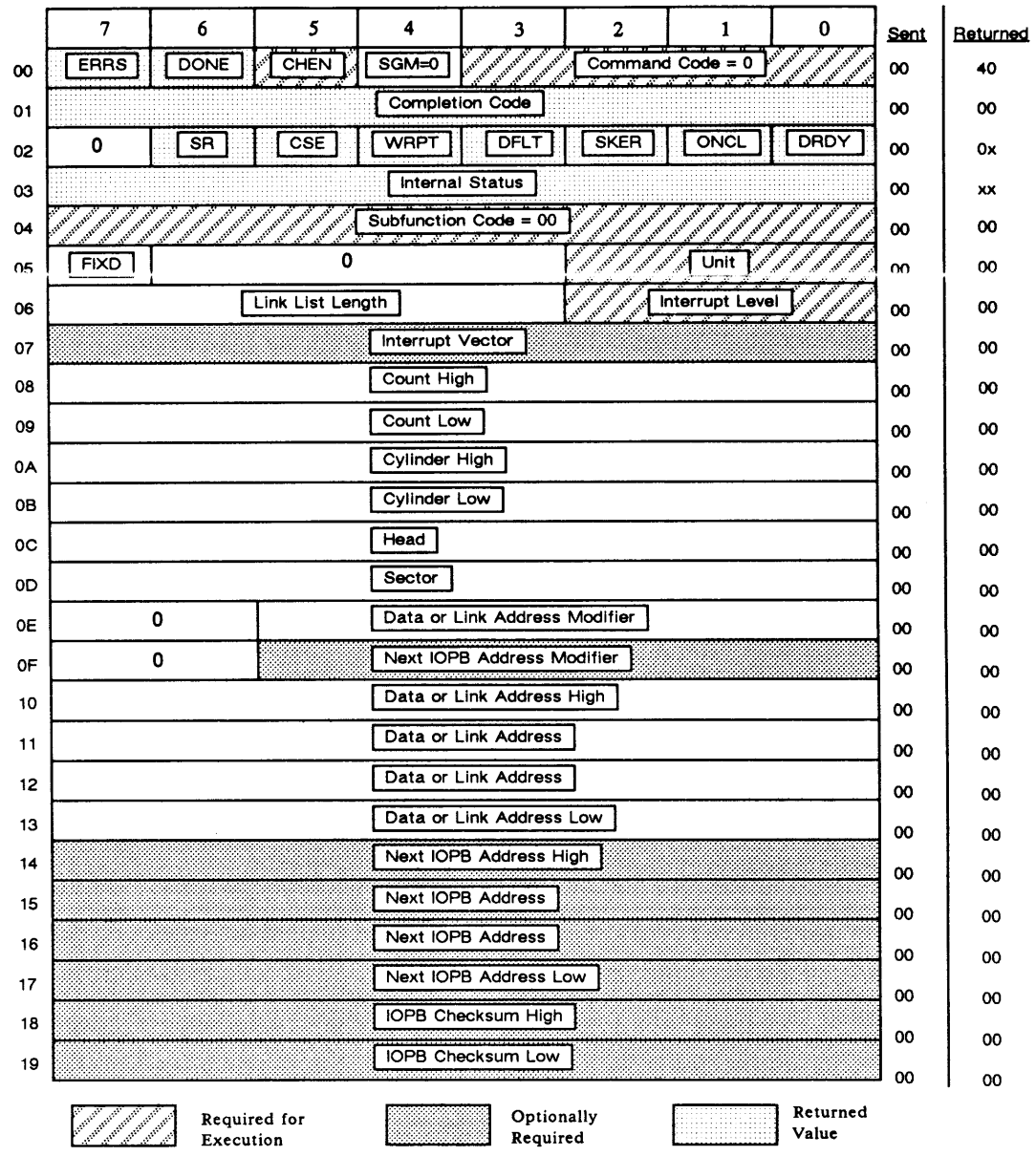


Figure 7-1. Sample NOP IOPB

7.1.3 Executing the NOP IOPB (continued)

At this point, the 753 performs the following functions:

1. Clears AIOP and sets *Busy*.
2. Reads the IOPB from host memory.
3. Decodes the command.
4. Performs the operation (NOP).
5. Sets *Done*.
6. Updates the IOPB.
7. Puts the completed IOPB's address into the registers.
8. Sets RIO.
9. Clears *Busy*.

7.1.4 Command Completion

Software has been polling RIO (since interrupts are not enabled [Interrupt Level = 0]). The 753 sets RIO when it finishes. Software should get the completed IOPB's address from the registers, and then clear RIO. This completes the NOP command.

> *Do not poll Done in the IOPB. The 753 sets Done before it finishes updating the rest of the IOPB.*

7.1.5 Returned Values

Done is set in the returned IOPB. Status Byte 2 reflects the status of Disk Drive 0. Status Byte 3 reflects the 753's internal status.

> *Status Byte 3 is proprietary to Xylogics and may change definition without notice.*

7.2 Read Controller Parameters

Next, implement a read parameters with a controller parameters subfunction (see Section 4.2). This command returns several controller parameters in the IOPB.

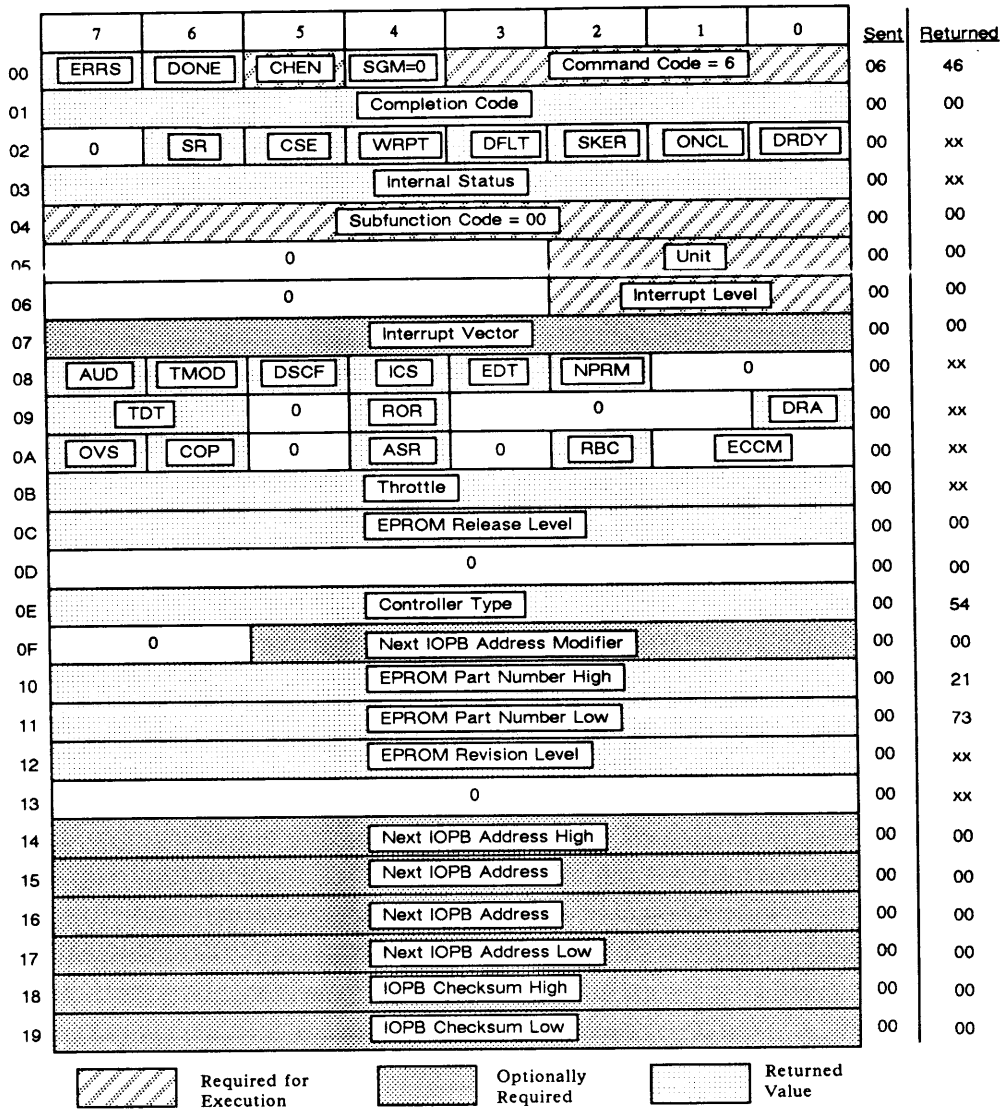


Figure 7-2. Sample Read Controller Parameters IOPB

7.2.1 Executing the Read Controller Parameters IOPB

Set up the IOPB in host memory; point the 753 to the IOPB. Set AIO and the 753 executes the IOPB in Figure 7-2.

The controller operation changes slightly from the example in Section 7.1.3: the 753 performs the Read Controller Parameters operation instead of the NOP; it gets the parameters from its internal store, puts them in the proper IOPB locations, and then fully updates the IOPB, including the returned values.

7.2.2 The Returned IOPB

The values in the returned IOPB describe the last setting of the software-programmable parameters. Determine if each value works for your application. After making any necessary changes, write the parameters back to the 753.

Specific bytes have known values. The Controller Type byte contains a 52H; the EPROM Part Number bytes contain 21H and 73H (see Section 4.2).

7.3 Write Controller Parameters

Next, write the controller parameters. Xylogics recommends reading the current parameters, modifying the ones in question, and then writing them back to the 753. This method allows you to change only those parameters that affect your system (see Figure 7-3).

7.3.1 Executing the Write Controller Parameters IOPB

The 753 executes the IOPB slightly different than in Sections 7.1.3 and 7.2.1: it performs this function by taking the values of all programmable parameters out of the IOPB and setting the appropriate flags and variables in its internal code (see Figure 7-3).

7.3.1 Executing the Write Controller Parameters IOPB (continued)

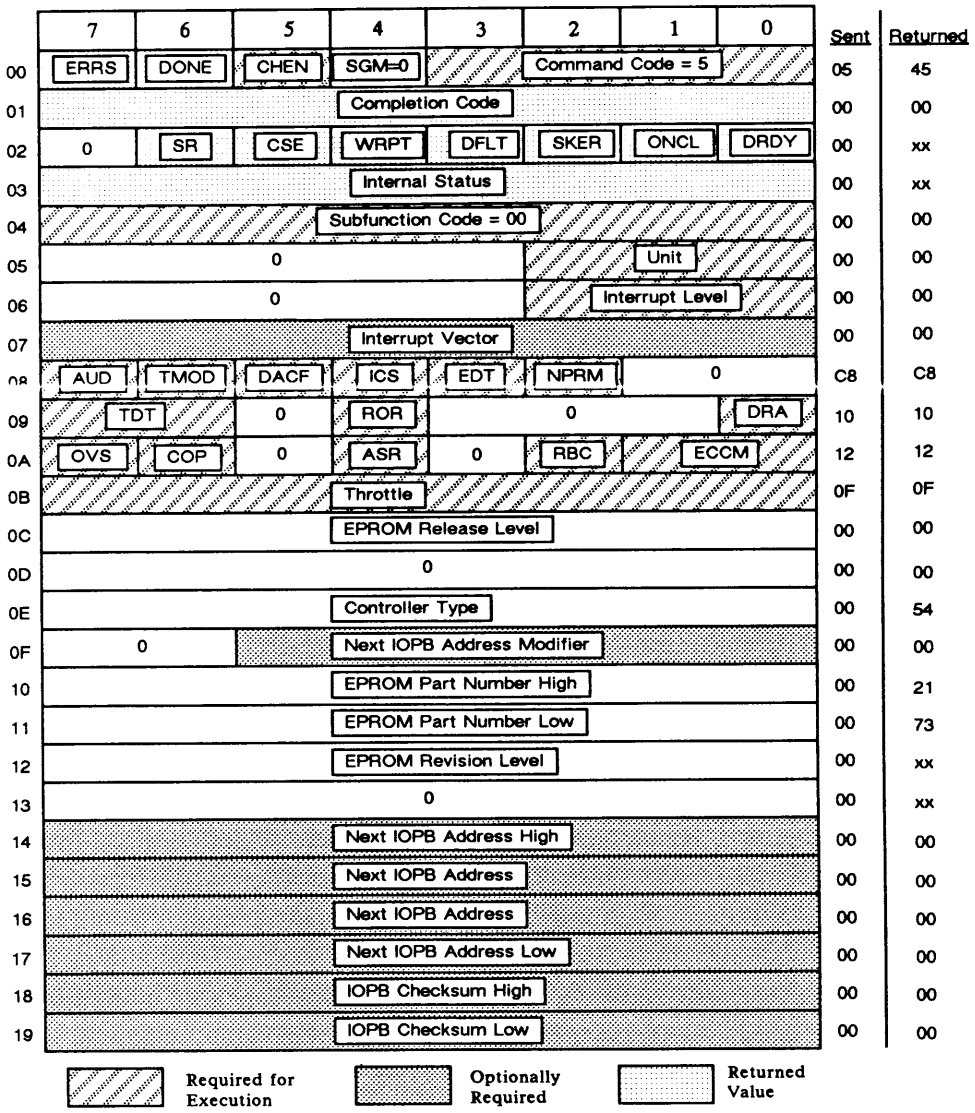


Figure 7-3. Sample Write Controller Parameters IOPB

7.4 Read/Write Format Parameters

The 753 handles format parameters like controller parameters. Use extra caution when modifying the format parameters as improper selection may cause data corruption and/or unreliable operation. The only parameter Xylogics recommends changing is the size of the data field.

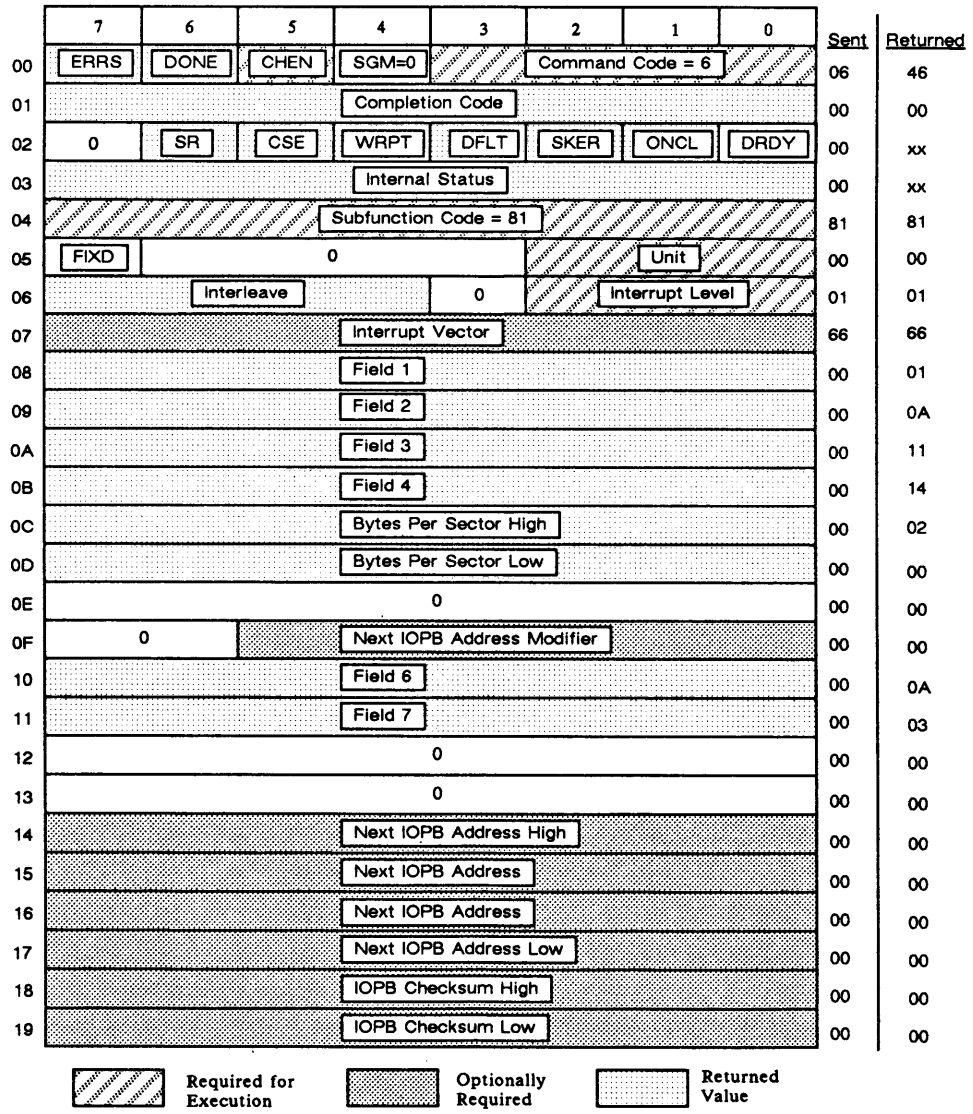


Figure 7-4. Sample Read Format Parameters IOPB

7.4.1 Executing the Read Format Parameters IOPB with Interrupts

For this example, enable interrupts by specifying an interrupt level and vector.

The 753 performs the operation almost identically to the examples in Sections 7.2.1 and 7.3.1, but with two additional steps. It makes sure the new format parameters are within the valid ranges. After the 753 sets RIO, it performs an interrupt sequence (see Figure 7-4).

7.4.2 Command Completion

Enabling interrupts modifies the command completion. Software does not poll RIO when it is set, but may be off doing something else (probably waiting for an interrupt). When the interrupt occurs, hardware and software execute an interrupt service routine (ISR) and process the interrupt. Hardware resets the actual hardware interrupt during an ISR.

The ISR reads the address of the completed IOPB from the registers, and clears RIO. This completes the Read/Write Format Parameters operation.

7.4.3 Returned Values

Figure 7-4 illustrates the read portion of this subsection. The 753 returns the data for which it was last programmed. The sector size is set to 512 (200H). The other fields are all set to the recommended values.

7.5 Read/Write Drive Parameters

The Read and Write Drive Parameters commands allow you to configure the 753 to your drive's size and parameters (see Section 4.3 and Figure 7-5).

7.5.1 Executing the Write Drive Parameters IOPB

On a Write Drive Parameters command, the 753 performs an operation similar to that of both controller and format parameters. Read Drive Parameters differs in that the 753 returns the number of physical sectors on the drive (see Figure 7-5).

7.5.1 Executing the Write Drive Parameters IOPB (continued)

The 753 selects the disk drive specified in the *Unit Select* bits. It times the interval between index pulses and, using this time value, counts the number of sector pulses. The 753 puts this count (the number of sectors per track) into IOPB Byte E.

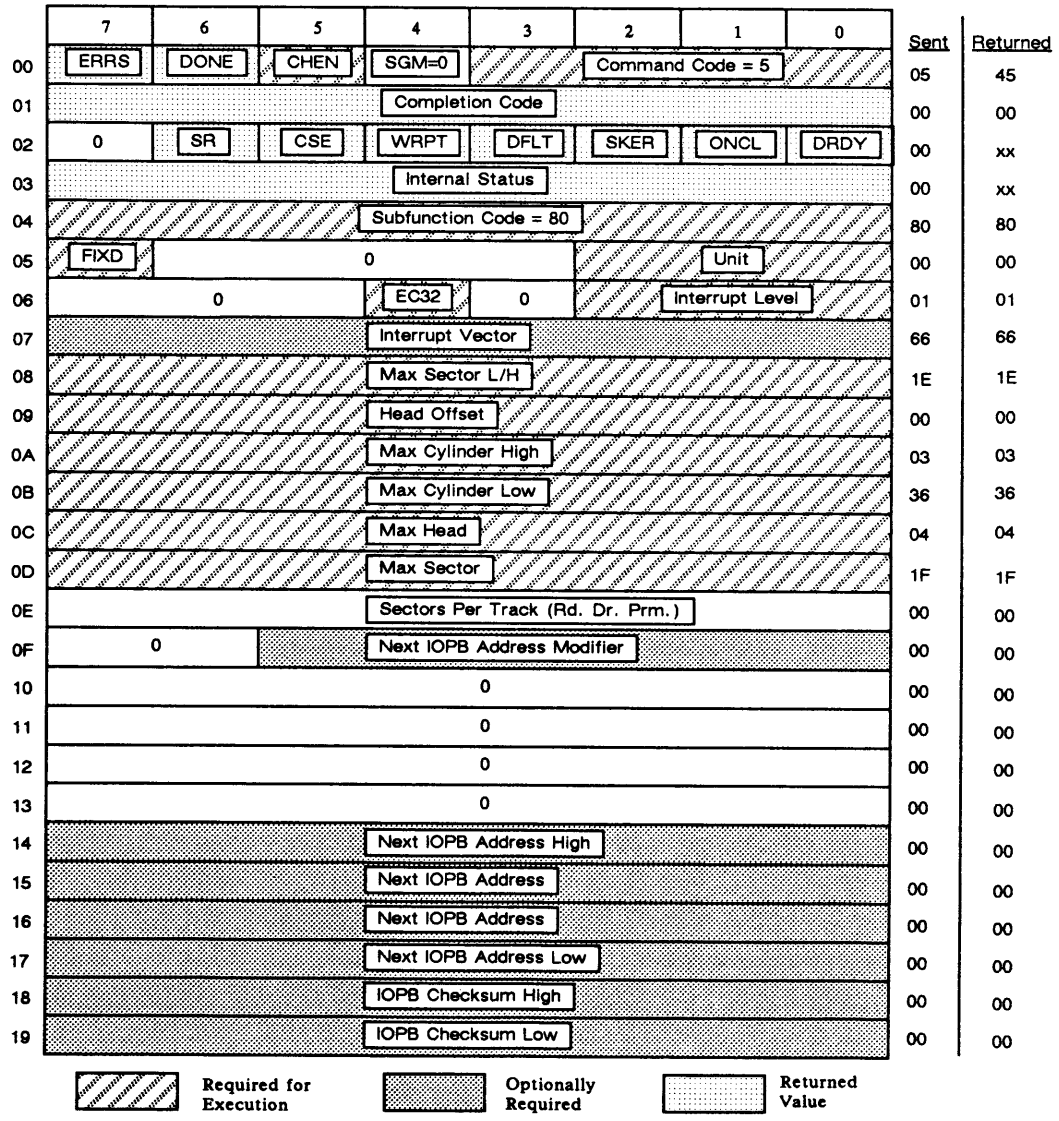


Figure 7–5. Sample Write Drive Parameters IOPB

7.6 Write Format

Up to this point we have been initializing the 753. Initialization specifies the drive size and parameters that the 753 requires before it can properly function. Now, let's format one track of the drive. The 753 can execute Read and Write commands only on a formatted track. Typically, formatting is performed infrequently during the lifetime of the media.

7.6.1 Executing the Write Format IOPB

The Format command is the first command in this tutorial that transfers data from the controller to the disk. The 753 operation for data transfer commands differs greatly from initialization commands (see Figure 7-6).

1. The 753 still clears AIOP, sets *Busy*, and reads the IOPB from memory. The next step occurs after the 753 has the IOPB.

2. The 753 decodes the function, and determines if it requires a seek. All write, read, write extended, and read extended functions require that the drive seek to the requested cylinder. Format is a write extended function; the 753 issues a seek to the drive by sending it the requested cylinder number.

3. The 753 waits for the drive to complete the seek; the drive indicates it's done by returning *on-cylinder*.

4. When the drive is *on-cylinder*, the 753 determines if the current physical sector count for that drive is valid. If the count is not valid, the 753 determines the actual physical sector count by repeating the sequence in Section 7.5.1. The 753 proceeds to Step 5 when the count is valid.

5. The 753 loads the data for the new sector header into the FIFO, waits for index, and writes the new header on Sector 0; it writes the data field with zeros and the ECC.

6. The 753 repeats Step 5 for each sector on the track, except is starts the operation with the sector pulse instead of index.

7. The 753 updates the IOPB with the ending values and completes the command.

7.6.1 Executing the Write Format IOPB (continued)

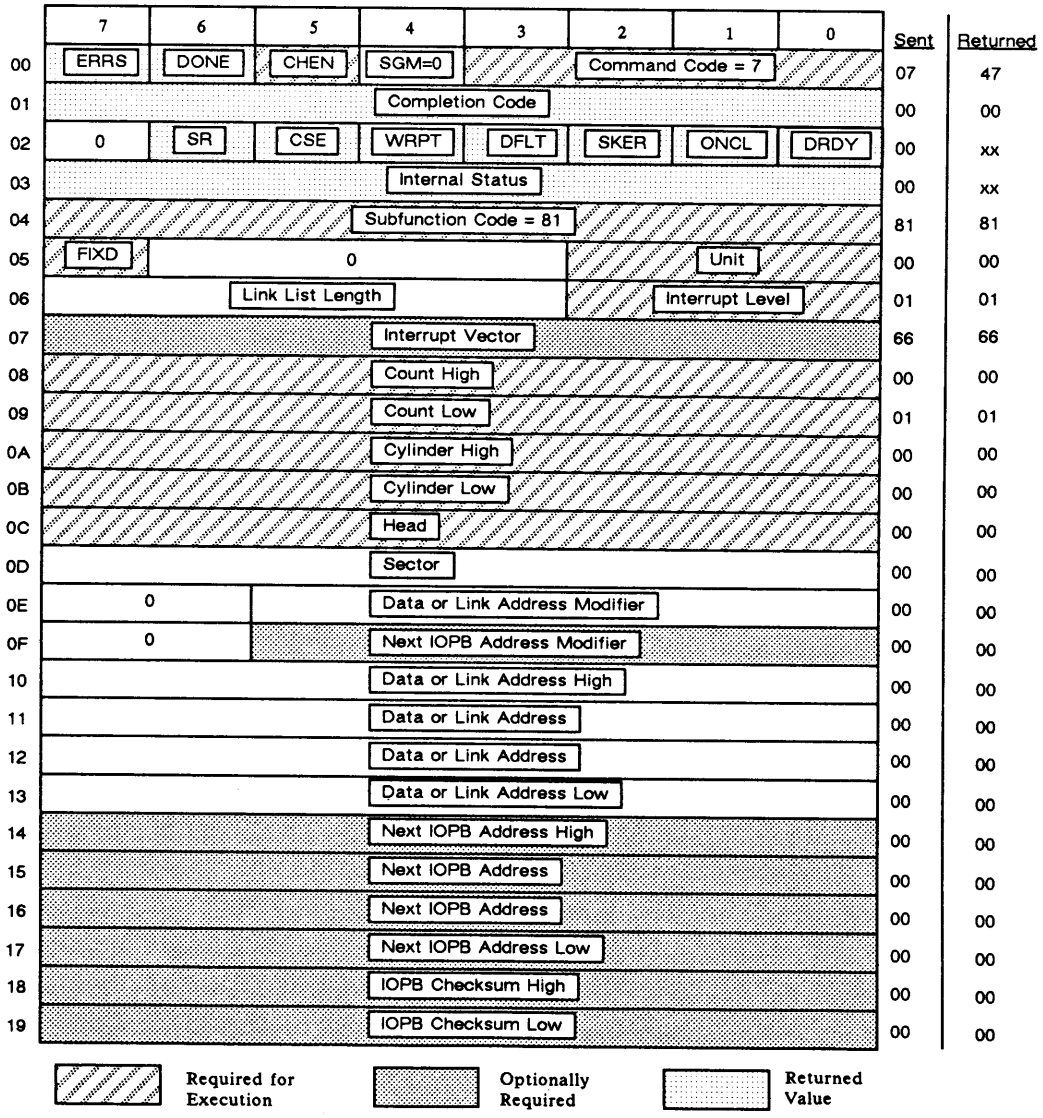


Figure 7-6. Sample Write Format IOPB

7.7 Read Track Headers

Now that the track is formatted, read the headers back and verify them. This command requires a data buffer; allocate space in host memory just as you did for the IOPB. The buffer length must be four bytes (per sector) times the number of sectors per track. The Read Drive Parameters command gives you the number of sectors per track. Make sure software passes the 753 the physical buffer address, not the virtual address (see Figure 7-7).

7.7.1 Executing the Read Track Headers IOPB

Read Track Headers is the first command in this tutorial that transfers data to or from host memory. Data transfers to or from memory modify the 753 operation as follows (see Figure 7-7):

1. The 753 reads and decodes the IOPB, and issues a seek as in the previous examples.
2. The 753 waits for index from the drive. It tests the physical sector count and determines if it is valid; if it is not valid, the 753 determines the actual count (see Section 7.5.1).
3. After index arrives, the 753 synchronizes itself with the data in the header, and reads the data into its on-board FIFO.
4. The 753 repeats Step 3 for each sector on the track, except it waits for the sector pulse instead of index. The actual physical sector count determines the number of sectors the 753 transfers into its FIFO.
5. The 753 begins the DMA after transferring all the sector headers into the FIFO; then it transfers the data from the FIFO to host memory.
6. When the 753 completes the DMA transfer, it places the updated information into the IOPB and completes the command.

7.7.1 Executing the Read Track Headers IOPB (continued)

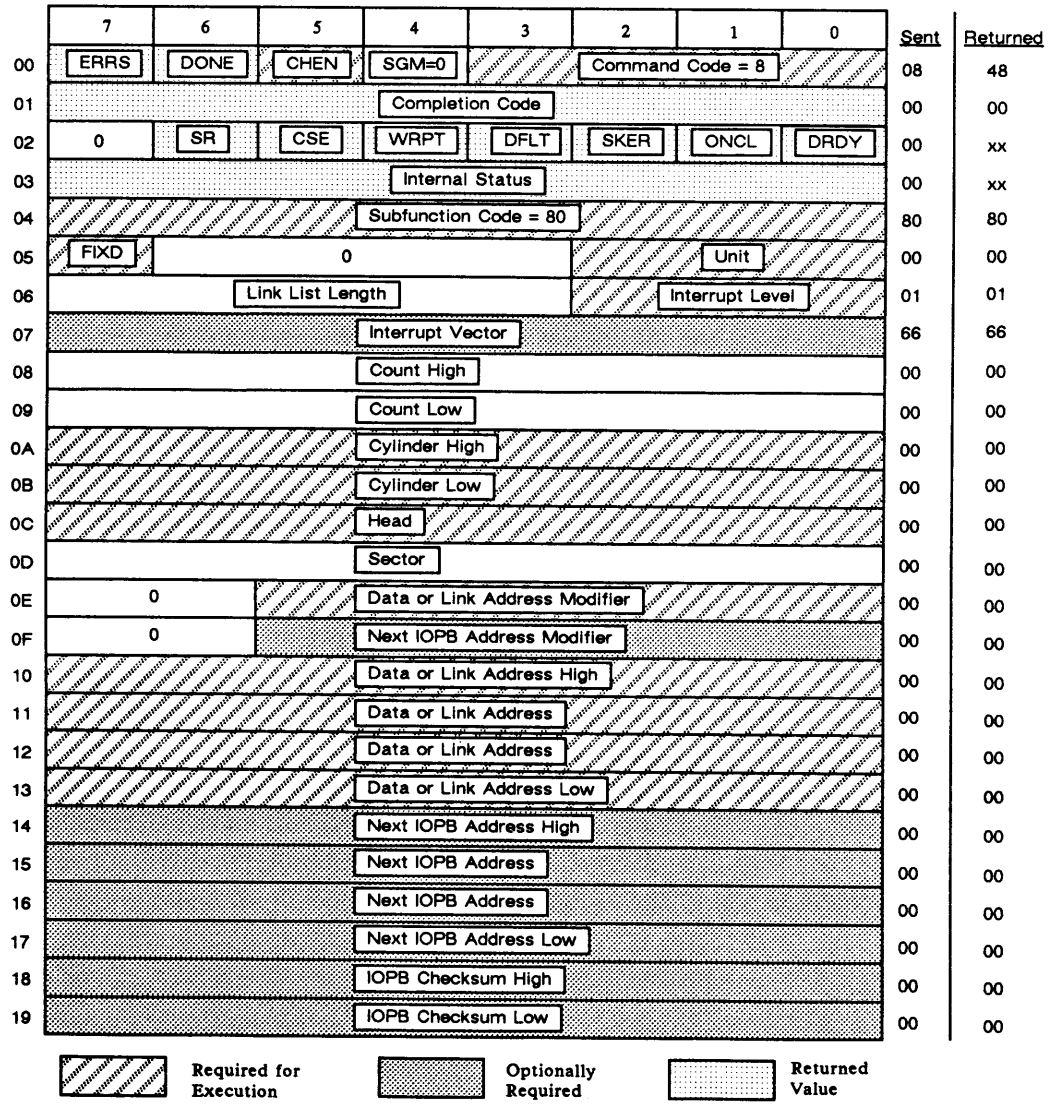


Figure 7-7. Sample Read Track Headers IOPB

7.7.2 Verify the Data

After the 753 completes the transfer, verify the data against the expected values. The data should be divided into groups having four bytes each. Each group describes a sector header. The first sector header should have four bytes of 00; the second header should have three bytes of 00 and one byte of 01; the third header should have three bytes of 00 and one byte of 02, etc. (see Section 8.3).

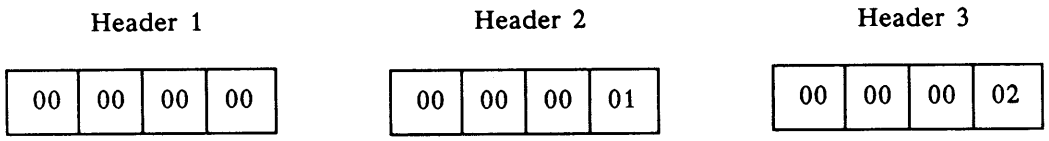


Figure 7-8. Sample Sector Headers

7.8 Write Data

The format is valid after verifying the headers. Allocate space in host memory for the buffer, and set up a data pattern; an incrementing count in the buffer will suffice.

7.8.1 Executing the Write Data IOPB

The 753 operation is similar to the previous examples; the differences are in DMAing data into the FIFO, and writing data to the disk (see Figure 7-9).

The 753 starts the DMA from host memory to the FIFO after determining the drive is *on-cylinder*; it enables the disk sequencer when the FIFO contains one full sector of data.

The 753 compares and verifies the header: the disk sequencer tests all the headers as they pass under the head, until it finds the sector designated for transfer. At the proper point in the sector, the 753 writes a new Sync byte, and then the data it read from memory. The 753, using the data to be written, generates and appends an ECC to the sector.

7.8.1 Executing the Write Data IOPB (continued)

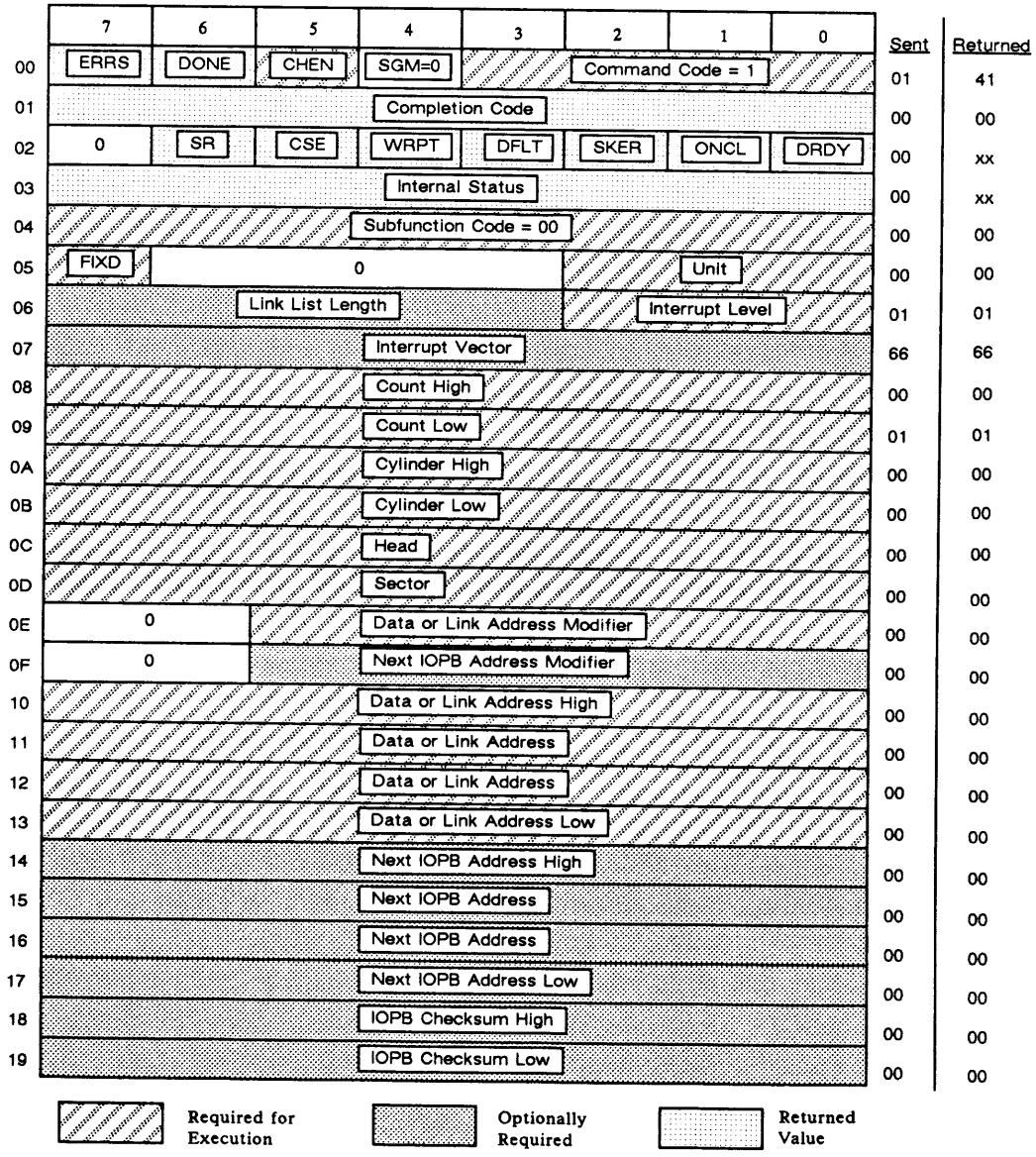


Figure 7-9. Sample Write Data IOPB

7.8.2 Command Completion

The command is complete as soon as the disk sequencer completes its operation. The 753 puts the ending values into the internal IOPB, and performs an appropriate update.

7.9 Read Data

In Section 7.8 the 753 wrote the data to the drive on Sector 0, Head 0, and Track 0. This subsection describes reading back and verifying the data. Allocate space in host memory for the data buffer. After allocation, fill the buffer with a known pattern that differs from the expected data (see Figure 7-10).

7.9.1 Executing the Read Data IOPB

The 753 treats this command like the previous operations, except in the way it DMAs the data into the FIFO and writes the data to the disk.

The 753 enables the disk sequencer as soon as the drive is *on-cylinder*. After the controller finds the correct header, it transfers the data from the disk to the FIFO. As soon as the first sector of data is available from the buffer, the DMA controller DMAs the data from the FIFO to host memory (see Figure 7-10).

7.9.1 Executing the Read Data IOPB (continued)

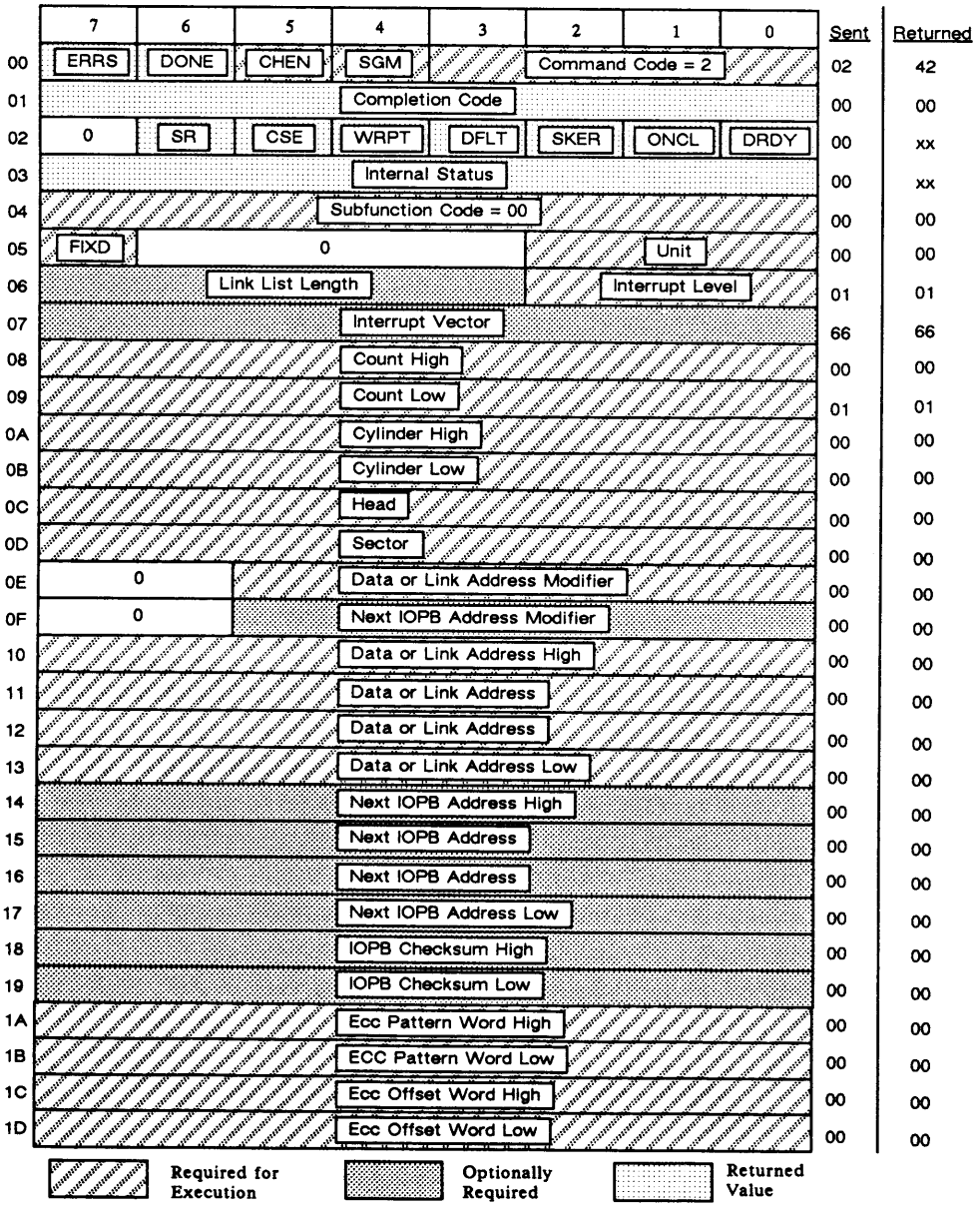


Figure 7-10. Sample Read Data IOPB

7.9.2 Command Completion

The transfer is done when the DMA controller completes the DMA to memory.

7.9.3 Verify the Data

First, make sure the buffer was modified. If it was not modified, either an error occurred, or software specified the wrong buffer address. Next, compare the data written with the data read; they should match.

7.10 Multiple Sector Transfers

You can repeat the steps in Sections 7.8 and 7.9 using a larger sector count. The 753 crosses head and cylinder boundaries, as required, to complete the required number of sectors. Make sure you allocate enough buffer space for the increased sector count.

7.11 Summary

This section was an exercise in testing the 753 in your system. The procedure is similar when the software driver controls the 753. Operating systems always allocate the buffers.

Section 8: Special Functions

Section 8: Special Functions

8.0 Introduction

This section describes how to implement the 753's special functions. Each subsection describes how minor functions implement a given major function.

8.1 Read Defect Map

The 753 reads the data recorded on the media by the manufacturer that describes the location and length of factory-detected media defects. The defect map's format must conform to the M2331/M2333 Disk Drive Engineering Specification, Fujitsu Ltd. Doc. No. B03P-4760-0101A, and CDC Product Specification No. 64400400.

8.1.1 The Defect Map

The header format is divided into two parts: fixed and variable. The fixed sector format is normally included in the first 56 bytes following index. The 753 does not support the variable sector format.

The following rules apply to defect recording:

1. A defect's position is listed in bytes (hex) after the index mark, plus or minus one byte.
2. The length of a defect is in bits (hex), plus or minus one bit.
3. The unused defect locations are all zeros.
4. Every track is recorded with this defect format whether or not the defects exist.
5. A track that has more than one defect may be flagged as defective. Log the first four media defects on the track.

8.1.1 The Defect Map (continued)

6. Figure 8-1 shows the format when there are no defects in the first 105 bytes after index.

7. Figure 8-2 shows the format when the beginning of a defect is located between Bytes 10 and 55; the 753 adds 60 bytes of zeros to Gap 1. In the extended defect map format, the 753 always relocates the defect map to a defect-free area.

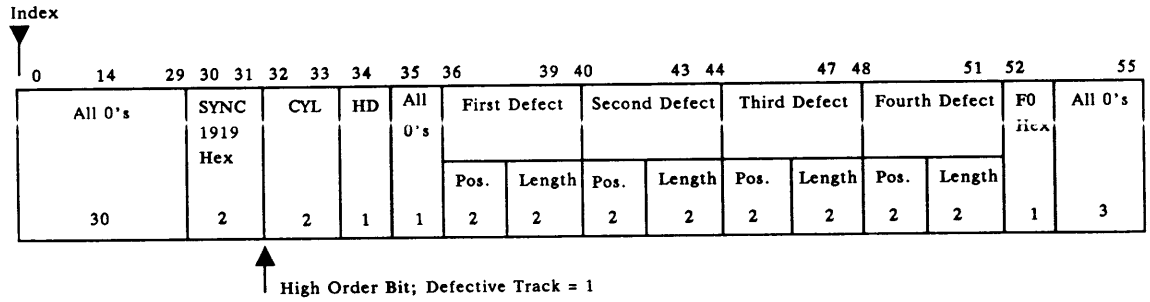


Figure 8-1. Defect Map Format

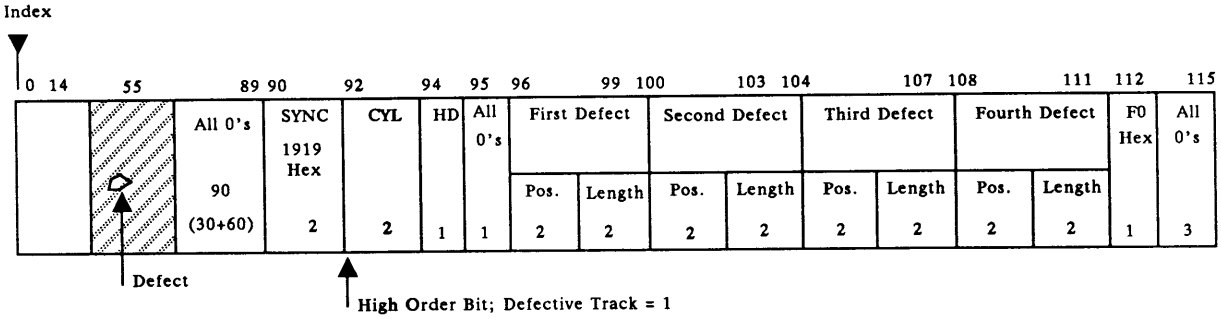


Figure 8-2. Extended Defect Map Format

8.1.2 Read the Defect Map

Set up a 24-byte long buffer in memory. Issue an extended read with a Read Defect Map subfunction. Since the defect maps do not use a checksum, verify the data.

8.1.3 Verify the Data

The first byte in the buffer should equal 19H. This is the second of two Sync bytes. The head and cylinder addresses should agree. The last byte should equal F0H.

8.1.4 Determining the Location of a Defect

The position of a defect is provided in bytes from the index mark. You must convert this value to a sector number before using it for defect mapping. The drive switch settings determine the actual number of bytes per sector. Divide this value into the defect position. This number, truncated, points to the physical sector with the defect. Adjusting this physical sector number by any interleaving or skewing scheme determines the logical sector number for remapping.

8.2 Formatting

This subsection describes formatting, including how to set the number of spares for use in media defect handling and setting the sector size.

8.2.1 Allocating Spare Sectors

You must allocate spare sectors at format time. The Write Drive Parameters command allows setting the maximum size parameters for the drive. The 753 marks any sectors in excess of the drive size parameters as spares. For example, a drive with 46 physical sectors is specified as having only 45 in the Write Drive Parameters command. The 753 formats 45 normal sectors and 1 spare sector.

8.2.1 Allocating Spare Sectors (continued)

A separate variable (max sector/last head) specifies the number of sectors on max head. This allows extra spares on max head for use with cylinder sparing. Given the above example, host software sets the max sector/last head to 40. The 753 formats the last head of the same drive with 40 normal sectors, and 6 spare sectors.

8.2.2 Specify Sector Gap Size

First a note of caution, *don't do this.* Modifying the gap sizes may reduce your disk subsystem's performance. The drive interface has many specifications and parameters to which you must adhere. Slight miscalculations may not show up in the engineering lab, but can cause unreliability in the field. The failure modes may be undetectable because the field lengths are only marginally long enough. A word to the wise: be very careful implementing this feature. Xylogics' values work with all standard SMD drives (see Sections 4.4.6 through 4.4.8). Modified values may work with some drives and cause unreliable results with others (see Figure 8-3).

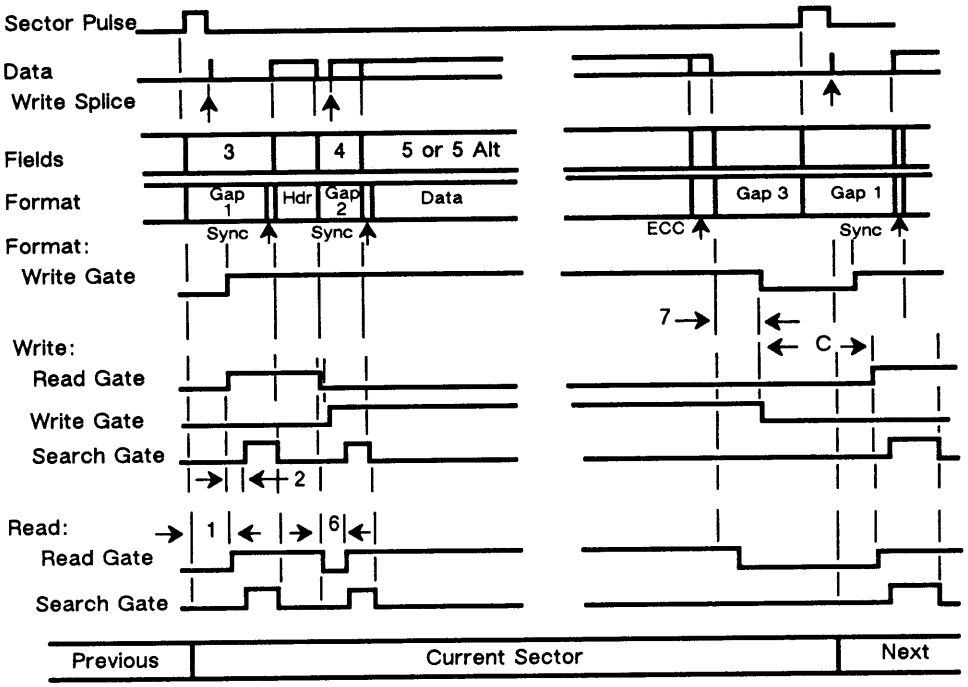


Figure 8-3. Sector Gap Sizes

8.2.2 Specify Sector Gap Size (continued)

The following subsections outline the parameters that affect each of the gap sizes. Many of the parameters that affect the gap sizes are specifications for the drive in use. *Caution: all drives do not have the same specifications.*

8.2.2.1 Field 1 - Read Gate Delay - Gap 1

Field 1 specifies the time, in bytes, from one byte past the leading edge of sector or index to when *Read Gate* is asserted. The applicable drive specifications include head settling time, allowing enough time to lock the phase lock oscillator, and the minimum *Write Gate* to *Read Gate* timing. Field 1 also enables skipping over the write splice area.

The head settling time is the time required for the heads to settle after the drive completes a seek. We do not know how long before the sector or index pulse the seek completed, therefore this field must be large enough to encompass the head settling time.

Field 1, when subtracted from Field 3, indicates the amount of time left for the phase lock oscillator in the drive to lock onto the data. Drive requirements vary from 3 to 16 bytes.

When *Write Gate* is deasserted, a minimum time must be allowed for the read heads and amplifiers to stabilize. In a multisector transfer, *Write Gate* is deasserted after the last sector, and *Read Gate* is asserted for reading the next header. Figure 8-3 indicates this time as *C*. Field 1, and the remaining bytes in the sector after Field 7, comprise this critical time. Drive requirements are usually in the 10 to 12 microsecond range.

Write splice: as *Write Gate* is asserted (or deasserted), the changing write current causes a magnetic field to build (or collapse) that *writes* garbage onto the disk. This field is called the write splice, and it occurs whenever *Write Gate* is asserted or deasserted.

8.2.2.2 Field 2 - Sync Search Delay

Field 2 is the delay from asserting *Read Gate* to comparing data for the Sync byte. Field 2 masks any read data from being detected as a sync until the data is stable.

8.2.2.3 Field 3 - Gap 1

Field 3 is equivalent to Gap 1. Fields 3 and 1 together define the amount of time allowed for the phase lock oscillator (PLO) in the drive to lock up (see Section 8.2.2.1).

8.2.2.4 Field 4 - Gap 2

Field 4 is equivalent to Gap 2. It is the time, in bytes, between the end of the Header ECC and the Data Sync byte. This field includes the time required for the PLO in the drive to sync to the data and skip the write splice. On a Write command, *Write Gate* is asserted four bit cells after deasserting *Read Gate*. This field is actually four bits longer than specified. These four bits are for testing the header.

8.2.2.5 Field 6 - Read Gate Delay

Field 6 asserts *Read Gate* after the write splice during Gap 2. The time left over, Field 4 minus Field 6, is the time allowed for the PLO in the drive to lock up.

8.2.2.6 Field 7 - Write Continuation

The write continuation field is necessary so that when *Write Gate* is deasserted, the collapsing magnetic field does not splash over the ECC that was just written. The end of Field 7 is the beginning of the minimum *Write Gate* to *Read Gate* period described in Section 8.2.2.1.

8.2.2.7 Head Switch Time

The 753 requires 6 bytes of time after Field 7 to switch heads. If the sector size is too small, the 753 misses revolutions on every head switch. Typically, the minimum *Write Gate* to *Read Gate* time allows for this 6-byte field.

8.2.3 Format Interleave

The 753 can optionally format with an interleave pattern from 2:1 to 16:1. Specify the interleave factor when writing the format parameters; it is invisible to the operating system.

8.2.3 Format Interleave (continued)

Interleaving can increase the throughput of a disk subsystem on a fully loaded system by effectively cutting the disk speed in half. On a contiguously formatted pack (1:1 interleave), the sectors increase by one each time. As the disk spins, the sectors arrive under the head in the following order for a 32-sector disk:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 ... 31

If you interleaved the same disk with a 2:1 interleave factor, it would look like this:

0 16 1 17 2 18 3 19 4 20 5 21 6 22 7 23 8 24 9 25 10 ... 31

The 2:1 interleave allows the 753 two sector times to transfer a sector to memory. For example, if you are transferring Sectors 0 and 1, the following occurs in each case:

In 1:1 interleaving, when the first sector finishes reading from the disk, the next sector is almost under the head and ready for reading; the 753 misses a revolution if there is not enough room in the buffer.

In 2:1 interleaving, when the first sector finishes reading from the disk, the next sector is still a full sector time away, thereby giving the 753 twice the time to empty the buffer to memory. The scheme above shows the extra sector time as Sector 16.

Interleaving schemes from 2:1 to 16:1 are software programmable. Since the 753 determines the drive's sector location by comparing headers, you can use the Write Track Headers command to customize the interleaving scheme to your application. For example, if your system transfers data in 2K-blocks, then the most effective interleaving scheme may be:

0 1 2 3 16 17 18 19 4 5 6 7 20 21 22 23 8 9 10 11 ...

8.3 Handling Media Defects

There are three methods for handling media defects: 1) slipping a sector; 2) remapping a sector to a new sector on the last head of that cylinder; and 3) remapping the entire track to a different track on the disk.

8.3.1 Slipping a Sector

Slipping a sector requires using the Read and Write Track Headers commands to mark the bad sector and slip the rest of the sectors into the next position on the disk. Figure 8-4 shows an 8-sector track before and after slipping Sector 3.

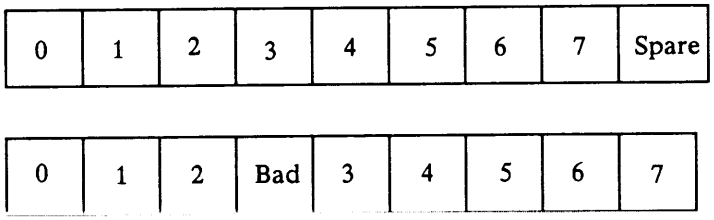


Figure 8-4. Sector Slip

8.3.1.1 Sector Slip Procedure

1. Determine the bad sector by writing and reading the track with several patterns and/or using the manufacturer's defect map information.
2. Read the track headers into a buffer in host memory.
3. Compare each header with the bad sector's header.
4. After locating the bad sector, mark it by writing EEH into each of the four header bytes (see Figure 8-6).
5. Test the last sector to determine if it is a spare. If it is a spare, continue; if not, you must find an alternate sparing method.
6. Move each sector header into the next location, *slipping* the sectors down the track (see Figure 8-4).
7. Write the track headers back to the disk.

8.3.1.1 Sector Slip Procedure (continued)

The following figures depict 753 headers:

Sync	Cylinder Low	Cylinder High	Head	Sector

Figure 8–5. Normal Header

Sync	EE	EE	EE	EE

Figure 8–6. Header Marked Bad

Sync	DD	DD	DD	DD

Figure 8–7. Header Marked Spare

00	00	00	00	00

*Figure 8–8. Runt Header **

* This header is invisible in normal operation. It is here for informational purposes only.

Sync	New Cyl Low	CC	New Cyl High	Head

Figure 8–9. Track Remap Header

8.3.1.2 Advantages of Sector Slipping

A full track of information is still transferred in one revolution of the disk. Other methods of sector slipping require two or more revolutions to transfer one track of information.

8.3.1.3 Disadvantages of Sector Slipping

Having one or more spares on each track uses disk space inefficiently.

8.3.2 Cylinder Sparing

Cylinder sparing is like sector slipping, except the spares are on the cylinder's max head.

8.3.2.1 Cylinder Sparing Procedure

1. Determine the defective sector.
2. Read the track headers and mark the defective sector bad (see Figure 8–6).
3. Write the track headers back to the disk.
4. Read the track headers on the maximum track of that cylinder.
5. Find a spare sector; it contains four bytes of 0DDH (see Figure 8–7).
6. Put the bad sector's header into this sector. The header contains the header and sector values for the sector being remapped on the original track.
7. Write the track headers back to the drive.

8.3.2.2 Advantages of Cylinder Sparing

Cylinder sparing uses less disk space for remapping bad sectors. You may decide to allow only ten spares for a drive with twenty heads. Sector slipping is less efficient as it requires a minimum of twenty sectors per cylinder for this drive (one per head).

8.3.2.3 Disadvantages of Cylinder Sparing

Cylinder sparing is slower than sector slipping. The 753 looks for the requested sector for one revolution plus one sector on the original track; then it switches to max head and looks for another revolution plus one sector. This method takes up to three revolutions to transfer one track of information, assuming only one bad sector.

8.3.3 Tracking Remapping

The 753 remaps an entire track to another location on the same disk drive by writing the defective track's headers with a code and new disk address.

8.3.3.1 Track Remapping Procedure

1. Read and save the defective track's headers.
2. Allocate space for a write track headers buffer. Each header has 0CCH in the second byte. The first, third, and fourth bytes contain the new head, cylinder high, and cylinder low. Write this information to the defective track with a Write Track Headers command (see Figure 8–9).
3. Write the track headers that were read in Step 1 to the destination track with a Write Track Headers command.

8.3.3.2 Advantages of Track Remapping

Track remapping is useful in sparing tracks with very large or multiple defects.

8.3.3.3 Disadvantages of Track Remapping

Track remapping is relatively slow as the 753 must seek to the first track and then determine if it has been remapped. Then the 753 seeks to the spare track area (usually at the end of the disk). This sequence can add up to four revolutions to the 753 overhead.

8.3.4 Recommended Remapping Procedure

Xylogics recommends using all three methods of defect mapping. Allowing one spare sector per track takes care of 95% of the media defects. An additional 0.2% of the sectors on the cylinder to be spared on the last head provide up to 99% remapping. Allowing three or four tracks for remapping should provide a defect-free media. Allocating this amount of disk space for defect handling totals 2% of the media. Having two spares per track on the same drive (a Fujitsu Eagle with 512-byte sectors) uses 4% of the media.

8.4 Multiple I/O Requests and Chaining

The 753 has two ways of speeding up multiple IOPB processing. One method allows the driver to add IOPBs to the 753's queue by the same procedure as starting the first IOPB. (The 753 command queue stores up to 10 full IOPBs.) The second method allows the driver to chain IOPBs together, and then give the 753 a command-chain.

8.4.1 Multiple I/O Requests

The following procedure allows you to add IOPBs to the 753 queue:

1. AIOP must be clear. If it is not clear, wait; it normally clears within 100 microseconds (see Sections 3.3.2 and 4.2.1).
2. Point the five IOPB address registers to the beginning of the IOPB (chain).
3. Write AIO.

8.4.2 Chaining

Each IOPB has a *Chain Enable* (CHEN) bit and a Next IOPB pointer. IOPBs can be chained together by setting CHEN and pointing the Next IOPB pointer to the next IOPB ready for execution. Each IOPB in the chain points to the next. Clearing CHEN in the last IOPB stops the chain.

> *The next IOPB address is the physical, not the virtual, address.*

8.4.3 753 Operation

The 753 treats IOPBs the same, regardless of how they were added to the queue. Overlap seeking functions only when enabled and the 753 is working with a queue or chain of IOPBs.

8.5 Error Recovery

The 753 may automatically retry errored operations. Two options are available: one involves retrying *drive fault* and *seek* errors; the other involves the retry algorithms for ECC recoveries. The Read and Write Controller Parameters commands enable these options.

8.5.1 Automatic Operation Retry

The 753 automatically retries an operation if the reason for the initial failure is a *drive fault* or *seek* error. Setting ASR with a Write Controller Parameters command enables this option.

8.5.2 ECC Error Recovery

The ECC algorithms have a much better chance of recovery if the 753 retries the operation before using the ECC. The 753 retries the Read operation once before applying the ECC if RBC is set. Set RBC with a Write Controller Parameters command.

There are three options for applying the ECC. Using Mode 0, the 753 provides the driver with the pattern and offset for correcting the actual error in memory. Mode 1 flags the error, but does not stop the transfer and calculate any correction information. Mode 2 calculates the correction information and applies it to the data in host memory.

8.5.3 Using the Error Recovery Options

Changing the appropriate controller parameter enables or disables the error recovery options. Once set, the 753 applies the retries as requested. The 753 posts a completion code indicating a recovery operation occurred. The driver applies the ECC in ECC Mode 0. If a retry fails, the completion code reflects the fatal error.

8.6 Maintenance Mode

Firmware supports a non-IOPB driven maintenance mode. It allows you to perform basic testing within the 753 by setting control bits in the CSR and entering the desired test number and data through the address registers. This firmware also provides a window through which internal registers may be examined or modified.

8.6.1 Register Use in Maintenance Mode

The function code in the Test Number Register determines whether or not the 753 uses the Input Data Byte and Output Data Byte Registers (see Table 8-1). Familiarize yourself with the Control and Status Register before reading this section (see Section 3.3).

Table 8-1. Register Use in Maintenance Mode

Register	Description
1	Test Number or Function Code
3	Input Address Low
5	Input Address High
7	Input Data Byte (If Required)
9	Output Data Byte (If Required)
B	Control and Status Register
D	Fatal Error Register

8.6.2 Maintenance Mode Protocol

8.6.2.1 Executing a Maintenance Command or Entering the Maintenance Mode

First, setting the *Maintenance Mode* (MM) and AIO bits forces entry into the maintenance kernel. The kernel initializes the CSR and poll mask and sets RIO; then clear RIO. The kernel expects the Input Address Low Register to contain a maintenance test number or function code for execution.

8.6.2.1 Executing a Maintenance Command or Entering the Maintenance Mode (continued)

Configure *Busy* and AIO for polling. Setting *Busy* and AIO selects the register image test; clearing *Busy* returns control to the maintenance kernel. Setting AIO causes the maintenance firmware to read and decode the command string from the Input Address Registers. After successfully decoding the command string, the firmware echoes it (command, address, and data) to the Output Address Registers and clears AIO. The 753 acknowledges receiving AIO and executes the requested command. After completing the command, the 753 updates the Output Address Registers with test-pertinent data and sets RIO. Setting RIO indicates the end of firmware involvement; the contents of the Output Address Registers are valid. The AIO/RIO protocol is identical to normal mode.

Since each test and its expected results are different in nature, the Output Address Registers hold the test result information (address, data, etc.). Firmware sets RIO upon command completion; it sets FERR if a failure occurs or if host software issues an illegal command.

8.6.2.2 Exiting the Maintenance Mode

To exit the maintenance mode, clear MM and RIO, and set AIO. This returns control to the normal mode kernel. The 753 acknowledges by setting RIO.

8.6.2.3 Diagnostic Considerations

The Input/Output Address Register Verify is the first test the diagnostic should execute. Firmware flags the power-up test failures by setting FERR. MM remains set. Firmware saves the self test error numbers internally until it verifies the Input and Output Registers.

8.6.2.4 Register Test

You must enter the maintenance mode to invoke the register test. After firmware acknowledges the request, you should set *Busy*. *Busy* remains set during this test.

> *You must enter the maintenance mode separately because the normal mode firmware does not allow setting Busy (defined as RMM in maintenance mode).*

8.6.2.4 Register Test (continued)

Writing the Input Address Registers, and then AIO, signals firmware to copy the data to the Output Address Registers. Firmware sets RIO after completing the copy; then host software clears RIO. Clearing *Busy* exits this test and returns the 753 to maintenance mode.

8.6.2.5 Test Variables

Some of the internal tests require the address and data to perform their particular function. On-board memory has space allocated for this data. These locations are loaded with default values for initial use. However, you may alter these variables through the manual mode.

8.7 Multiprocessor Support

Several 753 options make multiprocessor environments easier to support: the programmable interrupt vector, interrupt level, register address modifiers, and register busy semaphore.

8.7.1 Interrupts

Each IOPB specifies the interrupt level and vector for that command. In a multiprocessor environment, each processor can have its own interrupt level and vector.

8.7.2 Register Busy Semaphore

Hardware supports the RBS bit. RBS allows multiple processors to share the registers without colliding. The register access protocol involves reading the CSR. If RBS is clear, the host has control of the register, and retains control until it clears RBS in the Control Register. If the first read to the Status Register indicates that RBS is set, then another host has control of the register and must wait until RBS clears.

The 753 sets RBS immediately after a host reads the CSR. If a host attempts a read, and RBS is clear, then the 753 sets RBS; any successive reads by other hosts will *see* that RBS is set. When the host using the registers finishes, it must clear RBS. Clearing RBS and setting AIO can occur in the same register write. Clearing RBS without having control of the registers violates the register protocol.

8.7.3 Address Modifiers

You can use the address modifiers to assign each processor separate address space.

8.8 Command Optimization

Enable Command Optimization (COP) with a Write Controller Parameters command. Command optimization is the reordering of IOPBs in the 753's command queue. The reordering causes the 753 to enable elevator seeks and process several IOPBs within one revolution. The 753 starts the first IOPB it receives, and then reorders IOPBs as they are DMAed into the queue. COP is most effective when there are more than two IOPBs in the 753 queue.

The 753 places the IOPB in a position relative to the other IOPBs in its internal queue. Then it tests the IOPBs and determines if any are contiguous on the disk. The 753 links any contiguous IOPBs it finds, and executes them as one disk operation; it does not link the DMA portion of the transfer.

The 753 first links the IOPBs with respect to an ascending cylinder order. If the 753 receives an IOPB with a cylinder number lower than the current cylinder, it positions the IOPB at the end of the queue in descending cylinder order. Conversely, if the 753 is operating in descending cylinder order, it positions an IOPB with a cylinder number higher than the current cylinder at the end of the queue in ascending cylinder order.

8.9 Software Control

The 753 has many parameters that can be modified by software control. The parameters can be set in bulk with three write parameters commands: Write Format Parameters, Write Drive Parameters, and Write Controller Parameter.

8.9.1 Modifying a Single Parameter

The best method for modifying a single parameter involves executing a read parameters command for the associated parameter block, modifying the single parameter, and then writing the parameter block back to the 753 (assuming the controller was previously set up).

8.9.2 Modifying a Group of Parameters

Use the same method as in Section 8.9.1, or set all the parameters in the specific IOPB and execute the appropriate write parameters command. For example, Fields 1 through 7 and the interleave factor must be set to their appropriate values before issuing the Write Format Parameters command. The 753 sets all parameters to the new values contained in the IOPB.

8.9.3 Parameter Reference Point

When the 753 is all set, read the parameters and save the information for future use.

8.10 Scatter/Gather

Scatter/gather is used with standard Read and Write commands. In a scatter read, the 753 transfers the data to up to 32 blocks of memory. A gather write gathers data from up to 32 blocks of memory and writes it to the disk. Each memory block must have an even byte count and have less than 64K bytes. The blocks may be scattered throughout memory.

8.10.1 Scatter/Gather Link List

You can determine the linked list length by multiplying the number of elements in the list by 8 (each element is 8-bytes long). All data addresses must have word boundaries and an even byte count. For Read and Write operations, enter the number of elements in the linked list into Byte 6, bits 3 through 7. A zero in this field indicates the linked list has 32 elements.

Table 8-2. Scatter/Gather Link List

Link Number	Byte	Description
1	00-01	Byte Count (Multiples of 2)
	02	Reserved
	03	Data Address Modifier
	04-07	Data Address (Word Boundaries Only)
2	08-09	Byte Count
	:	
	:	
n	xx	

8.10.1 Scatter/Gather Link List (continued)

Table 8-3. Link List Field Values

Link Field Value	Decimal Equivalent
0	32
1	1
2	2
:	:
9	9
A	10
B	11
:	:
1E	30
1F	31

8.10.2 Setting Up a Scatter/Gather Transfer

The IOPB Data Address and Modifier bytes should now point to the start of the linked list. The linked list length field should equal the total number of elements on the list.

Elements of memory descriptors comprise the linked list. Each element describes the starting address and the length, in bytes, of the memory block. The sum of the byte count of all the elements in the linked list must equal the sector count times the sector size in bytes.

The IOPB and linked list in Figure 8-10 illustrate a read transfer to 6 blocks of memory. The sector size is 512-bytes per sector; we are transferring 3 sectors of information. The 753 transfers the first 16 data bytes from each sector to a separate data buffer. It scatters the bulk of the data, 512-bytes per sector, into memory as 3 blocks having 512 bytes each.

Set SGM and execute the IOPB in Figure 8-10.

8.10.3 753 Operation

The 753 proceeds as if doing a normal read until it starts the data transfer into memory. The contents of the linked list now controls the DMA processor; it gives the processor the byte count and address for each element on the list. The processor takes the data out of the FIFO and transfers it to memory as described in each element on the list.

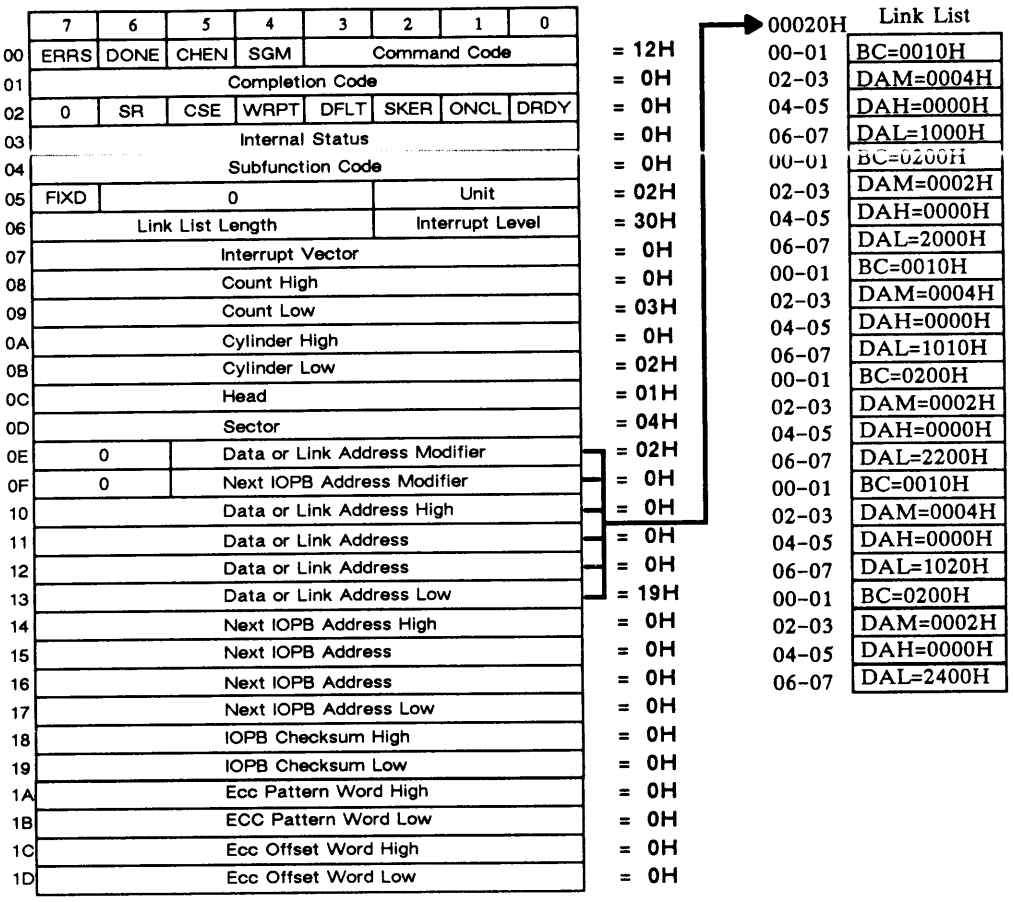


Figure 8-10. Scatter/Gather Transfers

8.11 IOPB DMA Throttle/Throttle Dead Time

The 753 always transfers IOPBs in word mode; it uses the last specified values for the throttle and throttle dead time.

Host software can set the throttle dead time (TDT) field in the controller parameters IOPB. This value defines the time that the 753 waits before attempting to regain control of the bus between throttle bursts. There are four valid TDT values.

Table 8-4. Throttle Dead Time Values

TDT Value	Time
0	0 microseconds
1	3.2 microseconds
2	6.4 microseconds
3	12.8 microseconds

8.12 IOPB Checksum

While debugging the driver, you may choose to append the checksum to the IOPB. The checksum is the sum of IOPB Bytes 0 through 17, and is expressed as a 16-bit quantity. The 753 generates a checksum with the data from the IOPB and compares it to the appended checksum; a miscompare causes a fatal error. If AUD and ICS are set, the 753 appends a new checksum as it updates the IOPB. Having a valid checksum in the Write Controller Parameters IOPB disables the checksum.

8.13 Fixed/Removable Media

The 753 accesses any physical drive with fixed and removable media, like the CDC, CMD, or LMD, as one logical unit. FIXD is clear for the removable media of Unit 0 and set for the fixed media of Unit 0.

8.13.1 Head Offset

The head offset refers to the bit(s) that must be set during a drive head select sequence to select between the fixed and removable portions of the drive. Host software must specify a head offset value for fixed/removable drives: one for the removable portion of the drive, and one for the fixed portion. The offset value is a hexadecimal number that the 753 adds to the head number. Two fixed/removable drives are the CDC Lark and CMD. The head offset value for the removable portion of a CMD is zero; the head offset value for the fixed portion is 10H. See the appropriate vendor manual for specific head offset values.

8.14 Read/Write Header, Header Verify, Data, and Data ECC

This maintenance command tests the controller and software driver. It simulates ECC errors to verify the ECC is working. The operation includes reading a sector with its header into memory, modifying the data, and then writing the sector back to the disk. The 753 does not recalculate the ECC.

The data read back is either 12 or 14 bytes larger than the data sector size (depending on EC32's status). For example, given a 512-byte sector size, the 753 returns 524 bytes if EC32 is set, and 526 bytes if EC32 is clear. Table 8–5 shows the relationship between EC32's status and the returned data.

Table 8–5. EC32 vs. Returned Data

Byte	EC32 Clear	Byte	EC32 Set
0-3	Header*	0-3	Header*
4-7	Redundant Header*	4-7	Header ECC
8-(n+8)	Data 8-(n+8)	Data	
(n+8)-(n+14)	48-bit ECC	(n+8)-(n+12)	32-bit ECC

* Figure 8–5 through Figure 8–9 describe the header information

8.14.1 Simulating an ECC Error

Simulating an ECC error involves reading a sector with a Read Header, Header Verify, Data, and Data ECC command, changing a data byte or bit, and writing the sector back with a Write Header, Header Verify, Data, and Data ECC command. Reading this sector with a normal Read command should return an ECC error.

There are two common problems associated with simulating an ECC error. First, the corrected data byte may be next to the one in error. Second, the data may not be serially written to the disk as it appears on your terminal screen. Thus, a 2-bit error crossing a byte boundary may not be correctable.

The 753 usually accesses memory in word or longword mode, but corrects data in ECC Mode 2 via byte transfers. Since some bus adapters reverse the byte addressing scheme within a word, the 753 corrects the wrong data. The only solution for this situation is to either correct the adapter or use ECC Mode 0.

The serial data is placed on the disk with bit 0 of each byte first. Table 8-6 shows a simulated 2-bit error crossing byte boundaries. Since the 2 bits in error are really 15 bits apart, they may be uncorrectable. This situation can only occur when testing because a 2-bit adjacent error refers to 2 adjacent bits on the media.

Table 8-6. Simulated 2-Bit Error Crossing Byte Boundaries

Before Simulated Error				
Memory Data:	45	67		
Serial Disk Data:	5 1010	4 0010	7 1110	6 0110

Simulated Error				
Memory Data:	44	E7		
Serial Disk Data:	4 0010	4 0010	7 1110	E 0111

8.15 Release on Request

When ROR is set, the 753 tests the VMEbus between each throttle for other pending bus requests. If another request is pending, the 753 releases the bus. If there are no bus requests, the 753 remains bus master. The throttle value determines how often the 753 tests the bus. With higher throttle values, the 753 tests the bus less frequently. Lower throttle values slow down the DMA.

Section 9:
Theory of Operation

Section 9: Theory of Operation

9.0 Introduction

This section is an overview of how the controller works. It explains the functional blocks of the hardware and microcode, and how the code affects the controller operation.

9.1 The Hardware

The 753 connects up to four SMD-E disk drives to VMEbus systems using the following logic blocks:

VMEbus Interface
Register Read, Write, and Interrupt (REGCEL)
Microcontroller
Direct Memory Access Controller (VMEDMA)
Disk Data Buffer (FIFO)
Disk Front End (DSKCEL)
SMD-E Interface

9.1.1 VMEbus Interface

This block contains interface logic for the signals on the VMEbus. The 753 is a VME slave for programming purposes, i.e., the register file. The 753 is also a slave when it responds to an interrupt acknowledge with the interrupt vector. The REGCEL performs both of these functions. Both the REGCEL (receive mode) and VMEDMA (driving mode) use the address modifier transceiver. The 753 is a VME master for DMA purposes; it uses DMA to read and update IOPBs and also to read and write disk data from host memory. The VMEDMA chip performs this function. The VME data bus is 8-, 16-, or 32-bits wide; the VME address bus is 32-bits wide.

The 753 uses only 16 address bits for decoding its register addresses (it only compares 12 bits); it drives all 32 address bits as a master. The VMEDMA updates the lower 16 address bits; the microcontroller updates the upper 16 bits.

9.1.2 Register Read, Write, and Interrupt

The REGCEL provides the program interface. The 753 uses registers to point to an IOPB ready for execution, to point to a completed IOPB, and to perform various control functions.

The VMEbus accesses the registers via a bus that is shared between the REGCEL and VMEDMA. There is no contention because the VMEDMA cannot acquire the bus while a slave (register) access is in progress. The REGCEL answers a register access by the VMEbus with the signal *DTACK*. The upper address bits are decoded and the proper address modifier is required for a register access.

The microcontroller (micro) accesses the registers via the internal data bus. It programs the REGCEL to interrupt if certain bits are set in the CSR or if a timer overflows. Other conditions are programmed to assert another interrupt to the micro.

The REGCEL also supports the VMEbus interrupt protocol. The micro programs the REGCEL to start an interrupt sequence by asserting the request line. When the system responds with *IACK* and *DS0*, the REGCEL drives the interrupt vector onto the bus and asserts *DTACK*. This process causes the system to execute an interrupt service routine (ISR). The ISR should clear RIO in the CSR. In this way, the 753 passes the completed IOPB back to the system.

The REGCEL has a register for storing the address modifier that the VMEDMA enables onto the bus. Like the VME address, the address modifier is pipelined. A new modifier can be loaded into the REGCEL while a different modifier is being driven onto the VMEbus.

The REGCEL also has two timers. Timer 0 is a watchdog timer: it expires and interrupts the micro if firmware does not reset it periodically. Timer 1 is a counter for header errors. The DSKCEL asserts a pulse when a disk header does not match the expected header. A *header not found* error occurs if Timer 1 overflows before the controller finds the desired header.

9.1.3 The Microcontroller

The 753 uses a 16-MHz 8031 microcontroller. It fetches instructions by asserting an address on Port 0, latching the address with *ALE-L*, and reading the data from the EPROM; it reads data into Port 0. Many instructions cause the micro to access an external byte using strobe decoders and a transceiver. Many of these external bytes are in the Xylogics LSI chips REGCEL, VMEDMA, and DSKCEL.

9.1.3 The Microcontroller (continued)

The 753 accesses external bytes through Port 0. It uses Port 1 for DSKCEL related outputs, Port 2 for the upper byte of the EPROM address, and Port 3 for miscellaneous control signals. Inputs can all be considered micro interrupts, although most are actually polled. Two inputs are from the REGCEL, one from the VMEDMA, and two from the DSKCEL.

This block also includes the internal RAM (IRAM) logic. The micro really uses the IRAM as a scratch pad RAM. It stores the controller, drive, and format parameters there as well as IOPBs. The IRAM is single ported, but is used by both the VMEDMA and the micro. Since the microcontroller starts the VMEDMA, it knows when the IRAM is off limits for micro access. The 753 always DMAs IOPBs in word mode.

9.1.4 Direct Memory Access Controller

The VMEDMA controls the transfer of data between the disk buffer or IRAM, and the VMEbus. The micro programs the VMEDMA to transfer a sector's worth of data, or less, to or from a specified area in system memory, ensuring proper handling of odd starting addresses. Thus, the DSKCEL can execute an even or odd address transfer on a per sector basis. Before the VMEDMA can transfer data, it must acquire the bus by sending out *BUSREQ*, which must be jumpered to one of the Bus Request lines. The system arbiter sends back *BGIN* via another jumper. The VMEDMA then asserts *Busy* on the VMEbus, and the 753 controls the VME address, data, and control lines.

A transfer involves asserting a valid address, *DS0* and/or *DS1* and valid write data or read data, then waiting for *DTACK*, and proper buffer control. The order of the buffer request and data strobe is reversed, depending on the direction of the transfer.

The DMA circuitry pipelines data to increase performance. The pipeline allows one word of data to be transferred on the bus while another is transferred to the buffer. Thus, the access times of the buffer and the bus can overlap, except for the first and last burst transfers. A prefetch primes the pipeline (which is emptied at the end of the burst).

The disk buffer is word wide and uses a longword wide pipeline. The interface logic turns a VMEDMA longword request into two buffer (word) requests; it turns a VMEDMA word request into two IRAM requests. The IRAM is byte wide and uses a word wide pipeline.

9.1.5 Disk Data Buffer

The 128K-byte buffer is organized word wide. The DSKCEL will not start a write unless the buffer contains a sector's worth of data; it will not start a read unless the buffer can hold a sector's worth of data. The micro tracks buffer use and starts or stops the VMEDMA and DSKCEL as necessary.

9.1.6 Disk Front End

The DSKCEL is a downloadable disk sequencer. The micro loads the disk read, write, and format programs into the DSKCEL on power-up, and modifies the program when new format parameters are loaded or an alternate command (such as Read Header, Data, ECC) is received. The DSKCEL issues some SMD control signals, such as *Read Gate* and *Write Gate*, while the micro issues others, such as *Cylinder Tag* and *Unit Tag*. Generally, if timing is critical, the DSKCEL issues the signal since it runs off the disk bit clock. The DSKCEL has serial registers for FIFO data, the header, and ECC. It performs sync bit search, header check, and ECC check and provides status bits to the micro. The DSKCEL interrupts the micro when done. DSKCEL *Done* may mean *header found, end of sector*, or *bad spot found*. Generally, the DSKCEL runs on a sector-by-sector basis, with the micro controlling how many sectors are transferred. The micro allows the DSKCEL to run. The program starts running when a sector or index pulse comes in from the disk. The micro informs the DSKCEL when the next sector involves an odd address DMA and when the current sector is the last. The DSKCEL also performs other functions, such as ECC, runt sector detection, and midtransfer head tags.

9.1.7 SMD-E Interface

This block contains interface logic for SMD connectors. The micro controls the tags for unit select, cylinder, head, and control (read/write). Both the micro and disk sequencer receive and use various disk status lines, and the sector and index pulses. The micro controls whether the pulses to the DSKCEL include sector pulses or just the index pulse.

The SMD interface chips require -5 volts from an on-board regulator to operate properly. A drive configured as any unit number, from 0 through 7, can connect to any port.

9.1.7 SMD-E Interface (continued)

Typically, during ECC, when no disk unit is selected, servo clock is a free-running 10-MHz clock. The clock synchronizer ensures that no clock slivers get into the DSKCEL when Read Gate is switched. The disk sequencer clock is servo clock (interface clock) when writing the disk and read clock when reading the disk. The write clock is simply delayed servo clock.

9.1.8 Power-up

During power-up, the *Open Cable Detect* signal is asserted on the drive interface. This signal disables any erroneous writes. The bus signal *SYSRESET* sets the error LED (L2) and asserts *SYSFAIL* on the bus. The micro runs its diagnostics and then clears L2 and *SYSFAIL*.

9.1.9 Power-down

During power-down, the 753 responds to *ACFAIL* by turning off the DSKCEL, allowing any writes to finish the current sector, and then asserting *Open Cable Detect* to the drive. This sequence generates a fatal error which cannot be reset until *ACFAIL* is deasserted.

9.1.10 System Reset

When the 753 detects *SYSRESET*, it resets its internal micro and the three custom integrated circuits. This sequence immediately terminates any writes to the disk (possibly leaving it with an unreadable sector). The 753 then executes the power-up diagnostics.

9.2 The Microcode

9.2.1 The Kernel

Figure 9–1 illustrates the kernel. It is entered after the power-up test and initialization. It has four major functions. The following subsections describe three of these functions; the fourth function, scheduling DMA, would needlessly complicate this discussion.

9.2.2 Is AIO Set?

Each time around the kernel, the 753 checks AIO's status. If the host sets AIO, the 753 reads the IOPB and places it in its internal command queue. If the queue is full, the 753 saves the IOPB address so that it can later read it into the queue. The queue holds 10 IOPBs; the 753 saves the next 31 IOPB (chain) addresses. This function is really tested several times in the kernel, but for simplicity Figure 9-1 shows it as a single function.

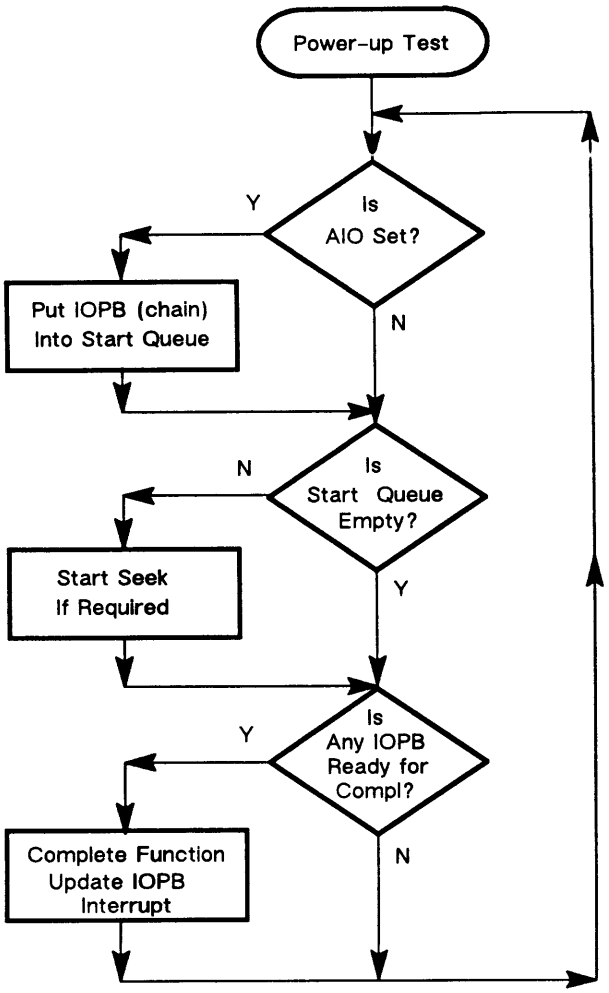


Figure 9-1. The Microcode Kernel

9.2.3 Is Start Queue Empty?

The IOPBs in the internal queue are divided into three groups: *seek not started, seek started, and seek complete or not required.* When first entering the queue, the IOPBs belong to *seek not started.* This step examines the IOPBs to determine if a seek is required. If a seek is required, and the drive is not busy, the 753 issues the Seek command to the drive and assigns the IOPB to *seek started.* If a seek is not required, the 753 marks the IOPB as having its seek complete and assigns it to *seek done or not required.*

9.2.4 Is Any IOPB Ready for Completion?

The 753 checks for an IOPB in the third group, *seek complete or not required.* It does this by first checking if an IOPB has *seek done* set; if not, it selects each drive in the second group, seek started, to determine if a seek is complete. The 753 executes the first *seek done* IOPB. When the 753 completes the function, it updates the IOPB and issues an interrupt.

9.2.5 Queuing IOPBs for Execution

The 753 command queue holds 10 IOPBs. The first AIOs have their respective IOPB (chains) read directly into the queue until it is full. If COP is set, the 753 reorders the IOPBs inside the queue and executes them in proper order.

As the 753 completes an IOPB, it frees a slot in the internal queue and reads in a new IOPB. If COP is set, the 753 inserts the new IOPB at the proper point in the reordered IOPBs.

9.3 Performing a Function

The 753 performs each function differently. If a function requires a seek, the 753 issues the seek and waits for it to complete before performing the function. The following subsections group similar functions together and explains their differences.

9.3.1 No Operation (NOP)

The NOP command verifies the 753 is operational: it reads the IOPB from host memory, sets *Done*, and posts a completion code.

9.3.2 Normal Reads and Writes

Normal reads and writes are very similar regarding 753 processing. The main difference between reads and writes is which way, and when, the data moves. On writes, the 753 DMAs the data and begins the disk transfer when the FIFO holds one sector's worth of data. On reads, the 753 begins the disk transfer immediately and DMAs the data as soon as the first word is available from the FIFO (see Section 9.3.3).

When enabled, the disk sequencer compares every header that arrives under the heads with the target sector. When a header compare succeeds, the 753 also tests the header verify. If both tests succeed, the transfer occurs on that sector. To continue the transfer, the 753 loads the next target header so it can compare the next sector that arrives under the heads. The 753 does not wait for index before comparing headers.

If software specifies more than one sector, the 753 increments the disk address on successive sectors. First, the controller increments the sector number until it reaches the maximum sector address. Then, it clears the sector address and increments the head address. When the 753 reaches the maximum head and sector addresses, it clears them and increments the cylinder address. When the 753 reaches the maximum sector, head, and cylinder addresses, the next sector causes an *illegal cylinder address* error. The 753 completes the IOPB as soon as it completes the DMA and disk transfers, or a hard error occurs.

9.3.3 Read Ahead

The 753, using a 128K-byte FIFO, reads data into the buffer before the UNIX device driver actually requests it. After completing any normal Read command, the 753 continues reading disk data into its buffer until the buffer fills or software requires a cylinder seek. If the next request to the 753 is a read for the next logically contiguous sector of data, it transfers the data from its buffer to host memory and completes the I/O request. The 753 satisfies subsequent reads from its buffer until it exhausts all read ahead data. It completes the next Read command with a standard disk read, and refills the read ahead buffer.

The 753 refills any buffer space freed by a Read command while the read ahead is in process before terminating read ahead. Thus, the 753 can transfer an entire cylinder's worth of data at disk speed even though the UNIX I/O requests may not be sent at disk speed. For any request other than a logically contiguous read, the 753 flushes the buffer and executes the specific command.

Unit/Command Look Ahead (UCLA) enhances read ahead performance by reordering read requests when they have been interleaved with write requests or read requests from a different disk or cylinder address.

9.3.3 Read Ahead (continued)

The 753 automatically enables UCLA when it enables read ahead. Since the 753 terminates read ahead when it receives a noncontiguous read, UCLA allows a higher percentage of contiguous read requests, better use of read ahead, and less frequent buffer flushes.

9.3.4 Dyna-throttle

During a Read command, each time firmware executes the DMA scheduler, the 753 calculates the amount of data currently in the buffer, and starts a DMA from one to six sectors to host memory. This method proves extremely efficient when transferring data previously brought in from the disk via read ahead; it also enables the 753 to catch up from heavy bus load periods when the DMA falls behind. Dyna-throttle does not override the normal throttle and throttle dead time features that tune system bus activity.

9.3.5 Seeks

Depending on the subfunction, the 753 may select the drive and read the first header that arrives under the heads. If it is a spare or bad header, the 753 reads the next header until it finds a good one. The 753 returns the data read in the header in the IOPB.

9.3.6 Drive Reset

The 753 issues the drive a fault clear and then a recalibrate (return to zero). The 753 waits for the drive to finish recalibrating before completing the IOPB. Since the *wait for recalibrate done* is a background task, the controller continues processing other IOPBs, except on the drive recalibrating.

9.3.7 Write and Read Parameters

Section 7 describes these functions in detail.

9.3.8 Extended Read and Write Commands

This section is similar to Section 9.3.2. The following subsections detail their differences.

9.3.8.1 Track Headers Commands

The disk sequencer waits for index before determining where the transfer begins. Track headers commands always start at index. The number of sector headers returned equals the physical sector count read with a Read Drive Parameters command.

9.3.8.2 Header, Header Verify, Data, and Data ECC

The disk sequencer waits for index before determining where the transfer begins. Header, Header Verify, Data, and Data ECC commands use the sector address as an offset count from index when determining where to start. The 753 increments the sector address during a multisector transfer, but does not clear it if it reaches maximum sector. The 753 inhibits the *illegal sector address* error, and does not increment the head and cylinder addresses.

9.3.8.3 Defect Map Extended

The disk sequencer waits for index, syncs up to the manufacturer's defect map, and reads it into memory. The 753 only executes one track per IOPB.

9.3.8.4 Read Verify

The disk sequencer executes a Read command, and the DMA sequencer performs as if it is doing a disk write. The 753 compares the serial data from the FIFO/SERDES with the data from the disk. A miscompare causes a *verify* failure. The 753 returns the failing disk address in the IOPB.

9.3.9 Diagnostics

The 753 executes the power-up self test.

9.4 Completing a Function

The 753 completes the transfer after completing both the DMA and disk transfers. The 753 updates the IOPB in host memory, interrupts, and clears *Busy* if all IOPBs in its queue are complete. If an error occurs, the 753 completes the faulty IOPB and continues processing the other IOPBs. If a fatal error occurs, the 753 finishes the IOPB(s) in process, posts the appropriate fatal error code, and sets FERR. The host must execute a *Controller Reset* before sending any IOPBs to the 753.

Section 10:
Interface Signals

Section 10: Interface Signals

10.0 Introduction

This section provides useful interface information for your 753 disk controller.

10.1 VMEbus Interface Signals

Mnemonic	Conn.	Pin	Used By 753	Description
A01	P1A	30	Y	
A02	P1A	29	Y	
A03	P1A	28	Y	
A04	P1A	27	Y	
A05	P1A	26	Y	
A06	P1A	25	Y	
A07	P1A	24	Y	
A08	P1C	30	Y	
A09	P1C	29	Y	
A10	P1C	28	Y	
A11	P1C	27	Y	
A12	P1C	26	Y	
A13	P1C	25	Y	
A14	P1C	24	Y	
A15	P1C	23	Y	Address Bus
A16	P1C	22	Y	
A17	P1C	21	Y	
A18	P1C	20	Y	
A19	P1C	19	Y	
A20	P1C	18	Y	
A21	P1C	17	Y	
A22	P1C	16	Y	
A23	P1C	15	Y	
A24	P2B	4	Y	
A25	P2B	5	Y	
A26	P2B	6	Y	
A27	P2B	7	Y	

10.1 VMEbus Interface Signals (continued)

Mnemonic	Conn.	Pin	Used By 753	Description
A28	P2B	8	Y	Address Bus
A29	P2B	9	Y	
A30	P2B	10	Y	
A31	P2B	11	Y	
AM0	P1B	16	Y	
AM1	P1B	17	Y	
AM2	P1B	18	Y	Address Modifier
AM3	P1B	19	Y	
AM4	P1A	23	Y	
AM5	P1C	14	Y	
D00	P1A	1	Y	
D01	P1A	2	Y	
D02	P1A	3	Y	
D03	P1A	4	Y	
D04	P1A	5	Y	
D05	P1A	6	Y	
D06	P1A	7	Y	
D07	P1A	8	Y	
D08	P1C	1	Y	
D09	P1C	2	Y	
D10	P1C	3	Y	
D11	P1C	4	Y	
D12	P1C	5	Y	
D13	P1C	6	Y	
D14	P1C	7	Y	
D15	P1C	8	Y	Data Bus
D16	P2B	14	Y	
D17	P2B	15	Y	
D18	P2B	16	Y	
D19	P2B	17	Y	
D20	P2B	18	Y	
D21	P2B	19	Y	
D22	P2B	20	Y	
D23	P2B	21	Y	
D24	P2B	23	Y	
D25	P2B	24	Y	
D26	P2B	25	Y	

10.1 VMEbus Interface Signals (continued)

Mnemonic	Conn.	Pin	Used By 753	Description
D27	P2B	26	Y	Data Bus
D28	P2B	27	Y	
D29	P2B	28	Y	
D30	P2B	29	Y	
D31	P2B	30	Y	
Strobes				
AS*	P1A	18	Y	Address Strobe
DS0*	P1A	13	Y	Data Strobe Zero
DS1*	P1A	12	Y	Data Strobe One
DTACK*	P1A	16	Y	Data Transfer Ack.
Clocks				
SERCLK	P1B	21	N	Serial Clock
SYSCLK	P1A	10	N	System Clock
DMA				
BBSY*	P1B	1	Y	Bus Busy
BCLR*	P1B	2	N	Bus Clear
BERR*	P1C	11	Y	Bus Error
BG0IN*	P1B	4	Y	
BG1IN*	P1B	6	Y	Bus Grant In
BG2IN*	P1B	8	Y	
BG3IN*	P1B	10	Y	
BG0OUT*	P1B	5	Y	
BG1OUT*	P1B	7	Y	Bus Grant Out
BG2OUT*	P1B	9	Y	
BG3OUT*	P1B	11	Y	
BR0*	P1B	12	Y	
BR1*	P1B	13	Y	Bus Request
BR2*	P1B	14	Y	
BR3*	P1B	15	Y	

10.1 VMEbus Interface Signals (continued)

Mnemonic	Conn.	Pin	Used By 753	Description
Interrupts				
IRQ1*	P1B	30	Y	
IRQ2*	P1B	29	Y	
IRQ3*	P1B	28	Y	Interrupt Request Levels
IRQ4*	P1B	27	Y	
IRQ5*	P1B	26	Y	
IRQ6*	P1B	25	Y	
IRQ7*	P1B	24	Y	
IACK*	P1A	20	Y	Interrupt Acknowledge
IACKIN*	P1A	21	Y	Interrupt Acknowledge In
IACKOUT*	P1A	22	Y	Interrupt Acknowledge Out
Miscellaneous				
ACFAIL*	P1B	3	Y	AC Failure
LWORD*	P1C	13	Y	Longword
RESERVED	P2B	3	N	Reserved
SERDAT*	P1B	22	N	Serial Data
SYSRESET*	P1C	12	Y	System Reset
WRITE*	P1A	14	Y	Write
Power				
+5V	P1A,B,C	32	Y	+5 VDC
+5V	P2B	1,13,32	Y	+5 VDC
+5V STDBY	P1B	31	N	+5 VDC Standby
+12V	P1C	31	N	+12 VDC
-12V	P1A	31	N	-12 VDC
GND	P1A	9,11,15,17,19	Y	Signal Ground
GND	P1B	20,23	Y	Signal Ground
GND	P2B	2,12,22,31	Y	Signal Ground
GND	P1C	9	Y	Signal Ground

10.2 Extended Storage Module Drive Interface

Several different pin–numbering systems define the SMD–E interface. Sections 10.2.1 and 10.2.2 list both CDC's method and the 60- and 26-pin high-density method Xylogics uses.

10.2.1 Pin-numbering for the A Cable

Signal Name	Pin No. Pair from CDC SMD-E Spec.	Pin No. Pair for 60-pin
UNIT SELECT TAG	22, 52	22, 52
UNIT SELECT 2^0	23, 53	23, 53
UNIT SELECT 2^1	24, 54	24, 54
UNIT SELECT 2^2	26, 56	26, 56
UNIT SELECT 2^3/TAG 5	27, 57	27, 57
TAG 1	1, 31	1, 31
TAG 2	2, 32	2, 32
TAG 3	3, 33	3, 33
BIT 0, BUS OUT	4, 34	4, 34
BIT 1, BUS OUT	5, 35	5, 35
BIT 2, BUS OUT	6, 36	6, 36
BIT 3, BUS OUT	7, 37	7, 37
BIT 4, BUS OUT	8, 38	8, 38
BIT 5, BUS OUT	9, 39	9, 39
BIT 6, BUS OUT	10, 40	10, 40
BIT 7, BUS OUT	11, 41	11, 41
BIT 8, BUS OUT	12, 42	12, 42
BIT 9, BUS OUT	13, 43	13, 43
OPEN CABLE DETECT	14, 44	14, 44
BIT 0, BUS IN (Unit Ready)	19, 49	19, 49
BIT 1, BUS IN (On-cylinder)	17, 47	17, 47
BIT 2, BUS IN (Seek Error)	16, 46	16, 46
BIT 3, BUS IN (Fault)	15, 45	15, 45
BIT 4, BUS IN (Write-protect)	28, 58	28, 58
BIT 5, BUS IN (Address Mark)	20, 50	20, 50
BIT 6, BUS IN (Index)	18, 48	18, 48
BIT 7, BUS IN (Sector Pulse)	25, 55	25, 55
PICK	29	29
HOLD	59	59
BUSY	21, 51	21, 51
TAG 4 (Bit 10)	30, 60	30, 60

10.2.2 Pin-numbering for the B Cables

Signal Name	Pin No. Pair from CDC SMD-E Spec.	Pin No. Pair for 26-pin
WRITE DATA	8, 20	8, 20
Ground	7	7
WRITE CLOCK	6, 19	6, 19
Ground	18	18
SERVO CLOCK	2, 14	2, 14
Ground	1	1
READ DATA	3, 16	3, 16
Ground	15	15
READ CLOCK	5, 17	5, 17
Ground	4	4
SEEK END	10, 23	10, 23
UNIT SELECTED	22, 9	22, 9
Ground	21	21
INDEX	12, 24	N/C
Ground	11	11
SECTOR	13, 26	N/C
Ground	25	25

Index

Index

A

B

C

D

Index

Index

N

O

P

R

S

Index

T

U

V

W

53 Third Avenue

Burlington, MA 01803