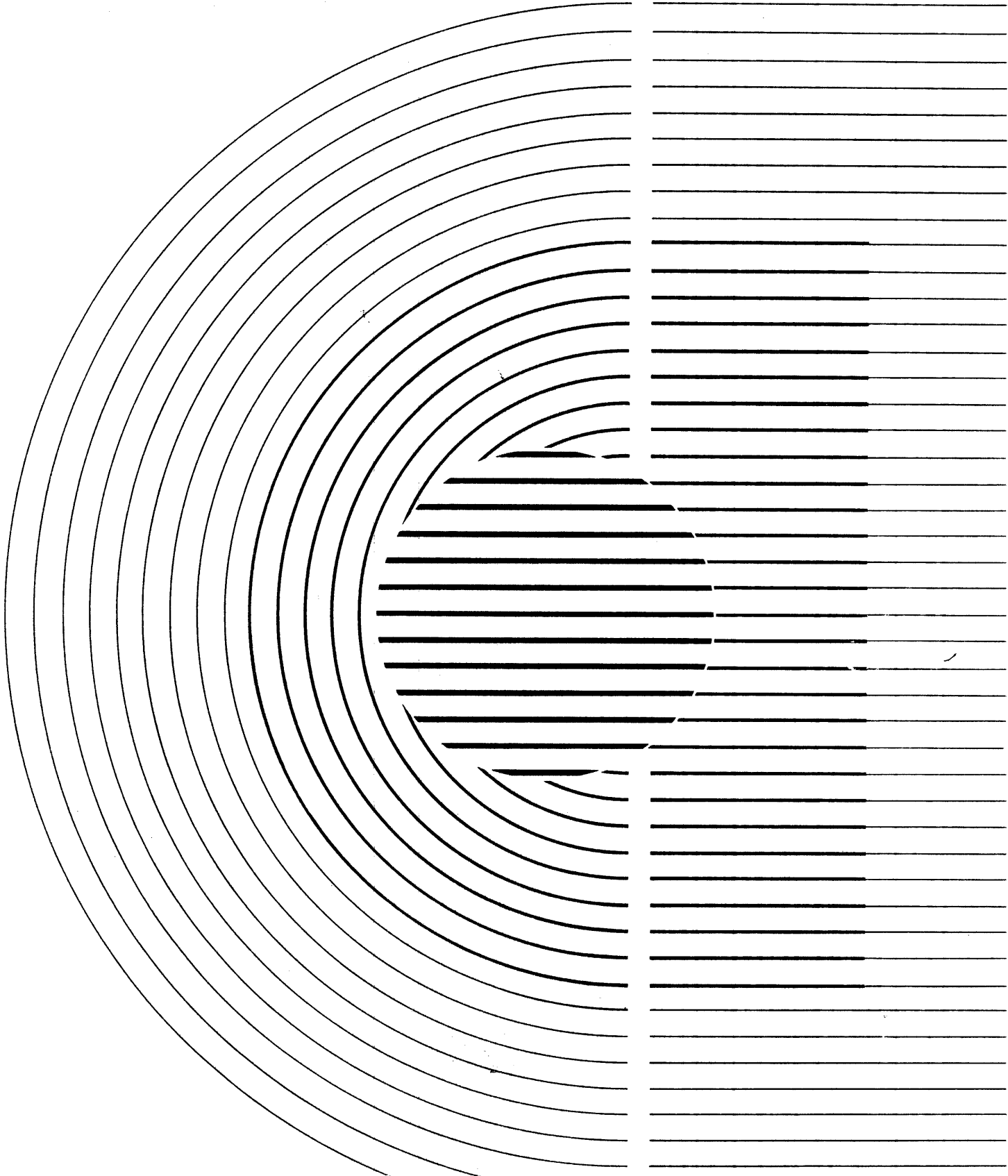




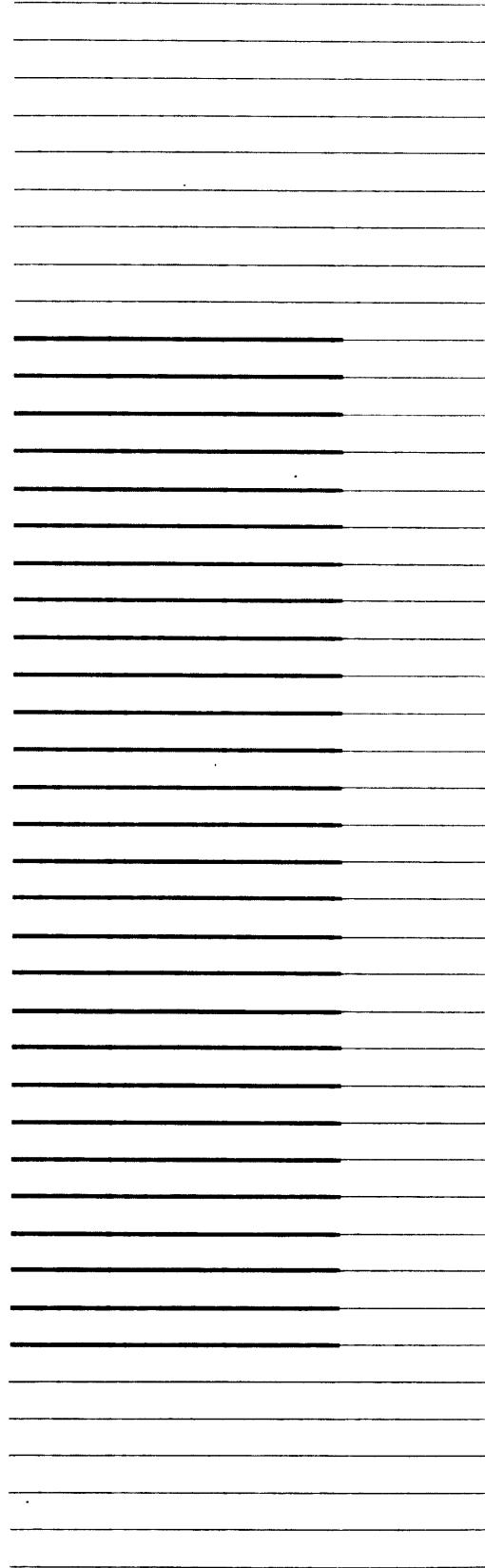
**adaptec, inc.**

**ACB-4000 Series  
User's Manual**



**ACB-4000 Series User's Manual  
5¼" Winchester Disk Controller**

October, 1985



## PREFACE

### COPYRIGHT

Copyright 1985 by Adaptec, Inc. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Adaptec, Inc., 580 Cottonwood Drive, Milpitas, California 95035.

### TRADEMARK

The trademarks referred to in this manual are identified and acknowledged as follows:

IBM and XT are registered trademarks of International Business Machines Corporation.

### WARRANTY

Adaptec, Inc. products are warranted against defects in material and workmanship for a period of one year. Refer to your Adaptec volume purchase agreement for complete warranty information.

### CHANGES

The material in this manual is for information only and is subject to change without notice.

Adaptec reserves the right to make changes in the product design without reservation and without notification to its users.

Adaptec, Inc. 580 Cottonwood Dr. Milpitas, CA 95035

Copyright 1985



## TABLE OF CONTENTS

SECTION	PAGE
<b>1.0 INTRODUCTION.....</b>	<b>1-1</b>
1.1 Scope and Purpose of Manual.....	1-1
1.2 Reference Documents.....	1-1
1.3 Overview of Product.....	1-1
1.4 Block Diagram.....	1-4
1.5 Available Controllers.....	1-4
1.6 Hardware and Software Requirements.....	1-5
1.7 Product Specifications.....	1-5
1.7.1 Physical Dimensions.....	1-5
1.7.2 Power Requirements.....	1-5
1.7.3 Environmental Requirements.....	1-5
1.8 Quality Assurance.....	1-6
<b>2.0 THEORY OF OPERATION/GENERAL OPERATION.....</b>	<b>2-1</b>
2.1 General Operation.....	2-1
2.2 Performance Operations.....	2-2
2.2.1 Non-interleaving Operation.....	2-2
2.2.2 Sector Level Defect Skipping.....	2-2
2.3 Flexible Operation.....	2-4
2.3.1 Device Independence & Autoconfiguration.	2-4
2.3.2 Variable Sector Size.....	2-4

## TABLE OF CONTENTS

SECTION	PAGE
3.0 HARDWARE INSTALLATION.....	3-1
3.1 Introduction.....	3-1
3.2 Environmental Requirements.....	3-1
3.3 Unpacking.....	3-1
3.4 Board Layout.....	3-1
3.5 Integration of Controller and Drive.....	3-4
3.5.1 Drive Selection and Termination.....	3-4
3.5.2 Controller Jumper Selection.....	3-6
3.5.3 Controller and Drive Cabling.....	3-8
3.5.4 Controller Power Requirements and Grounding.....	3-10
3.6 Integration of Controller and Host Adapter.....	3-11
3.6.1 ACB-4000 Series SCSI Hardware Implementation.....	3-11
3.6.2 SASI and SCSI Host Adapter Considerations	3-12
3.6.3 Controller Addressing and Termination....	3-12
3.6.4 Controller and Host Adapter Cabling.....	3-13
3.7 Power Up Operation.....	3-13
4.0 SOFTWARE INSTALLATION.....	4-1
4.1 Introduction.....	4-1
4.2 Command Structure.....	4-1
4.3 Status Structure.....	4-2
4.4 Basic Functions and Responses.....	4-2
4.4.1 ACB-4000 Series SCSI Software Implementation.....	4-2

## TABLE OF CONTENTS

SECTION	PAGE
4.4.2 Request Sense.....	4-3
4.4.3 Re-zero Unit.....	4-3
4.4.4 Test Unit Ready.....	4-3
4.4.5 Mode Select.....	4-4
4.4.6 Format.....	4-4
4.4.7 Read Capacity.....	4-5
4.4.8 Write.....	4-5
4.4.9 Read.....	4-6
4.5 Examples.....	4-6
4.5.1 First Time Format.....	4-6
4.5.2 Format With Manufacturer's Defect List...	4-8
5.0 SCSI COMMAND SET.....	5-1
5.1 General Description.....	5-1
5.2 Command Descriptor Block (CDB).....	5-3
5.2.1 Class Code.....	5-5
5.2.2 Operation Code.....	5-5
5.2.3 Logical Unit Number.....	5-5
5.2.4 Command Specific Bits.....	5-5
5.2.5 Logical Block Address.....	5-5
5.2.6 Number of Blocks.....	5-5
5.2.7 Control Byte.....	5-5

## TABLE OF CONTENTS

SECTION	PAGE
5.3 Class 00 Command Description.....	5-5
5.3.1 Test Unit Ready (00 hex).....	5-7
5.3.2 Rezero Unit (01 hex).....	5-8
5.3.3 Request Sense (03 hex).....	5-9
5.3.4 Format Unit (04 hex).....	5-9
5.3.5 Read (08 hex).....	5-13
5.3.6 Write (0A hex).....	5-14
5.3.7 Seek (0B hex).....	5-16
5.3.8 Translate (0F hex).....	5-17
5.3.9 Set Error Threshold (10 hex).....	5-19
5.3.10 Read/Reset Usage Counter (11 hex).....	5-20
5.3.11 Write Data Buffer (13 hex).....	5-22
5.3.12 Read Data Buffer (14 hex).....	5-23
5.3.13 Mode Select (15 hex).....	5-23
5.3.14 Mode Sense (1A hex).....	5-28
5.3.15 Start/Stop Unit (1B hex).....	5-29
5.3.16 Receive Diagnostic (1C hex).....	5-29
5.3.17 Send Diagnostic (1D hex).....	5-31
5.4 Class 1 Command Description.....	5-35
5.4.1 Read Capacity (25 hex).....	5-36
5.4.2 Read (28 hex).....	5-37
5.4.3 Write (2A hex).....	5-38



## TABLE OF CONTENTS

SECTION	PAGE
5.4.4 Write and Verify (2E hex).....	5-39
5.4.5 Verify (2F hex).....	5-40
5.4.6 Search Data Equal (31 hex).....	5-41
5.5 Completion Status Byte.....	5-45
5.5.1 Request Sense (03 hex).....	5-46
5.5.2 Sense Bytes.....	5-47
<b>6.0 TROUBLESHOOTING.....</b>	<b>6-1</b>
6.1 Introduction.....	6-1
6.2 Self Diagnostics.....	6-3
6.3 Error Codes and Probable Causes.....	6-4



LIST OF APPENDIX

APPENDIX	PAGE
APPENDIX A ACB-4000 SERIES SCSI IMPLEMENTATION.....	A-1
A.1 Introduction.....	A-1
A.2 General Description of SCSI.....	A-1
A.3 Bus Signals.....	A-2
A.4 Bus Phases.....	A-4
A.4.1 Bus Free Phase.....	A-4
A.4.2 Selection Phase.....	A-4
A.4.3 Information Transfer Phases.....	A-5
A.4.3.1 Command Phase.....	A-7
A.4.3.2 Data Phase.....	A-7
A.4.3.3 Status Phase.....	A-7
A.4.3.4 Message Phase.....	A-7
A.4.4 Signal Restrictions Between Phases.....	A-8
A.5 Message System.....	A-9
A.5.1 Command Complete (00H).....	A-9
A.6 Bus Conditions.....	A-10
A.6.1 Attention Condition.....	A-10
A.6.2 Reset Condition.....	A-10
A.7 Phase Sequencing.....	A-11
A.8 Timing.....	A-11
A.8.1 Selection Abort Time.....	A-11
A.8.2 Bus Clear Delay.....	A-11

## LIST OF APPENDIX

APPENDIX	PAGE
A.8.3 Bus Set Delay.....	A-11
A.8.4 Bus Settle Delay.....	A-12
A.8.5 Cable Skew.....	A-12
A.8.6 Deskew Delay.....	A-12
A.8.7 REQ Response Timeout.....	A-12
A.8.8 Reset Hold Time.....	A-12
A.8.9 Select Timeout.....	A-12
APPENDIX B BASIC SCSI HOST ADAPTER.....	B-1
APPENDIX C HOW TO WRITE A SCSI I/O DRIVER.....	C-1
C.1 Bus Phase Considerations.....	C-1
C.2 I/O Driver Flow Chart.....	C-2
C.2.1 Selection Phase.....	C-2
C.2.2 Command, Data, Status and Message Phases.	C-3
C.3 Timing Considerations.....	C-4
C.4 Sample Driver Routine.....	C-4
APPENDIX D SASI TO SCSI CONVERSION AND DIFFERENCES.....	D-1
D.1 Quick Installation of the ACB-4000 Series Controllers.....	D-1
D.2 Introduction.....	D-1
D.3 Hardware.....	D-2
D.4 Software.....	D-7
D.5 Summary.....	D-9

## LIST OF APPENDIX

APPENDIX	PAGE
APPENDIX E HOST AND DRIVE INTERFACES.....	E-1
E.1 Host Adapter Interface.....	E-1
E.1.1 Host Adapter Interface - Physical.....	E-1
E.1.2 Host Adapter Interface - Electrical.....	E-3
E.2 Disk Drive Interface.....	E-4
E.2.1 Disk Drive Interface - Physical.....	E-4
E.2.2 Disk Drive Interface - Electrical.....	E-6
APPENDIX F DISK FORMAT.....	F-1
APPENDIX G ADVANCED EXAMPLES.....	G-1
G.1 Using Data Search Not Equal to Verify After Format.....	G-1
G.2 Translate, Format and Interleaved Operation.....	G-2
G.3 Creating a Defect List from a Drive Using Translate.....	G-6
G.4 Write Protecting a Drive and Recover from Blown Format.....	G-8
G.5 Using Send Diagnostics Command to Turn Off ECC and Retries.....	G-9
APPENDIX H ACB-4000A TO ACB-4070 CONVERSION.....	H-1
H.1 Drive Capability.....	H-1
H.2 Host Adapter Transfer Rate.....	H-2
H.3 Defect Handling.....	H-2



## LIST OF FIGURES

FIGURE	PAGE
3-1. ACB-4000A Board Layout.....	3-2
3-2. ACB-4070 Board Layout.....	3-3
3-3. One Drive, One Controller Termination Example.....	3-5
3-4. Two Drives, One Controller Termination Example.....	3-5
3-5. Controller Cabling.....	3-9
3-6. Connector J3 Pin Assignments.....	3-10
3-7. Single Point Grounding Scheme.....	3-11
5-1. Class 00 Commands (6-Byte Commands).....	5-4
5-2. Class 01 Commands (10-Byte Extended Block Address)	5-4
5-3. Test Unit Ready Command.....	5-7
5-4. Rezero Unit Command.....	5-8
5-5. Format Unit Command.....	5-9
5-6. Defect Data Block.....	5-12
5-7. Read Command.....	5-13
5-8. Write Command.....	5-14
5-9. Seek Command.....	5-16
5-10. Translate Command.....	5-17
5-11. Translate Data.....	5-18
5-12. Set Threshold Command.....	5-19
5-13. Read/Reset Usage Counter Command.....	5-20
5-14. Read/Reset Usage Counter Parameters.....	5-21
5-15. Write Data Buffer Command.....	5-22
5-16. Read Data Buffer.....	5-23
5-17. Mode Select Command.....	5-24
5-18. Mode Select Parameter List.....	5-24
5-19. Extent Descriptor List.....	5-25

## LIST OF FIGURES

FIGURE	PAGE
5-20. Drive Parameter List.....	5-26
5-21. Mode Sense Command.....	5-28
5-22. Start/Stop Unit Command.....	5-29
5-23. Receive Diagnostic Command.....	5-29
5-24. Receive Diagnostic Data.....	5-30
5-25. Send Diagnostic Command.....	5-31
5-26. Send Diagnostic Parameter Format.....	5-32
5-27. Read Capacity Command.....	5-36
5-28. Read Command.....	5-37
5-29. Write Command.....	5-38
5-30. Write and Verify Command.....	5-39
5-31. Verify Command.....	5-40
5-32. Search Data Equal Command.....	5-42
5-33. Search Data Equal Argument.....	5-43
5-34. Completion Status Byte.....	5-45
5-35. Request Sense Command.....	5-46
5-36. Request Sense Data.....	5-47
6-1. Request Sense Command.....	6-4
6-2. Request Sense Data.....	6-5
E-1. SCSI Bus Pin Assignments.....	E-2
E-2. Host Adapter Bus Termination.....	E-3
E-3. System Interconnect Diagram.....	E-4
E-4. Control Driver/Receiver Lines.....	E-6
E-5. ACB-4000 Series Data Receiver/Driver Pairs.....	E-7



## LIST OF TABLES

TABLE	PAGE
2-1. ACB-4000A Sectors Per Track.....	2-5
2-2. ACB-4070 Sectors Per Track.....	2-5
3-1. Definition of Jumpers.....	3-6
3-2. Controller to ST506/412 Connector Definition.....	3-8
3-3. Power Requirements.....	3-10
5-1. ACB-4000A/4070 SCSI Command Set.....	5-2
5-2. Class 00 Command Code Summary.....	5-6
5-3. Format Options.....	5-10
5-4. Class 01 Command Code Summary.....	5-35
5-5. Search Data Equal Argument.....	5-44
6-1. ACB-4000 Series Controller Self Diagnostics.....	6-3
6-2. Class 00 Error Codes in Sense Byte (Drive Errors).	6-6
6-3. Class 01 Error Codes in Sense Byte (Target Errors).	6-7
6-4. Class 02 Error Codes (System-Related Errors).....	6-7
A-1. Basic Functions.....	A-1
A-2. Bus Signals.....	A-2
A-3. Bus Phases.....	A-4
A-4. Information Transfer Phase.....	A-6
A-5. SCSI Bus Timing.....	A-13
E-1. Connector J2 Pin Assignments.....	E-5
E-2. Connector J0 and J1 Pin Assignments.....	E-6



## 1.0 INTRODUCTION

---

### 1.1 SCOPE AND PURPOSE OF MANUAL

The purpose of this manual is to guide the system integrator through a successful installation of any Adaptec ACB-4000 series SCSI Controller. This includes theory of operation, hardware and software installation, command definition and use, as well as troubleshooting information. Also included is the Small Computer System Interface (SCSI) operation, host adapter and I/O device driver design, and upgrade information for existing designs.

### 1.2 REFERENCE DOCUMENTS

- \* Proposed American National Standard (ANSI) X3T9.2/82-2 SCSI - Small Computer System Interface.
- \* Appropriate Host Adapter manual.
- \* Appropriate disk drive OEM manual.

### 1.3 OVERVIEW OF PRODUCT

The Adaptec ACB-4000 Series SCSI Controllers are high performance, 5-1/4" Winchester Disk Controllers for ST506/412 type drives that interface with the SCSI I/O bus. The ACB-4000 Series controllers consist of the following products:

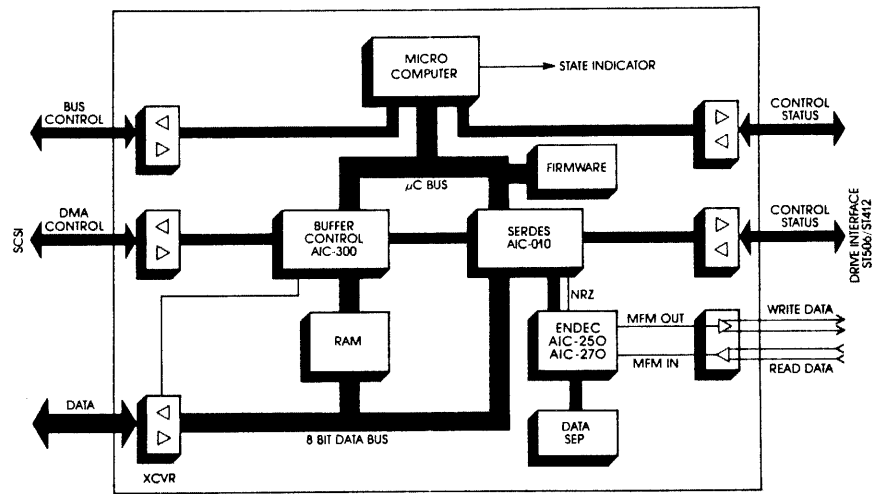
- O The ACB-4000A Controller. It supports SCSI features and controls two Winchester drives. The ACB-4000A also supports both hard-sectored and soft-sectored removable drives, as well as all ST506/412 type fixed drives. The ACB-4000A was previously known as two separate products, the ACB-4000 and ACB-4010. It is fully compatible with these products.
- O The ACB-4070 Controller. It supports SCSI features and controls two Winchester drives. The ACB-4070 uses 2,7 Run Length Limited (RLL) encoding on ST506/412 type drives compared to the ACB-4000A, which uses MFM encoding. The ACB-4070 is fully software compatible with the ACB-4000A, and gives 50% more capacity on the same ST506/412 drive.

The Adaptec ACB-4000 Series Controllers have the following features:

- 0 SCSI standard command set, plus SCSI extended optional and vendor unique commands. This gives the confidence of a ANSI standard I/O interface plus Adaptec's vast knowledge of SCSI and disk controllers.
- 0 1-to-1 interleaving of the disk. The 1K dual-ported buffer allows the host to read a track of data in a single revolution of the disk, and one cylinder of data without losing a revolution between heads. This gives the highest performance possible from the disk.
- 0 Sector level defect skipping. By flagging defective sectors, not tracks, lengthy seeks to alternate tracks are eliminated and disk capacity is maximized. This gives the fastest and most efficient method of handling defects on the disk.
- 0 Auto-configuration of the drive. The controller writes the drive characteristics onto the drive during format and reads them on power-up. This allows formatting on one controller and no need to reinitialize on another. This is ideal for field upgrade of drives with no user intervention. This eliminates the need for host initialization of the controller on power up. Also, this allows mixing drives of any capacity and manufacturer on the same controller.
- 0 Controls both 5-1/4" and 3-1/2" ST506/412 type Winchester disk drives. This allows the system integrator to interchange drives to optimize space requirements.
- 0 32-Bit ECC on both ID and data fields. This gives excellent data integrity and provides correction for single burst errors up to 8 bits long on the data field.
- 0 Variable sector sizes of 256 bytes, 512 bytes, and 1K bytes. This allows controller flexibility to meet your host-dependent block sizes.
- 0 Support of high speed data search gives you the ability to compare a full sector of data for equal or unequal data patterns. This allows quick determination of where defect bits are located within the sector.
- 0 Support of wedge servo drives such as Microscience, Syquest, Tulin, and Quantum. This gives drive flexibility and no need to order special firmware.

- 0 13 microsecond high performance seek step rate. This achieves maximum performance of the drive by reducing the seek access time. Many drives that are now on the market take advantage of this feature.
- 0 Supports 16 heads and 2048 cylinders. This allows use of high capacity drives such as Vertex and Maxtor.
- 0 Power-on diagnostics and jumper enabled self diagnostics for controller self checking. This gives a high confidence level that the controller is functioning properly.
- 0 The ACB-4000A supports hard sectored drives. This allows the use of drives which implement a sector pulse to identify the beginning of a sector.
- 0 The ACB-4000A supports removable media drives. The "Cartridge Changed," "Change Cartridge," and "Write Protect" functions are supported. This provides the flexibility to mix removable as well as fixed Winchester drives on the same controller.

## 1.4 BLOCK DIAGRAM



## 1.5 AVAILABLE CONTROLLERS

### MODEL NUMBER

### DESCRIPTION

ACB-4000A

SCSI Winchester Disk Controller that controls two ST506/412 fixed and removable drives. Hardware and software compatible with previous ACB-4000 and ACB-4010.

ACB-4070

SCSI Winchester Disk Controller with 2,7 Run Length Limited (RLL) encoding that controls two 2,7 RLL qualified ST506/412 drives. Software compatible with ACB-4000, 4010 and 4000A.

## 1.6 HARDWARE AND SOFTWARE REQUIREMENTS

### HARDWARE

1. ACB-4000A or ACB-4070 Controller Board.
2. Host CPU.
3. SCSI Host Adapter (or equivalent SCSI port).
4. 20-pin, 34-pin (drive), and 50-pin (SCSI) cables.
5. Power supply, +5 VDC and +12 VDC.

### SOFTWARE

1. Host operating system.
2. I/O driver for Host Adapter.
3. Uses written format utility.
4. (Optional) Landing zone utility.
5. (Optional) Any user defined controller utilities.

## 1.7 PRODUCT SPECIFICATIONS

### 1.7.1 PHYSICAL DIMENSIONS

Length: 7.75 inches (19.7 cm)  
Width: 5.75 inches (14.6 cm)  
Height: 0.75 inches (1.9 cm)

### 1.7.2 POWER REQUIREMENTS

<u>VOLTAGE</u> (VOLTS)	<u>TOLERANCE</u>	<u>CURRENT</u> (MAX. AMPS)	<u>RIPPLE</u> (VOLTS, RMS)
+5VDC	+/- 5%	1.5A	150 MV
+12VDC	+/- 10%	300 MA	150 MV

### 1.7.3 ENVIRONMENTAL REQUIREMENTS

	<u>OPERATING</u>	<u>STORAGE</u>
Temperature (Degrees) F/C	32/0 to 131/55	-40/-40 to 167/75
Humidity	10% to 95%	10% to 95%
Altitude (feet)	Sea level to 10,000	Sea level to 20,000
MTBF	20,000 POH at 55C	

Exhaust air flow may be required to keep air on both sides of the board at or below the maximum operating temperature if adequate convective ventilation is not available.

### 1.8 QUALITY ASSURANCE

The ACB-4000 Series of Controllers have been processed through Adaptec's extensive quality control procedure. All Adaptec custom IC's have been fully tested at temperature and voltage margins. All boards have been fabricated and assembled under close quality inspection. All boards have passed complete in-circuit test procedures, have been burned-in at elevated temperatures, and have been fully functionally tested. Adaptec should be notified immediately of any deviations from our high standard of quality.



## 2.0 THEORY OF OPERATION

---

### 2.1 GENERAL OPERATION

The ACB-4000 Series of controllers are SCSI to ST506/412 Winchester disk controllers.

The controllers' functions are divided into two general areas, the data path and control path. The data path, when writing data to the disk, consists of: the SCSI receivers, 1K RAM data buffer, 8-bit parallel to Non-Return to Zero (NRZ) serial conversion, NRZ encoding into MFM (or 2,7 RLL for the ACB-4070), and ST506/412 drivers. The data path, when reading data from the disk, consists of: ST506/412 receivers, data separation circuitry, MFM (or 2,7 RLL) to NRZ decoding, NRZ serial to 8-bit parallel, 1K data buffer and the SCSI drivers.

The controllers' data path design is based upon Adaptec's proprietary chip set. This chip set consists of the AIC-010 Winchester Disk Controller chip, the AIC-250 Encoder/Decoder chip, the AIC-270 2,7 RLL Encoder/Decoder chip and the AIC-300 Dual-Ported Buffer Controller.

The AIC-010 controls serialization and deserialization of data, Error Correction Code (ECC) generation and checking, and high speed data search. The AIC-250 converts NRZ data to and from MFM data, marks and detects ID and data fields, and precompensates write data. The AIC-270 converts NRZ data to and from 2,7 RLL data, and marks and detects ID and data fields. The AIC-300 controls the 1K-byte RAM to be a dual-port FIFO buffer and provides handshaking for data transfer to and from the buffer. The data path is capable of supporting a SCSI data transfer rate of 1.3 Mbytes per second.

The control path consists of: the SCSI drivers and receivers, microprocessor and ST506/412 drivers and receivers. The control path plus housekeeping for the data channel is handled by the on-board 8085 microprocessor. The functions handled by the 8085 are: SCSI command interpretation, SCSI control signals, drive head selection, drive step rate, drive control signals as well as data path housekeeping operations.

## 2.2 PERFORMANCE OPERATIONS

The ACB-4000 Series Controllers achieve high performance by using two important features: non-interleaved operation and sector level defect skipping operation. These features are found on all Adaptec SCSI products and are unmatched in the SCSI controller market.

### 2.2.1 NON-INTERLEAVED OPERATION

The ACB-4000 Series Controllers are capable of reading one track of data from the disk in one revolution of the disk. This is called non-interleaved, or 1-to-1 interleaved operation. In order to achieve this, the controller is able to read the sector ID and Data fields, check for correct ECC and pass data to the buffer before the next sector is under the disk head. Non-interleaved operation is achieved for 256 and 512 bytes per sector on the ACB-4000 Series Controllers. For a 1K sector size, data is written and read in a non-interleaved fashion except for multisector reads.

This speed gives the ACB-4000 Series Controllers the ability to read an entire cylinder of data without losing a revolution of the disk during head switches. Since one revolution of the disk is 16.7 microseconds, this is an important contribution to speed when reading large sequential files.

### 2.2.2 SECTOR LEVEL DEFECT SKIPPING

Another important contribution to the speed of reading data from the disk is how the controller handles defects. All Winchester disks have some sort of media imperfection due to the nature of media processes. In SCSI, the controller is responsible for mapping the defective areas of the disk that are unusable for data storage to areas that are free from defects.

All other controllers mark an entire track as unusable and assign an alternate track at the inner cylinders of the drive to be the defect free track. An entire track of 10,416 bytes is flagged as defective when usually the defect is only 10 or less bits long! This wastes a great deal of usable data space due to the inability of the controller to handle defects efficiently.

The most important performance degradation is due to the controller moving the drive heads to the alternate track. In sequential reads, when a defective track is encountered, the controller must determine the alternate track, seek to the track, wait an average latency time of the disk (8.3 milliseconds), read the data and seek back to the next sequential track. This may take 120 milliseconds for a drive with a 30 millisecond average access time, just to handle one 10-bit defect.

An important feature of the ACB-4000 series controllers is the use of sector level defect skipping.

At format time, a defective sector is marked bad, then rewritten with the same logical address in the next logical location of the disk (the physical location changes according to the interleave factor). By handling defects in this manner, only one sector of useable data will be mapped out for each bad sector. All subsequent sectors are then moved toward the end of the disk one physical sector location.

To prevent an increase in access times due to the physical change of location, as a consequence of bad sectors, the drive is divided into 18 zones. During power-up diagnostics the drive seeks to the innermost cylinder on the drive, then seeks back to track 0, stopping in each of the 18 zones to check the number of mapped out sectors in the zone. The controller then maintains this information, enabling the controller to determine if a logical sector has moved to a different physical head or cylinder.

One important difference to note between the ACB-4000A and ACB-4070 in handling defects is the nature of the different encoding schemes. Most drive manufacturers give their defect listing in cylinder, head and bytes from index in MFM coding. With 2,7 RLL each track now has 50% more bits per track. The defect map must be converted for 2,7 RLL when using the ACB-4070. In order to do this the following formula must be followed.

$$\text{MFM BYTES FROM INDEX} \times 1.5 = \text{2,7 RLL BYTES FROM INDEX}$$

If the result is a fraction, round up to the next integer, e.g., 7.5 rounds to 8. This 2,7 RLL information must be used at format time to determine the location of defects on the disk. If using the TRANSLATE command to determine the location of defects on the disk, the bytes from index value returned is in 2,7 RLL encoding. See the example in Appendix G.

## 2.3 FLEXIBLE OPERATION

### 2.3.1 DEVICE INDEPENDENCE AND AUTOCONFIGURATION

Adaptec's device independence allows flexibility not provided with other hard disk controllers. Device independence allows the user of an Adaptec controller to configure each drive independent of the other drives, and independent of system software.

Device independence is obtained through the use of the MODE SELECT and FORMAT commands during the installation of the hard disk(s).

During FORMAT the disk parameters provided by the MODE SELECT command will be encoded onto the disk. These parameters are written to the ID field of cylinder 0. This information is then read to the controller during the power-up initialization process. Access to these parameters is also available using the MODE SENSE command.

This scheme obsoletes the need for the host software to initialize the controller on power up. This device independence allows the system to be completely unaware of the exact physical characteristics of the drive.

### 2.3.2 VARIABLE SECTOR SIZE

The ACB-4000A and ACB-4070 controllers are capable of handling variable sector sizes. The sector size is selectable to be 256, 512 and 1024 bytes per sector. These sector sizes allow easy integration of the controller into your specific application.

The ACB-4000 Series Controllers are capable of adding one more sector per track than other controllers. The reason Adaptec is able to add this extra sector is due to the superior data separator design. The design needs little sync time between sectors so the gaps between sectors can be less.

See Tables 2-1 and 2-2 for the variable sector sizes and related information. Appendix F shows details of the sector format.

TABLE 2-1. ACB-4000A SECTORS PER TRACK

<u>SECTOR SIZE</u>	<u>BYTES/SECTOR</u>	<u>INTERLEAVE</u>	<u>SECTORS/TRACK</u>
(DATA FIELD)	(TOTAL BYTES)		
256	320	1	32
256	310	>1	33
512	576	1	17
512	566	>1	18
1024	1088	1	9
1024	1078	>1	9

TABLE 2-2. ACB-4070 SECTORS PER TRACK

<u>SECTOR SIZE</u>	<u>BYTES/SECTOR</u>	<u>INTERLEAVE</u>	<u>SECTORS/TRACK</u>
(DATA FIELD)	(TOTAL BYTES)		
256	333	1	46
256	323	>1	47
512	589	1	25
512	579	>1	26
1024	1101	1	14
1024	1091	>1	14

Since 2,7 RLL encoding increases the actual number of bytes on a track by 50%, more sectors per track are achievable. The ACB-4070 increases the number of sectors per track 42% (for 256 bytes per sector and >1 interleave) to 55% (for 1024 bytes per sector) over the ACB-4000A controller. An added benefit to more sectors per track is that less seeks are required of the drive.

### 3.0 HARDWARE INSTALLATION

---

#### 3.1 INTRODUCTION

This section describes the steps necessary to install the ACB-4000A and 4070 hardware. First the operating environment, unpacking procedure, and board layout are described. This section also describes the integration of the drive, controller, and host adapter.

#### 3.2 ENVIRONMENTAL REQUIREMENTS

The ACB-4000 Series Controllers will perform properly over the following range of conditions:

	<u>Operating</u>	<u>Storage</u>
Temperature (F/C)	32/0 to 131/55	-40/-40 to 167/75
Humidity	10% to 95%	10% to 95%
Altitude, feet	Sea level to 10,000	Sea level to 20,000
MTBF, Hours	20,000 POH @ 55 C	

#### 3.3 UNPACKING

The carrier is responsible for damage incurred during shipment. In case of damage, have the carrier note the damage on both the delivery receipt and the freight bill, then notify your freight company representative so that the necessary insurance claims can be initiated.

After opening the shipping container, use the packing slip to verify receipt of the individual items listed on the slip. Retain the shipping container and packing material for possible later reuse should return of the equipment to the factory be necessary.

#### **NOTE:**

THE ACB-4000A AND 4070, LIKE ALL ELECTRONIC EQUIPMENT, ARE STATIC SENSITIVE. PLEASE TAKE THE PROPER PRECAUTIONS WHEN HANDLING THE BOARD. KEEP THE BOARD IN ITS CONDUCTIVE WRAPPING UNTIL IT IS CONFIGURED AND READY TO BE INSTALLED IN YOUR SYSTEM.

#### 3.4 BOARD LAYOUT

The ACB-4000A is shown in Figure 3-1. The ACB-4070 is shown in Figure 3-2. These figures show the location of the firmware, key components, terminators, jumpers and connectors.

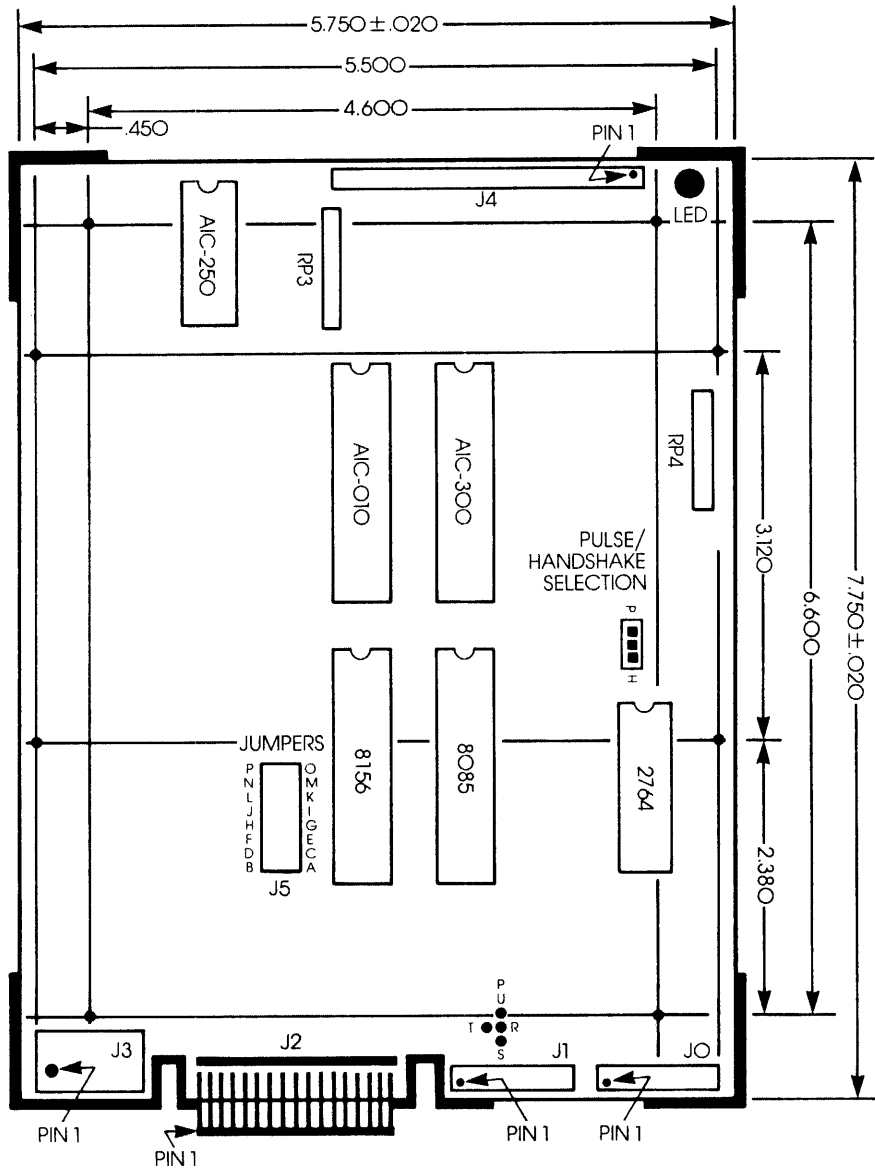


FIGURE 3-1. ACB-4000A BOARD LAYOUT

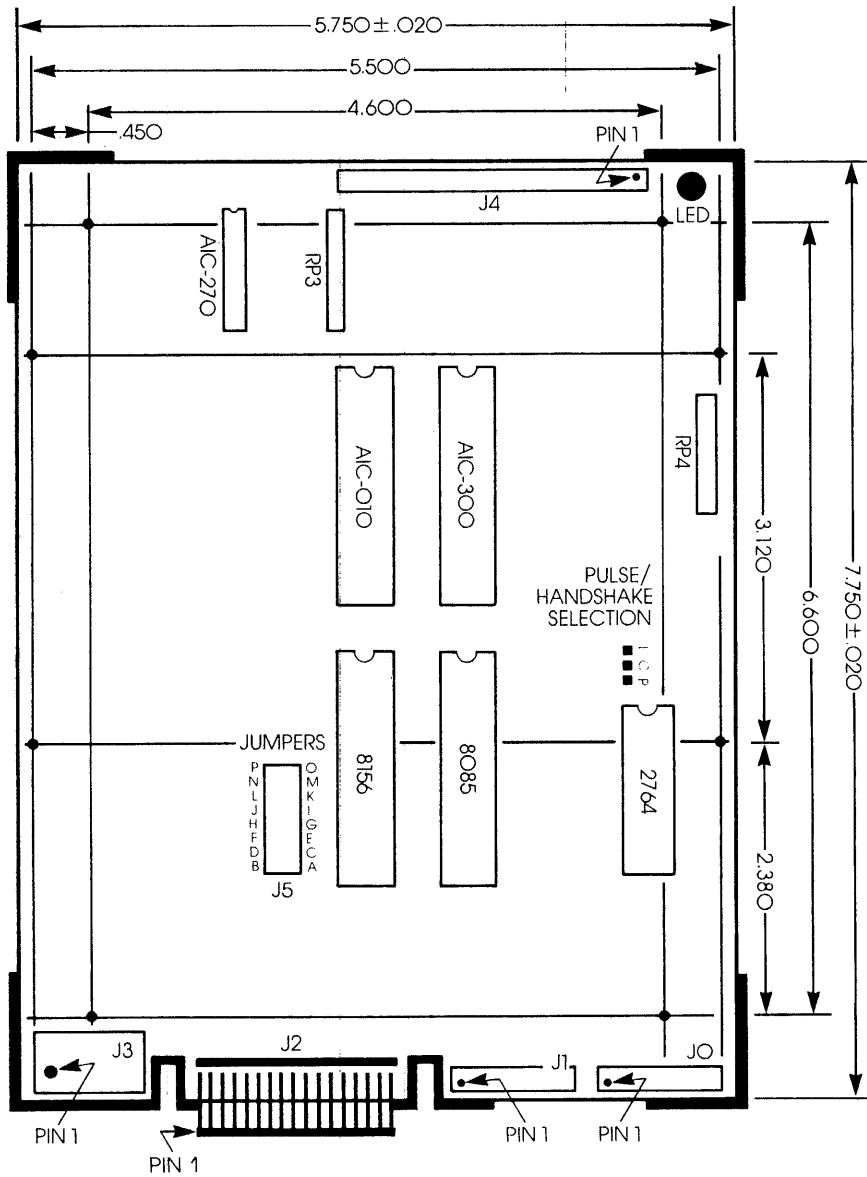


FIGURE 3-2. ACB-4070 BOARD LAYOUT



### 3.5 INTEGRATION OF CONTROLLER AND DRIVE

To install the Adaptec ACB-4000A or 4070 board into your system you must first configure the drive(s), set the controller jumpers and connect the drive cables properly. This section describes all the necessary steps needed to successfully install this hardware.

In order to configure the drives, you need the Disk Drive OEM Manual that was supplied with your drive. (If you do not have this manual call your Drive vendor for assistance.) This manual is required to obtain the drive characteristics for your particular drive.

#### 3.5.1 DRIVE SELECTION AND TERMINATION

The drive changeable parameters are the drive selection switches (or jumpers) and the drive termination. These parameters allow a drive to be selected as drive 0, 1, 2 or 3. This is accomplished by changing the drive address selection switches or jumpers.

**NOTE:**

SOME DRIVE MANUFACTURERS HAVE DESIGNATED THE DRIVE ADDRESSES TO BE 1,2,3,4 INSTEAD OF 0,1,2,3. DO NOT SET THE DRIVE AS A RADIAL SELECTED DRIVE. RADIAL SELECTION WILL SET ALL DRIVE OUTPUT SIGNALS TO BE ACTIVE, EVEN IF THE DRIVE IS NOT SELECTED. IN THIS CASE, THE DRIVE LED WILL BE ON AT ALL TIMES.

Use the two lowest drive addresses available as drive "0" and drive "1" to be seen by the controller.

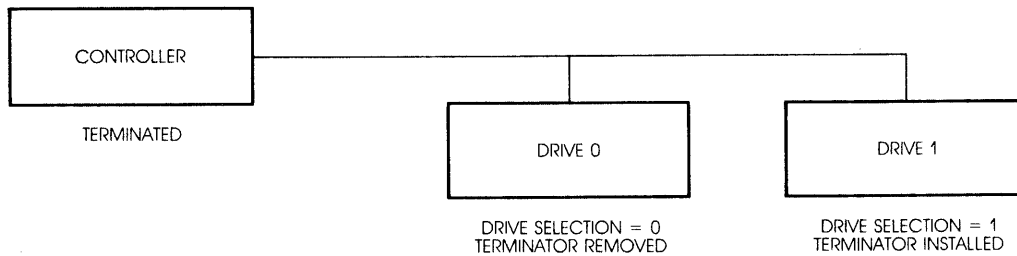
Before the drives can be cabled to the controller the drive cable terminator must be properly set. The terminator is used to reduce signal "ringing" in the cables. The terminator, as its name implies, must be at the end of each cable in order to have the controller and drive communicate properly. The controller has a permanent terminator built-in. The disk drives, since they can be connected in a daisy chain configuration have a removable terminator. This is usually a 16-pin DIP resistor package. The last physical drive in the chain must have its terminator installed (see Figures 3-3 and 3-4).

Example 1: When one drive and one controller are used, both must be terminated.



**FIGURE 3-3. ONE DRIVE, ONE CONTROLLER TERMINATION EXAMPLE**

Example 2: When two drives and one controller are used, only the last one in the chain is terminated.



**FIGURE 3-4. TWO DRIVES, ONE CONTROLLER TERMINATION EXAMPLE**

Now select the proper drive addresses and remove or install the required terminators for your configuration.

### 3.5.2 CONTROLLER JUMPER SELECTION

The controller changeable parameters are defined as the variables that can be changed on the controller to accommodate the different drive characteristics. These parameters can easily be changed by use of the MODE SELECT command and by the use of the jumpers defined in Table 3-1.

TABLE 3-1. DEFINITION OF JUMPERS

<u>Jumper</u>	<u>Definition</u>	<u>Installed</u>	<u>Removed</u>
A-B	LSB of Controller's SCSI Address	bit = 1	bit = 0
C-D	Controller's SCSI Address	bit = 1	bit = 0
E-F	MSB of Controller's SCSI Address	bit = 1	bit = 0
G-H	DMA Transfer Rate	SYSCLOCK/4	DATALOCK/2
I-J	Extended Command Set (See Appendix D)	Enabled	Disabled
K-L	Not Used		
M-N	Support of Syquest 312, DMA 360 and drives that drop SEEK COMPLETE signal during head switching	Enabled	Disabled
O-P	Self Diagnostics	Enabled	Disabled
R-PU *	Write Precompensation turned off for both drives	Enabled	Disabled
R-S *	Write Precomp starts at same cylinder as reduced write current for both drives	Enabled	Disabled
R-T *	Write Precomp is applied to all tracks for both drives	Enabled	Disabled

NOTE: \* denotes that these jumpers are for the ACB-4000A only, they are not present on the ACB-4070.

The jumpers are divided into five categories: SCSI address, host adapter options, drive options, write precompensation options and self diagnostics.

Jumpers A-B, C-D and E-F are used to select the controller's SCSI device ID or address. They determine how the controller will be identified when installed in the SCSI bus.

Jumper G-H is used to select two DMA transfer rates on the SCSI bus. When removed, the controller will transfer data at the maximum rate allowed by the controller, equal to  $\text{DATA\!CLOCK}/2$ . Some host adapters cannot support this data rate. When this jumper is installed, the data transfer rate is reduced to run at a rate of  $\text{SYSCLOCK}/4$  on single sector transfers. This is one-half of the controller's maximum DMA speed. Multisector transfers are always made at the maximum rate of  $\text{DATA\!CLOCK}/2$ , with or without this jumper.

Jumper I-J is used to enable the extended command set. This is used only when replacing SASI-type controllers. See Appendix D for details.

Jumper M-N is installed to support drives that drop the SEEK COMPLETE line on the ST506/412 interface during head switches. Soft-sectored removable media drives that have wedge servo information of all surfaces require this jumper to be installed. Syquest 312 10 MB and DMA 360 10 MB drives require this jumper to be installed. The installation of this jumper does not affect operation of drives that do not drop SEEK COMPLETE between head switches.

Jumpers R-PU, R-S and R-T are used to select write precompensation for both drives. Only one of these options can be used at a time. This hardware jumper overrides any software selection made in the MODE SELECT command (see Section 5). The write precompensation used is 12ns. The ACB-4070 controller does not have these jumpers since 2,7 RLL encoding does not use write precompensation. Please see Figure 3-1 for location of these jumpers.

**NOTE:**

ON THE ACB-4000A, THESE JUMPERS ARE ORIENTED DIFFERENTLY THAN ON THE ACB-4000.

Jumper O-P is used to enable the ACB-4000 Series Controllers' Self Diagnostics. These diagnostics test the internal circuitry of the controller and can be used for incoming inspection of boards and test the internal circuitry of the controller. See Section 6 for operation and associated error codes.

Configure the jumpers at this time to meet your drive and system requirements.

### 3.5.3 CONTROLLER AND DRIVE CABLING

Now that the drive and controller are configured, they can be connected together. The controller has three drive cable connectors J0, J1, and J2. These are described in Table 3-2.

**TABLE 3-2. CONTROLLER TO ST506/412 CONNECTOR DEFINITIONS**

<u>Connector</u>	<u>Signals</u>	<u>Cable</u>
J0	Data	20-pin flat ribbon cable Connected to drive 0
J1	Data	20-pin flat ribbon cable Connected to drive 1
J2	Control	34-pin flat ribbon cable Connected to both drives 0 and 1

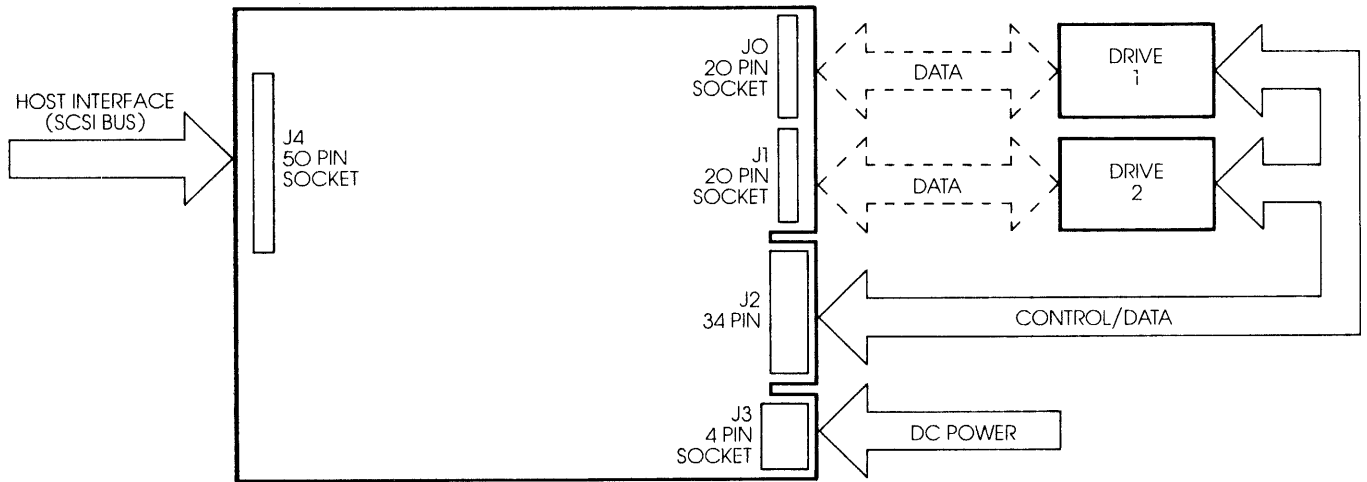
<u>Connector</u>	<u>Recommended Plug</u>	<u>Maximum Length</u>
J0	3M Part # 3421	20 feet (6 meters)
J1	3M Part # 3421	20 feet (6 meters)
J2	3M Part # 3414	20 feet (6 meters)

The connector locations and pin orientation for the ACB-4000A connectors are shown in Figure 3-1, and for the ACB-4070 in Figure 3-2.

**NOTE:**

PIN 1 OF J0, J1 AND J2 IS LOCATED ON THE SIDE CLOSEST TO THE POWER CONNECTOR.

Connect the cables as shown in Figure 3-5.



**FIGURE 3-5. CONTROLLER CABLING**

### 3.5.4 CONTROLLER POWER REQUIREMENTS AND GROUNDING

The power requirements for the ACB-4000 Series Controllers are shown in Table 3-3.

TABLE 3-3. POWER REQUIREMENTS

<u>Voltage</u> (Volts)	<u>Tolerance</u>	<u>Current</u> (Max. Amps)	<u>Ripple</u> (Volts, RMS)
+5 VDC	+/- 5%	1.5 A	150 mV
+12 VDC	+/- 10%	300 mA	150 mV

The power is applied through the 4-pin connector J3. The recommended mating connector is AMP Part # 1-480424-0. Connector J3 pin assignments are shown in Figure 3-6.

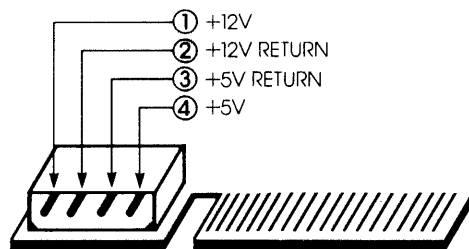
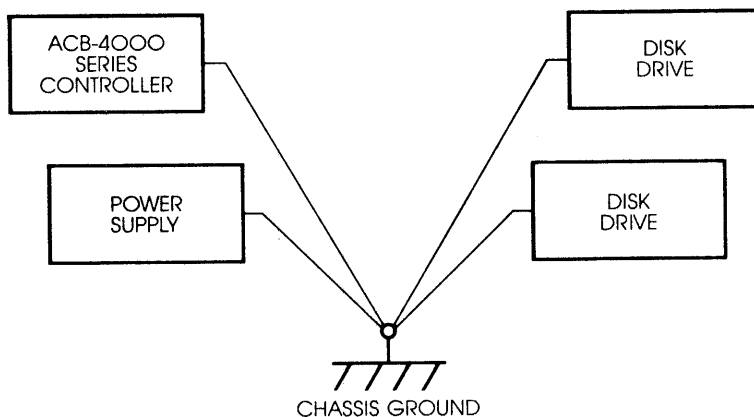


FIGURE 3-6. POWER CONNECTOR J3

The ACB-4000 Series Controller and attached Disk Drives should be grounded using a Single Point Grounding Scheme. This scheme connects all grounds from controller, drives, plus all other major components within the cabinet to one point that is then connected to the chassis ground. See Figure 3-7.

**NOTE:**

IF PROPER GROUNDING IS NOT FOLLOWED, RANDOM FORMAT, WRITE AND READ ERRORS MAY OCCUR.



**FIGURE 3-7. SINGLE POINT GROUNDING SCHEME**

**3.6 INTEGRATION OF CONTROLLER AND HOST ADAPTER**

Now that the controller and drive have been configured, the controller must now be attached to the SCSI host adapter. The successful integration of controller and host adapter must take into account controller addressing, termination, cabling and the exact implementation of SCSI used by the host adapter and controller. This section addresses these topics.

**NOTE:**

IF A SASI-TYPE HOST ADAPTER IS USED, OR IF YOU ARE REPLACING A SASI-TYPE CONTROLLER WITH THE ACB-4000 SERIES CONTROLLERS, PLEASE SEE APPENDIX D FOR TIMING AND SOFTWARE CONSIDERATIONS.

**3.6.1 ACB-4000 SERIES SCSI HARDWARE IMPLEMENTATION**

The ACB-4000 Series Controllers support the proposed ANSI Standard X3T9.2/82-2 Revision 14, Small Computer System Interface (SCSI).



The ACB-4000 Series supports 17 active lines and 25 ground lines in a 50-pin flat cable. All odd number pins are ground. This is known as an unbalanced SCSI bus. The bus has open-collector drivers and is terminated at both ends by a 220 ohm pull-up resistor to +5V and a 330 ohm pull-down resistor to ground. The bus is low-active, thus a grounded line is considered active or asserted.

Eight of the lines are the byte-wide bi-directional data bus. The ACB-4000 Series supports DB7-0 data lines and does not support the data parity line. The data parity line is not terminated at the controller.

Nine of the lines are the control and status lines. The ACB-4000 Series supports all nine lines including BSY, SEL, C/D, I/O, MSG, REQ, ACK, ATN AND RST.

The RST, reset line, is a "hard" reset and causes the controller to abort its current operation and to get off of the bus. Any outstanding operation is aborted. The SCSI specification has a minimum RST pulse width of 25 microseconds. The ACB-4000 Series allows a minimum RST pulse width of 50 nanoseconds to accommodate older SASI-type host adapters.

### 3.6.2 SASI AND SCSI HOST ADAPTER CONSIDERATIONS

When using a SASI or SCSI host adapter the following areas must be considered: SCSI handshake timing, transfer rate of controller, use of ATN line, and I/O driver design termination. Appendices A, B, C, and D address these topics.

### 3.6.3 CONTROLLER ADDRESSING AND TERMINATION

In order for the ACB-4000 Series Controllers to operate properly in a system environment, the controller must be properly selected and terminated. The controller can be jumpered to be selected as SCSI device ID or address 0 through 7. See Table 3-1 for the definition of these jumpers and install any that are needed.

In a single controller system use the controller SCSI ID = 0 which requires no jumpers for locations A-B, C-D and E-F on the controller.

The SCSI bus is a daisy chained bus between host adapter and I/O controllers. In order to reduce signal ringing, the two ends of the bus must be terminated. This termination consists of a 220-ohm resistor to +5 Volts and a 330-ohm resistor to ground for each signal line. Check your host adapter to see if it is terminated. The ACB-4000 Series Controllers have socketed terminator packs located at RP3 and RP4 for this termination as noted in Figures 3-1 and 3-2.

In a single controller system both must be terminated, i.e. RP3 and RP4 must be installed. In a multiple I/O controller system only the last controller must be terminated, i.e. RP3 and RP4 must be installed only if the ACB-4000 Series Controller is the last one in the chain.

#### 3.6.4 CONTROLLER AND HOST ADAPTER CABLING

Now that the controller and host adapter are addressed and terminated properly, they can be cabled together.

The controller 50-pin connector J4 is the SCSI bus connector. A 50-pin flat ribbon cable with a maximum length of 20 feet (6 meters) is required. A 3M Part # 3425-3000 cable connector is recommended.

#### 3.7 POWER UP OPERATION

If the drive is correctly formatted, the controller will seek the drive to the last cylinder and read the largest block address present. The parameter information and largest block address are saved on the ACB-4000 Series Controller.

Once the last block address has been read, the controller will seek the drive back to Track 0, stopping several times in 'zones' on the way back to read the defect count at that point. This defect count is also saved in the controller to allow the controller to better predict the location of the block on the disk.

In addition to the drive seeks and reads, the Adaptec ACB-4000 Series Controllers do a series of self-diagnostics after power-up. Immediate selection of the disk and movement of the heads during this period is a sign of properly functioning Adaptec ACB-4000 Series Controllers.

When power is supplied to the system, the controller will enter a power-up mode and wait for a minimum of 18 seconds for the drive to become ready. During the 18 second power-on sequence, the controller is checking for drive 0 and drive 1 to become ready (nine sec/driver). If the host senses a command requiring access to a drive before it has become ready (and before 18 seconds have elapsed) the controller will accept the command and continue to check for a ready status. Once the drive comes ready, the controller will then execute the command; if 18 seconds elapse and the drive does not come ready, a DRIVE NOT READY (04 hex) error will result. The controller will then check for a ready status on the next command requiring access to that drive.

Once a drive comes ready, the controller will recalibrate the head to track 0 if needed. If the drive started at track 0, the controller will step the head off of track 0 to confirm that the drive can seek and that the track 0 signal was valid. With the drive's ability to seek confirmed, the controller then seeks back to track 0. The drive actuator (if it can be seen) appears to make a short 'blip.'

The controller then attempts to read from track 0 parameter information which is written during formatting. If the drive is unformatted or had been formatted by another vendors controller, the parameter information is not present so the controller then sets a bit in its memory called 'blown format' to warn the user that the drive is unuseable. If the drive format is blown, the reset sequence is stopped and the controller is ready for a command. The drive must be formatted to allow a READ or WRITE access to disk data.

**NOTE:**

IF THE CONTROLLER DOES NOT OPERATE AS DESCRIBED, PLEASE SEE SECTION 6 FOR TROUBLESHOOTING PROCEDURES.

## 4.0 SOFTWARE INSTALLATION

---

### 4.1 INTRODUCTION

This section describes the fundamental software operation and installation of the ACB-4000 Series Controllers. This includes the structure of command and status, the basic commands needed to initially get the host adapter and controller to function, plus detailed examples. This section will give you enough information to begin using the ACB-4000 Series Controllers. Once you have mastered this chapter, higher level commands can be added to make full use of the controllers' features. All commands are described in Section 5 of this manual.

**NOTE:**

FOR MORE INFORMATION ABOUT THE FOLLOWING TOPICS PLEASE SEE THE APPROPRIATE APPENDICES:

SCSI COMMAND SET	SECTION 5
TROUBLESHOOTING PROCEDURES	SECTION 6
SCSI	APPENDIX A
WRITING AN SCSI I/O DRIVER	APPENDIX C
CONVERTING FROM SASI TO SCSI	APPENDIX D
ADVANCED EXAMPLES	APPENDIX G

IF YOU ARE DESIGNING YOUR SOFTWARE FROM SCRATCH, PLEASE READ APPENDIX C BEFORE READING THIS CHAPTER.

IF YOU ARE REPLACING A LOWER PERFORMANCE SASI CONTROLLER WITH THE ACB-4000 SERIES, PLEASE READ APPENDIX D BEFORE READING THIS CHAPTER.

### 4.2 COMMAND STRUCTURE

An I/O request from a host is made by passing a COMMAND DESCRIPTOR BLOCK (CDB) to the controller. The CDB provides information that the controller needs to process the command correctly. See Section 5.2 for a detailed description of the CDB.

**NOTE:**

THE ACB-4000 SERIES CONTROLLERS REQUIRE THAT RESERVED BYTES MUST BE SET TO ZERO OR THE COMMAND WILL BE REJECTED.

### 4.3 STATUS STRUCTURE

Status is always set by the controller at the end of a command. Any abnormal condition encountered during the command execution causes command termination and ending status. Normal operation presents a return status of 00. See Section 5.5 for a detailed description.

### 4.4 BASIC FUNCTIONS AND RESPONSES

#### 4.4.1 ACB-4000 SERIES SCSI SOFTWARE IMPLEMENTATION

The ACB-4000 Series of SCSI Controllers meets the ANSI specification X3T9.2/82-2 Revision 14. The controllers support all hardware and software standards for Direct Access Devices as defined by this specification. The key options that are not supported by the ACB-4000 Series are command chaining disconnect/reconnect, arbitration and parity. These are fully implemented in Adaptec's ACB-5500 Series of Controllers.

The SCSI software command set supported by the controllers is a combination of standard, optional, extended and vendor unique commands, both Class 0 and Class 1 commands. These commands were chosen to give you the highest performance and flexibility for a SCSI Winchester Disk Controller. These commands are completely compatible with the Adaptec ACB-5500 Series of Controllers. Table 5-1 shows the SCSI command set as implemented in the ACB-4000 Series Controllers.

#### 4.4.2 REQUEST SENSE

The REQUEST SENSE (03 hex) command is the command that gives the most detailed description of status from the controller. This command is needed whenever a CHECK STATUS is found in the COMPLETION STATUS BYTE. If the CHECK BIT is set after a command, a REQUEST SENSE must follow in order to read status and clear the check condition.

See Sections 5.5.1 and 5.5.2 for details. Section 6.3 gives details of any error codes and probable causes, should they occur.

#### 4.4.3 REZERO UNIT

The REZERO UNIT command is the lowest level command that can be performed to verify that the host adapter, controller and drive are functioning properly. This command should be done first in any testing or I/O driver troubleshooting to insure commands are being serviced properly. See Section 5.3.2 for details.

This command will move the heads of the selected drive to track zero and return COMPLETION STATUS. The command is shown below.

A COMPLETION STATUS may give a check condition that leads to the possible errors of: No Seek Complete, Drive Not Ready and No Track Zero.

#### 4.4.4 TEST UNIT READY

The TEST UNIT READY command verifies that the drive is powered on and the DRIVE READY line is true. Once the drive is rezeroed, with no errors, this command will check to see that the drive is ready to write and read data. Drive write fault condition is also checked. See Section 5.3.1 for details.

A COMPLETION STATUS may give a check condition that leads to the possible errors of: Drive Not Ready and Write fault.

#### 4.4.5 MODE SELECT

Now that the drive is rezeroed and ready, it next must be prepared for writing and reading. To do this, the drive needs to be characterized for auto-configuration and formatted.

Adaptec's MODE SELECT command is used to define the drive unique characteristics. The FORMAT command writes the drive parameters in the ID fields of Track 0, and cylinder 0. These are retrieved when the drive is later powered on. These drive parameters do not use data fields, and thus do not affect the user data area. This command eliminates the requirement of the I/O driver to tell the controller what type of drive is connected to it every time that power is applied.

The MODE SELECT command is used before formatting a drive and should be part of your format utility. This command can also be used to write protect a drive as shown in Appendix G.

The MODE SELECT command must precede the FORMAT command or the FORMAT command will be rejected. See Section 5.3.13 for details.

A COMPLETION STATUS may give a check condition that leads to a possible Bad Argument error.

The information written to the disk can be read by use of the MODE SENSE command. See Section 5.3.14 for details.

#### 4.4.6 FORMAT

The FORMAT command writes drive characteristics, ID and data fields onto the drive and writes a fill pattern into the user data field. This fill pattern can be changed by use of the FORMAT command. This feature is useful in writing worst case data patterns onto the drive at format time.

When using the FORMAT command, a drive defect list can be appended to the command in a cylinder, head and "bytes from index" form. This form is the same form that most drive manufacturers use. This form can also be generated from the drive by using the TRANSLATE with the SEARCH DATA NOT EQUAL command. See Appendix G for examples of these methods.

A COMPLETION STATUS may give a check condition that leads to the possible errors of: Bad Argument, all class 00 errors, unformatted or bad format, and cartridge changed error.

See the examples in Appendix G for methods of generating defect lists and verifying data integrity on the disk.

#### 4.4.7 READ CAPACITY

Now the drive is ready to read and write data. One of many useful commands that is implemented in the Adaptec Controllers is the READ CAPACITY command. This command is a class 1, 10-byte command that returns the total number of logical blocks available. Also, the size of the block, 100, 200, or 400 hex (256, 512 or 1024, decimal) is returned. See Section 5.4.1 for details.

A COMPLETION STATUS may give a check condition that leads to the possible errors of: Bad argument, all class 00 errors, ID ECC error, ID address mark not found, seek error and record not found.

Now the SCSI bus will return read data, giving 4 bytes of the capacity block address (maximum usable logical block) followed by four bytes of the block size.

This command is a good way to test the ability of the controller to read data from the disk.

#### 4.4.8 WRITE

The WRITE command is used to write data from the host to the disk. See Section 5.3.6 for details.

A COMPLETION STATUS may give a check condition that leads to the possible errors of: Bad argument, all class 00 errors, ID ECC error, ID address mark not found, seek error and record not found, plus others as defined in Chapter 5.

The controller now expects 255 blocks of data from the host adapter to follow. These will be written onto the disk, starting at logical block 0 and continuing to 254.



#### 4.4.9 READ

The READ command is used to read data from the drive to the host. See Section 5.3.5 for details.

A COMPLETION STATUS may give a check condition that leads to the possible errors of: Bad Argument, all class 00 errors, ID ECC error, ID address mark not found, seek error and record not found, plus others as defined in Section 5.3.5.

If a Data ECC error occurs during the read, the controller will re-read the block up to four times to establish a solid error correction block (syndrome). Correction may occur after two retries are completed if the error syndrome is repeated twice consecutively. Correction is done directly into the controller's data buffer, transparent to the host.

The host now expects 255 blocks of data from the drive to follow. These will be read from the disk, starting at logical block 0 and continuing to 254.

#### 4.5 EXAMPLES

##### 4.5.1 FIRST TIME FORMAT

1. Rezero and Test for unit ready.
2. Send the MODE SELECT command with the appropriate drive parameters. This example shows the parameters for a 10 MB ST412 type drive.

<u>BYTE</u>	<u>CONTENT</u>	<u>MEANING</u>
00	15	MODE SELECT
01	00	For Drive 0, 20 for Drive 1
02	00	Reserved
03	00	Reserved
04	16	Data Block Length (Soft Sector fixed drive)
05	00	Reserved

<u>BYTE</u>	<u>CONTENT</u>	<u>MEANING</u>
00	00	Reserved
01	00	Reserved
02	00	Reserved
03	08	Length of Extent Descriptor List
04	00	Data Density Code
05	00	Reserved
06	00	Reserved
07	00	Reserved
08	00	Reserved
09	00	High Byte of Block Size
10	01	Middle Byte of Block Size (256 for this example)
11	00	Low Byte of Block Size
12	01	List Format Code ( 01 for fixed disk, soft sector drives)
13	01	High Byte of Reduced Write Current (256 for this example)
14	32	Low Byte of Cylinder Count
15	04	Number of Data Heads
16	01	High Byte of Reduced Write Current (256 for this example)
17	00	Low Byte of Reduced Write Current
18	01	High Byte of Precomp Cylinder (256 for this example)
19	00	Low Byte of Precomp Cylinder
20	00	Landing Zone Position (# cylinders above maximum cylinder)
21	01	Stepping code (01 = 28usec/step, buffered)

3) Send the FORMAT command with no defect list appended.

<u>BYTE</u>	<u>CONTENT</u>	<u>MEANING</u>
00	04	FORMAT
01	00	Indicates that there is no unique data Fill byte and no defect list for drive 0
02	00	Defaults to a '6C' data fill pattern
03	00	High byte of interleave
04	03	Low byte of interleave
05	00	Reserved

After formatting the drive is ready to write and read data. The drive should now be verified for data integrity. This method of formatting without defects is not recommended for normal use and is shown for example only. See the following examples for recommended formatting and data checking methods.

#### 4.5.2 FORMAT WITH MANUFACTURER'S DEFECT LIST

This example demonstrates how to format the ACB-4000 with an appended bad block map as provided by the manufacturer.

1. Rezero and Test for unit ready.
2. Send the MODE SELECT command with the appropriate drive parameters. See example 4.5.1 above.
3. Now use the following FORMAT command:

```
04  FORMAT
1C  For Drive 0, 3C for Drive 1, with a defect map appended
00  Desired data field (default is '6c')
00  High byte of interleave (must be 0)
01  Low byte of interleave
00  RESERVED
```

This FORMAT command with appended defect map for mapping out defects would appear as follows:

```
00  Reserved
00  Reserved
00  High byte of length of defect list (8 X Number of defects)
08  Low byte of length of defect list (0008 in this example)
00  High byte of cyl number
00  Middle byte of cyl number
13  Low byte of cyl number (19d in this example)
07  Head number of defect
00  High byte of bytes from index
00  Second byte of bytes from index
0A  Third byte of bytes from index
0A  Low byte of bytes from index (2570d in this example)
```

The drive will be formatted and is ready to use for writing and reading data.

## 5.0 SCSI COMMAND SET

---

### 5.1 GENERAL DESCRIPTION

This section details the SCSI command set implemented in the ACB-4000A and ACB-4070 controllers. This command set meets all standard commands defined in the SCSI ANSI X3T9.2 for Direct Access Devices. Also, Class 0 optional and vendor unique commands are implemented as well as Class 1 optional and extended commands. The command set is defined in Table 5-1. Sections 5.3 and 5.4 give commands, OP codes, and page number for reference.

By using the optional command MODE SELECT plus vendor unique parameters, Adaptec is able to achieve device independence and drive flexibility. MODE SELECT drive parameters are written to the disk and later retrieved on power-up by auto-configuration without I/O driver intervention (Section 2.0). MODE SENSE allows you to read these drive parameters. Adaptec Vendor Unique commands allow you to exercise functions of the controller that guarantee the highest in data integrity. TRANSLATE gives you the ability to convert logical addresses to physical cylinder, head and bytes from index. This can be used to generate a defect list for the FORMAT command.

The ACB-4000A and ACB-4070 have three error check counters: Uncorrectable Data Check, Correctable Data Check, and Seek Data Check counters, plus there are two usage counters: Read Usage and Seek Usage.

TABLE 5-1. ACB-4000A/4070 SCSI COMMAND SET

6 BYTE COMMANDS

CLASS 0/STANDARD COMMANDS

<u>OP Code</u>	<u>Command</u>
03	Request Sense
04	Format Unit
08	Read
0A	Write

CLASS 0/OPTIONAL COMMANDS

<u>OP Code</u>	<u>Command</u>
00	Test Unit Ready
01	Rezero Unit
0B	Seek
15	Mode Select
1A	Mode Sense
1B	Start/Stop
1C	Receive Diagnostics
1D	Send Diagnostics

CLASS 0/VENDOR UNIQUE COMMANDS

<u>OP Code</u>	<u>Command</u>
0F	Translate
10	Set Error Threshold
11	Read Counters
13	Write Buffer
14	Read Buffer

10 BYTE COMMANDS

CLASS 1/OPTIONAL COMMANDS

<u>OP Code</u>	<u>Command</u>
2E	Write and Verify
2F	Verify
31	Search Data Equal

CLASS 1/EXTENDED COMMANDS

<u>OP Code</u>	<u>Command</u>
25	Read Capacity
28	Read
2A	Write

Adaptec's READ COUNTERS command allows you to read these error check and usage counters. The SET ERROR THRESHOLD gives you the ability to set up drive reliability and performance checkpoints. When the counters reach the Error Threshold a Flag is set that can be used for hard, soft, and seek error rates. Read and Seek usage can be used to determine the efficiency of the I/O drivers and system performance.

The WRITE DATA BUFFER and READ DATA BUFFER commands give you the ability to write data to the controller's buffer and then read it. This can be used for system level diagnostics and checking the SCSI bus data transfer integrity.

SEARCH DATA EQUAL command gives you the ability to search data bit-by-bit for an equal or unequal block of data. When used after format, it can be used to verify disk integrity bit-by-bit rather than just ECC verification of a data field.

This powerful SCSI command set gives both performance and flexibility to your system design.

## 5.2 COMMAND DESCRIPTOR BLOCK (CDB)

An I/O request to a device is made by passing a Command Descriptor Block (CDB) to the controller. The first byte of the CDB is the command class and operation code. The remaining bytes specify the Logical Unit Number (LUN), block starting address, control byte and the number of blocks to transfer. Commands are categorized into two classes supported by Adaptec controllers.

### **NOTE:**

THE SCSI COMMAND DESCRIPTOR BLOCK CONTAINS RESERVED BYTES (FOR FUTURE SCSI DEFINITION). THE ACB-4000A AND ACB-4070 REQUIRE THAT THOSE BYTES BE SET TO ZERO OR THE COMMAND WILL BE REJECTED.

Two classes of commands are supported by the ACB-4000A and ACB-4070. They are Class 0, six-byte commands, and Class 1, 10-byte commands.

Figures 5-1 and 5-2 show typical command descriptor block formats.

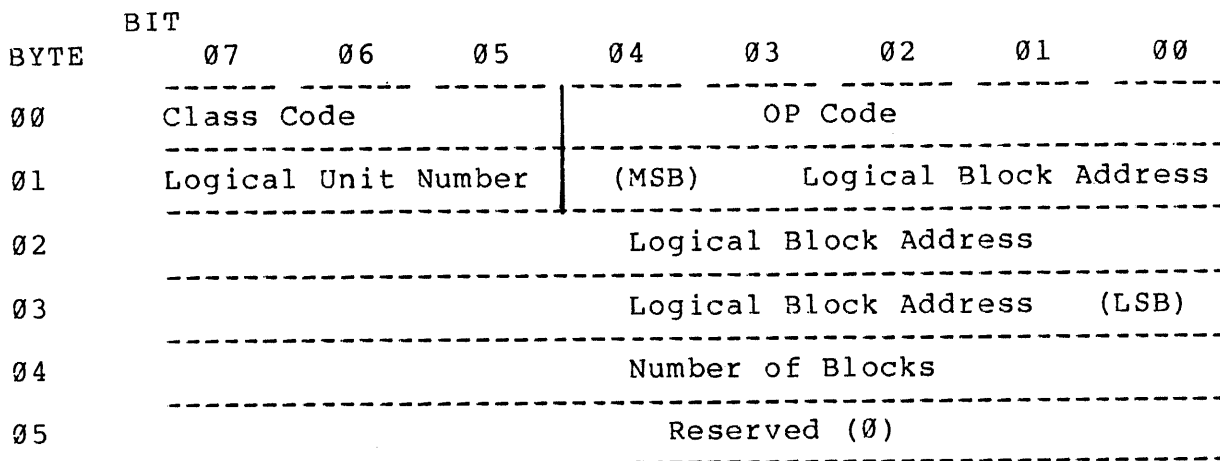


FIGURE 5-1. CLASS 00 COMMANDS (6-BYTE COMMANDS)

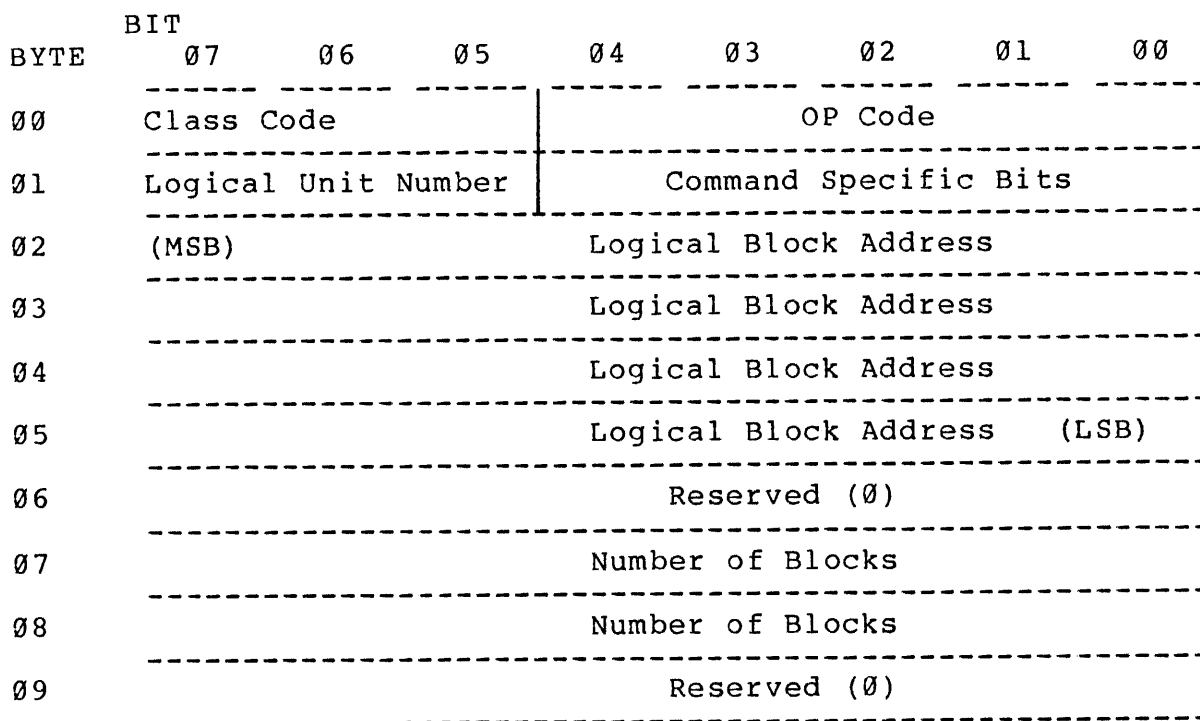


FIGURE 5-2. CLASS 01 COMMANDS (10-BYTE EXTENDED BLOCK ADDRESS)

### 5.2.1 CLASS CODE

The class code can be 0 to 7, but only 0 and 1 are used at this time.

### 5.2.2 OPERATION CODE

The operation code for each class allows 32 commands (00 to 1F hex).

### 5.2.3 LOGICAL UNIT NUMBER

The ACB-4000A and ACB-4070 accommodate two devices per controller which must be devices 0 to 1.

### 5.2.4 COMMAND SPECIFIC BITS

Class 1 Commands byte 01, bits 01-04 specify options which depend upon the particular command.

### 5.2.5 LOGICAL BLOCK ADDRESS

Class 0 commands contain 21-bit starting block addresses while class 1 supports 32-bit block addressing.

### 5.2.6 NUMBER OF BLOCKS

A variable number of blocks may be transferred under a single command. Class 00 commands may transfer up to 256 blocks, while class 01 commands may transfer up to 64K blocks.

#### **NOTE:**

A ZERO BLOCK NUMBER COUNT DEFAULTS TO THE MAXIMUM VALUE.

### 5.2.7 CONTROL BYTE

All bits in the control byte, (byte 5 for Class 0), Byte 9 for Class 1) are reserved and must be zero.

## 5.3 CLASS 00 COMMAND DESCRIPTIONS

The following section describes and details the complete command set for the ACB-4000 Series Controllers.

Table 5-2 shows a series of command descriptions.



TABLE 5-2. CLASS 00 COMMAND CODE SUMMARY

OP CODE	COMMAND	PAGE	OP CODE	COMMAND	PAGE
00	Test Unit Ready	5-7	10	Set Error Threshold	5-19
01	Rezero Unit	5-8	11	Read Counters	5-20
03	Request Sense	5-46	13	Write Data Buffer	5-22
04	Format Unit	5-9	14	Read Data Buffer	5-23
08	Read	5-13	15	Mode Select	5-23
0A	Write	5-14	1A	Mode Sense	5-28
0B	Seek	5-16	1B	Start/Stop Unit	5-29
0F	Translate	5-17	1C	Receive Diagnostic	5-29
			1D	Send Diagnostic	5-31

### 5.3.1 TEST UNIT READY (00 hex)

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	0	0	0	0	0	0	0	0
01	Logical Unit Number			Reserved 0				
02	Reserved (0)							
03	Reserved (0)							
04	Reserved (0)							
05	Reserved (0)							

FIGURE 5-3. TEST UNIT READY COMMAND

This command returns zero status if the requested unit is powered on and the DRIVE READY signal is asserted. If not ready, a check condition will be set in the status byte. For drives that assert DRIVE READY before seeking to track 0, a REZERO command should be done before TEST UNIT READY.

Valid Errors:

<u>Error</u>	<u>Error Code</u>
Drive Not Ready	04
Write Fault	03

5.3.2 REZERO UNIT (01 hex)

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	0	0	0	0	0	0	0	1
01	Logical Unit Number			Reserved (0)				
02	Reserved (0)							
03	Reserved (0)							
04	Reserved (0)							
05	Reserved (0)							

FIGURE 5-4. REZERO UNIT COMMAND

This command sets the heads on the selected drive to track zero and then sends completion status.

Valid Errors:

<u>Error</u>	<u>Error Code</u>
No Seek Complete	02
Drive Not Ready	04
Bad Argument	24

### 5.3.3 REQUEST SENSE (03 hex)

See Section 5.5 for details of the complete command as well as a complete discussion of returned sense data.

### 5.3.4 FORMAT UNIT (04 hex)

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	0	0	0	0	0	1	0	0
01	Logical Unit Number			Data	Cmplt	List Format Bits		
02	Data Pattern							
03	(MSB)			Interleave				
04				Interleave		(LSB)		
05	Reserved (0)							

FIGURE 5-5. FORMAT UNIT COMMAND

The ACB-4000 Series will write from index to index all ID and DATA fields with the format specified by an immediately previous MODE SELECT (15 hex) command. If no MODE SELECT command has been executed, the previous format will be used as read from the drive on power-up. On unformatted disks or those whose format cannot be determined (sense byte error code 1C hex returned following a READ), a MODE SELECT command should be used prior to the format command. If no mode select is used, the controller defaults to parameters for an ST506 5 MB drive. Data fields are completely written with 6C hex unless otherwise specified in the format command.

The ACB-4000 Series formats out all indicated disk defects during disk formatting.

Byte 01 is used to indicate if a list of defect locations is appended and whether unique fill characters are to be used.

Bits 0, 2, 3, and 4 indicate the presence and format of the defect list. The ACB-4000 Series only supports bytes from index format. Bit 1 indicates whether a unique fill character is to be written into the data fields during format. Table 5-3 details the format options provided by the FORMAT UNIT command.

TABLE 5-3. FORMAT OPTIONS

<u>Data</u>	<u>Bit</u>		<u>Format</u>		<u>Defect List</u>	<u>Fill Byte</u>
	<u>Cmplt</u>					
4	3	2	1	0		
0	0	0	0	0	No defect list	6C hex
0	0	0	1	0	No defect list	Value in Byte 02
1	1	1	0	0	Complete defect list in bytes from index format	6C hex
1	1	1	1	0	Complete defect list in bytes from index format	Value in Byte 02

Sector interleaving may be required because of performance limitations in the host. The sector interleave number is equivalent to the number of disk revolutions required to read or write a full track of data.

The ID fields will be interleaved as specified in byte 04 of the CDB. The controller does not require interleaving because of a high speed buffer controller. An interleave number of 1 results in sequential ID fields being written on the disk. Any interleave number between 1 and the number of sectors per track results in interleaved formatting. A 0 in this field will cause the default interleave factor of two to be used. Byte 3 must always be zero. The value in byte 4 must not exceed the number of sectors per track minus one. An error code of 1A hex (Interleave Error) is returned if byte 4 is too large.

An example of an interleave number of 3 with 32 sectors per track follows:

```
P - 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16
F - 00 22 11 01 23 12 02 24 13 03 25 14 04 26 15 05 27

- 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
- 16 06 28 17 07 29 18 08 30 19 09 31 20 10 32 21
```

P = Physical sector count  
F = Formatted logical sector locations

If invalid data bytes are noted by the controller while reading the defect list, all formatting is stopped and a Bad Argument (24 hex) is returned to the host.

Invalid Data Bytes consist of:

- 0 Defect on cylinder 0.
- 0 Defect list is not in ascending order.
- 0 Byte displacement is too large.

Valid Errors:

<u>Error</u>	<u>Error Code</u>
All Class 0 Errors	00-05 hex
Unformatted or Bad Format	1C hex
Interleave Error	1A hex
Bad Argument	24 hex
Write Protect	27 hex
Cartridge Changed	28 hex

The following is the format for the appended defect list. This list includes the physical coordinates of known media flaws in cylinder, head, and bytes from index.

**NOTE:**

ALL DEFECTS MUST BE LISTED IN ASCENDING ORDER.

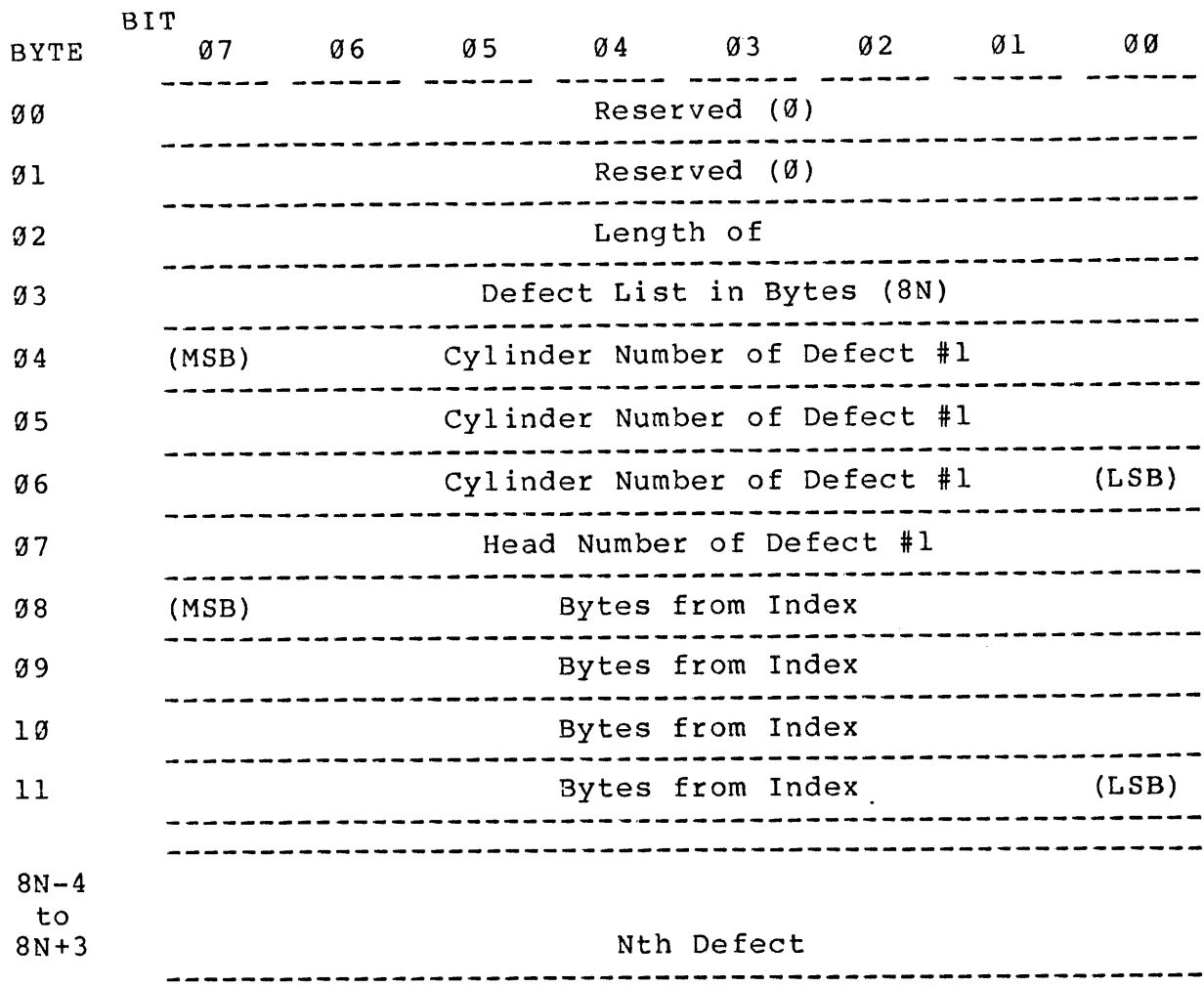


FIGURE 5-6. DEFECT DATA BLOCK

5.3.5 READ (08 hex)

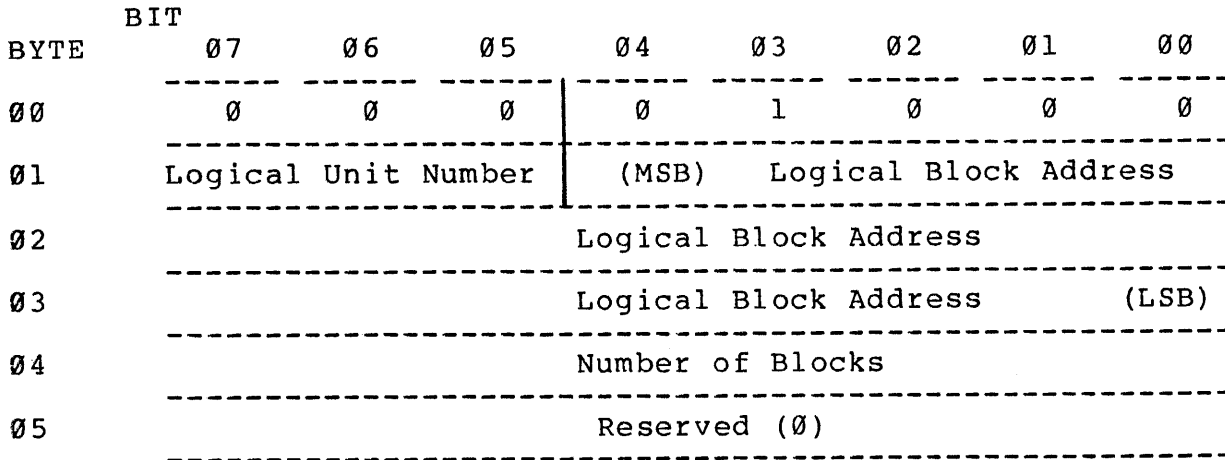


FIGURE 5-7. READ COMMAND

This command transfers (to the host) the specified number of blocks starting at the specified block address.

The control unit will verify a valid seek address and proceed to seek to the specified starting logical block address. When the seek is complete the controller then reads the starting address data into the buffer, checks ECC and begins DMA data transfer.

Subsequent blocks of data are transferred into the buffer in a similar manner until the block count is decremented to zero. Cylinder switching is transparent to the user. On a data ECC error, the block is reread up to four times to establish a solid error syndrome. Correction may occur after two retries are completed if the error syndrome is repeated twice consecutively. Correction is done directly into the data buffer transparent to the host.

Blocks containing uncorrectable data errors will be transferred to the host prior to an ending check status. A REQUEST sense will return an uncorrectable data error (11 hex).

See the SEND DIAGNOSTICS Command for ECC and retry options.



Valid Errors:

<u>Error</u>	<u>Error Code</u>
All Class 0 Errors	00-05* hex
I.D. CRC Error	10* hex
Uncorrectable Data Error	11* hex
I.D. AM Not Found	12* hex
Record Not Found	14* hex
Seek Error	15* hex
Data Check (No Retry Mode)	18* hex
Bad Format	1C hex
Illegal Block Address	21 hex
Volume Overflow	23 hex
Bad Argument	24 hex
Cartridge Changed	28 hex

\*Address will be valid in sense data.

This set of errors is collectively referred to as Read Operation Errors.

5.3.6 WRITE (0A hex)

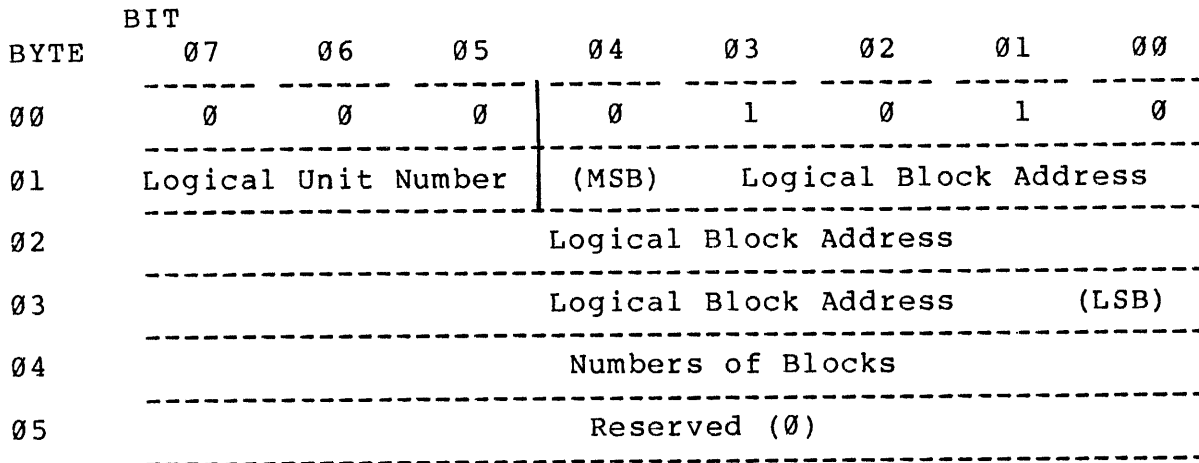


FIGURE 5-8. WRITE COMMAND

This command transfers (to the target device) the specified number of blocks beginning at the specified logical starting block address. The controller seeks to the specified logical starting block. When the seek is complete, the controller transfers the first block into its buffer and writes its buffered data and its associated ECC into the first logical sector.

Subsequent blocks of data are transferred as available from the FIFO buffer until the block count is decremented to zero. Cylinder switching and defect skipping are transparent to the user.

The ACB-4000A and ACB-4070 also support corresponding extended READ and WRITE commands using the class 01 CDB format.

Valid Errors:

<u>Error</u>	<u>Error Code</u>
All Class 0 Errors	00-05* hex
I.D. CRC Error	10* hex
I.D. AM Not Found	12* hex
Record Not Found	14* hex
Seek Error	15* hex
Bad Format	1C hex
Illegal Block Address	21 hex
Volume Overflow	23 hex
Bad Argument	24 hex
Cartridge Changed	28 hex

\*Address will be valid in sense data.

This set of errors is collectively referred to as Write Operation Errors.

### 5.3.7 SEEK (0B hex)

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	0	0	0	0	1	0	1	0
01	Logical Unit Number			(MSB)	Logical Block Address			
02	Logical Block Address							
03	Logical Block Address (LSB)							
04	Reserved (0)							
05	Reserved (0)							

FIGURE 5-9. SEEK COMMAND

This command causes the selected drive to seek to the specified starting address. The ACB-4000A and ACB-4070 returns completion status immediately after the seek pulses are issued and head motion starts, allowing it to free the bus and accept further commands prior to actual seek completion.

**NOTE:**

ANY COMMAND RECEIVED FOR A UNIT WITH A SEEK IN PROGRESS WILL IMMEDIATELY COMPLETE WITH A COMMAND COMPLETION STATUS OF BUSY (BIT 3 SET). THIS IS DONE TO ALLOW THE HOST TO USE THE SCSI BUS TO DO OTHER PROCESSING WHILE WAITING FOR SEEK COMPLETE.

The drive is stepped to the addressed track position but no ID field verification is attempted.

All ACB-4000A and ACB-4070 use an implied seek on READ, WRITE and SEARCH DATA EQUAL commands. This eliminates the need for the issuance of SEEK commands with each operation.

Valid Errors:

<u>Error</u>	<u>Error Code</u>
No Seek Complete	02 hex
Drive Not Ready	04 hex
Bad Format	1C hex
Illegal Block Address	21 hex
Bad Argument	24 hex
Invalid Logic Number	25 hex

### 5.3.8 TRANSLATE (0F hex)

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	0	0	0	0	1	1	1	1
01	Logical Unit Number			(MSB)	Logical Block Address			
02	Logical Block Address							
03	Logical Block Address							(LSB)
04	Reserved (0)							
05	Reserved (0)							

FIGURE 5-10. TRANSLATE COMMAND

This command performs a logical address to physical address translation and returns the physical location of the requested block address in a cylinder, head, bytes from index format. This data can be used to build a defect list for the FORMAT command.

Eight bytes are returned in the format of defect descriptors required by FORMAT.

If there is a data error in the ID field, an error status will be returned. It is then necessary to TRANSLATE the blocks before and after the targeted block to determine the location of the target block. The use of interleaved sectors and formatted (skipped) defects may complicate the determination of the error location. See Appendix G for examples using TRANSLATE. When generating a defect list, it is advisable to perform the TRANSLATE command with non-interleaved operation. This is done in order to simplify logical block calculations.

BYTE	BIT								
	07	06	05	04	03	02	01	00	
00	(MSB)			Cylinder Number					
01				Cylinder Number					
02				Cylinder Number				(LSB)	
03				Head Number					
04	(MSB)			Bytes from Index					
05				Bytes from Index					
06				Bytes from Index					
07				Bytes from Index				(LSB)	

FIGURE 5-11. TRANSLATE DATA

Valid Errors:

<u>Error</u>	<u>Error Code</u>
All Class 0 Errors	00-05 hex
I.D. CRC Error	10 hex
I.D. AM Not Found	12 hex
Record Not Found	14 hex
Seek Error	15 hex
Bad Format	1C hex
Illegal Block Address	21 hex
Bad Argument	24 hex
Cartridge Changed	28 hex

5.3.9 SET ERROR THRESHOLD (10 hex)

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	0	0	0	1	0	0	0	0
01	Logical Unit Number				Reserved (0)			
02	Reserved (0)							
03	Reserved (0)							
04	Bytes to be Transferred (01)							
05	Reserved (0)							

**FIGURE 5-12. SET THRESHOLD COMMAND**

The ACB-4000 Series optionally provides an error logging capability for those errors that are normally retried without any notification to the host system. The controller counts blocks transferred, seek mode and the frequency of error presentation is established by the SET THRESHOLD command, while the actual error information is presented by the READ/RESET USAGE COUNTERS command. The default state is error logging, but not reported. Power on reset establishes the default state.

One byte of parameter data will be transferred.

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	Threshold Value							

A threshold value of 0 specifies that no error reporting will take place. A value between 1 and 255 will request that error reporting takes place. When the number of errors of any single type exceeds the threshold, the command that finds that error is completed normally. The subsequent commands will be terminated immediately with Check Condition. Sense status will indicate 2C hex, Error Count Overflow. When a READ/RESET USAGE COUNTER command is executed, the usage and error counters are off loaded and normal operation continues. The same threshold value remains in effect.

Valid Errors:

<u>Error</u>	<u>Error Code</u>
Drive Not Ready	04 hex
Bad Argument	24 hex
Cartridge Changed	28 hex
Counter Overflow	2C hex

5.3.10 READ/RESET USAGE COUNTER (11 hex)

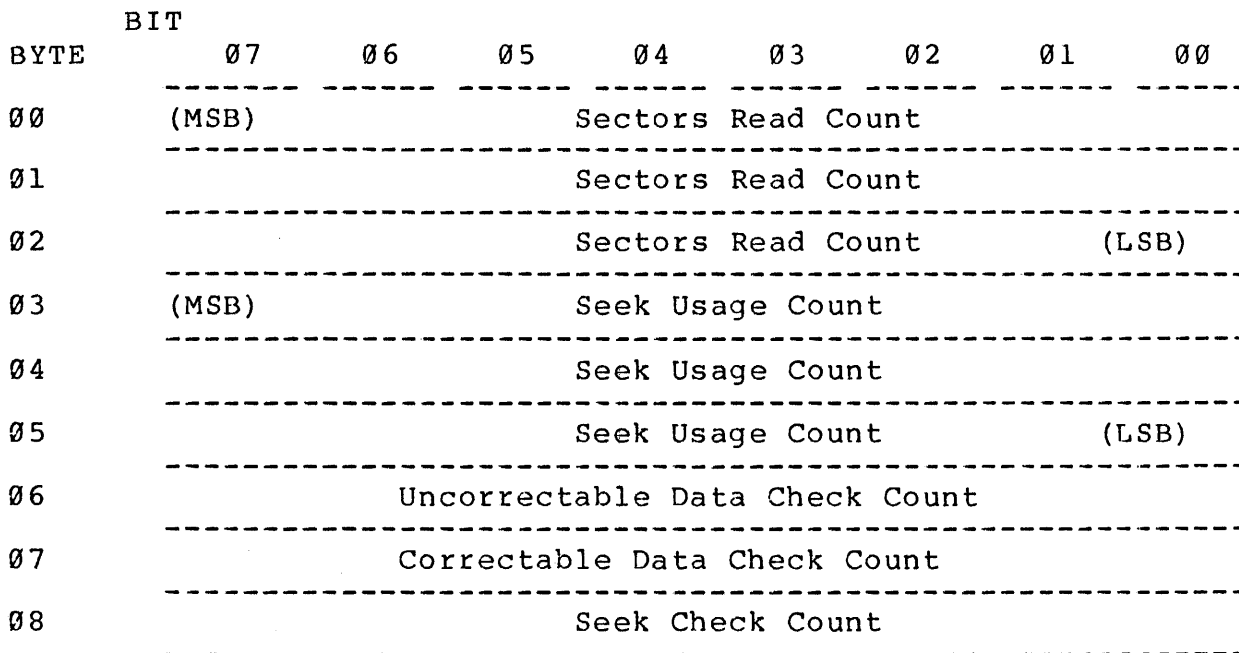
BYTE	BIT	07	06	05	04	03	02	01	00
00		0	0	0	1	0	0	0	0
01		Logical Unit Number			Reserved (0)				
02		Reserved (0)							
03		Reserved (0)							
04		Bytes Allocated				(09)			
05		Reserved (0)							

**FIGURE 5-13. READ/RESET USAGE COUNTER COMMAND**

The READ/RESET USAGE COUNTERS command recovers the information stored by the ACB-4000 Series for the particular disk device. The information is valuable to observe the statistical performance of the device and to point to devices which may need service before their performance degrades system operation.

All seeks and sectors read are counted in three-byte counters. Seek errors, correctable data errors, and uncorrectable data errors are counted in one-byte counters.

When one of the error counters exceeds the threshold, all subsequent commands for that device will terminate immediately with Check Condition status and an error code of 2C hex, Error Counter Overflow. This will continue until execution of the READ/RESET USAGE COUNTER command, which recovers the nine bytes of counter information and resets the counters.



**FIGURE 5-14. READ/RESET USAGE COUNTER PARAMETERS**

The sectors Read Count is a complete count of all logical blocks read to any host from the specified drive. This provides usage information against which error counts can be calibrated.

The Seek Usage Count is a complete count of all occurrences of an initial seek by the drive. Cylinder switching is not counted.

The Uncorrectable Data Check Count counts all occurrences of an uncorrectable data check on the specified device. Each uncorrectable data check was also posted as an 11-hex error code.

The Correctable Data Check Count counts all occurrences of the successful recovery of a logical block that was unsuccessfully read at first. This information is available only through error logging, since these errors are recovered without notifying the host unless a diagnostic mode has been invoked.

The Seek Check Count counts all occurrences of a seek error whether or not recovery was successful. This information is available only from error logging, since seek errors are normally recovered without notifying the host.

**Valid Errors:**

<u>Error</u>	<u>Error Code</u>
Drive Not Ready	04 hex
Bad Argument	24 hex
Cartridge Changed	28 hex
Counter Overflow	2C hex



### 5.3.11 WRITE DATA BUFFER (13 hex)

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	0	0	0	1	0	0	1	1
01	Logical Unit Number			Reserved (0)				
02	Reserved (0)							
03	Reserved (0)							
04	Reserved (0)							
05	Reserved (0)							

FIGURE 5-15. WRITE DATA BUFFER COMMAND

This command serves buffer RAM diagnostic purposes. The controller will fill the buffer with 1K bytes of data from the host.

#### Valid Errors:

<u>Error</u>	<u>Error Code</u>
Drive Not Ready	04 hex
Bad Argument	24 hex
Cartridge Changed	28 hex
Counter Overflow	2C hex

### 5.3.12 READ DATA BUFFER (14 hex)

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	0	0	0	1	0	1	0	0
01	Logical Unit Number			Reserved (0)				
02	Reserved (0)							
03	Reserved (0)							
04	Reserved (0)							
05	Reserved (0)							

FIGURE 5-16. READ DATA BUFFER COMMAND

READ DATA BUFFER will pass the host 1K of data from the buffer. It is intended for RAM diagnostic purposes. In addition, although data remains in the buffer after normal data operations, the ordering of the data found there may vary.

Valid Error:

<u>Error</u>	<u>Error Code</u>
Drive Not Ready	04 hex
Bad Argument	24 hex
Cartridge Changed	28 hex
Counter Overflow	2C hex

### 5.3.13 MODE SELECT (15 hex)

This command is used in the ACB-4000 Series Controllers to specify format parameters and should always precede the FORMAT command. When a blown format error (code 1C) is detected due to the controller being unable to read the drive information from a drive already formatted, the user should use this command to inform the controller about the drive information. Once initialized, most data on the drive will be recoverable. The information can then be recovered and the drive reformatted, and writes to the drive will not be permitted.

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	0	0	0	1	0	1	0	1
01	Logical Unit Number			Reserved (0)				
02	Reserved (0)							
03	Reserved (0)							
04	Number of Bytes							
05	Reserved (0)							

FIGURE 5-17. MODE SELECT COMMAND

Byte 04 of the command specifies the number of information bytes to be passed with the command. A minimum of 12 bytes (0C hex) must be specified. If drive parameters are being specified, the count should be 22 bytes (16 hex) for soft-sectored drives and 24 bytes (18 hex) for hard-sectored and removable drives.

The parameter list is four bytes long with the first three bytes reserved (zeros). The fourth byte contains the length in bytes of the extent descriptor list; this is always eight. (Only a single extent is supported.)

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	Reserved (0)							
01	Reserved (0)							
02	Reserved (0)							
03	Length of Extent Descriptor List = 08 hex							

FIGURE 5-18. MODE SELECT PARAMETER LIST

Byte 0 of the extent descriptor list specifies the data density of the drive, and a value of 00 in this byte is required. Bytes 1, 2, 3, and 4 are reserved and must be zero, specifying that the entire drive is to be formatted. Bytes 5 through 7 are used to specify the data block size. The block size must not be less than 256 or exceed the RAM buffer capacity which is 1024 characters. For a hard sector drive, this must be equal to the sector size specified by the drive manufacturer.

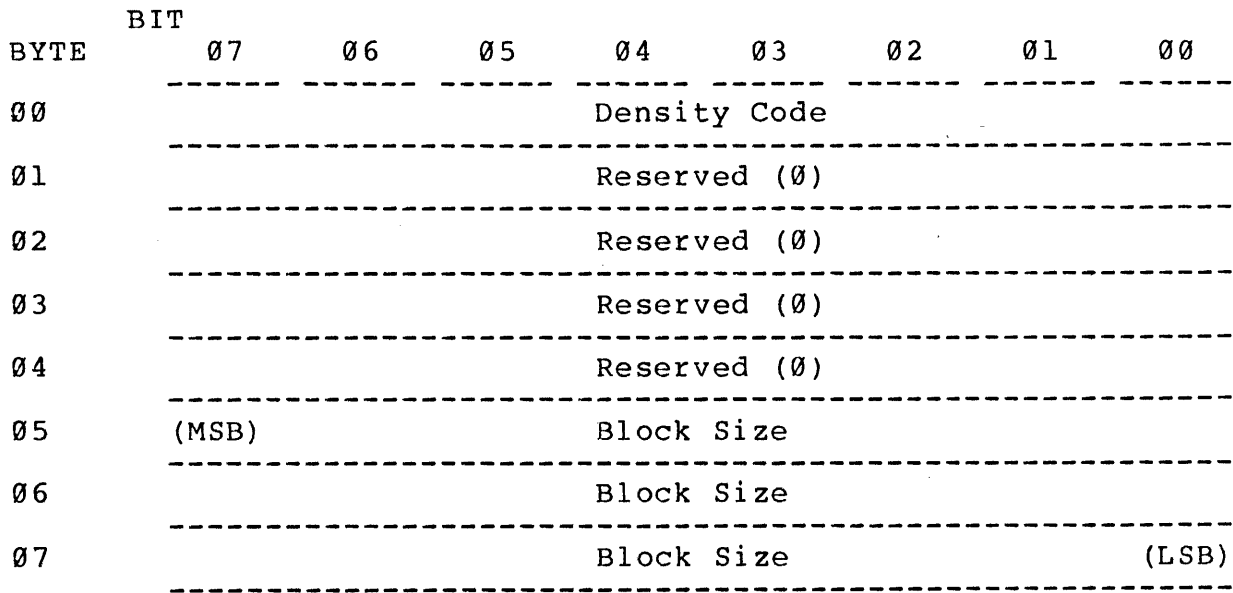


FIGURE 5-19. EXTENT DESCRIPTOR LIST

The extent descriptor list and following drive parameter list are a single large data block which follows the command. The ACB-4000A and ACB-4070 must be set-up with a block size value 256, 512 or 1024 bytes in hexadecimal.

Any violation of the above constraints will result in Check Status with an error code of 24H, indicating an invalid argument in parameter data.

BYTE	BIT	07	06	05	04	03	02	01	00
00		List Format Code							
01	MSB	Cylinder Count							
02		Cylinder Count							LSB
03		Data Head Count							
04	MSB	Reduced Write Current Cylinder							
05		Reduced Write Current Cylinder							LSB
06	MSB	Write Precompensation Cylinder							
07		Write Precompensation Cylinder							LSB
08		Landing Zone Position							
09		Step Pulse Output Rate Code							
0A		0	0	0	0	F/R	H/R	0	0
0B		Sectors Per Track							

----- Hard-Sectored and Removable Drives  
 \_\_\_\_\_ Soft-Sectored Fixed Drives

FIGURE 5-20. DRIVE PARAMETER LIST

The drive parameter list includes all the data necessary to specify a drive. It is optional, but if present must be completed and the items must be within the limits stated. If these parameters are not supplied, the format operation will use previously supplied values if available or the default values given below.

The list format code must be 01 for the soft-sectored fixed drives and must be 02 for hard-sectored and removable drives.

The cylinder count is the number of data cylinders on the drive. Due to the sector level defect skipping, formatting cylinders normally set aside as spares may be included in this total. The minimum is sixteen. The maximum supported is 2048. The default value is 306.

The data head count is the number of usable data surfaces. The heads will be selected from 0 to head count minus one. The minimum is one; maximum is 16 decimal. A drive with nine or more heads will use the reduced write current line as the high order head select. The default value is two.

The reduced write current cylinder is the cylinder number beyond which the controller will assert the reduced write current line. Minimum value is 0; maximum is 2047. The default value is cylinder 128. Note that reduced write current assumes a different meaning on drives with more than eight heads.

The write precompensation cylinder is the cylinder beyond which the controller will compensate for inner track bit shift. The specifications for this function agree with those of most disk manufacturers, that is, 12 nanoseconds. Minimum value is 0; maximum is 2047.

**NOTE:**

ON THE ACB-4000A AND ACB-4070 THIS FIELD IS IGNORED. THE PRECOMP THRESHOLD IS THE SAME AS THE REDUCED WRITE CURRENT VALUE. AS MOST DRIVES NOW IGNORE THE REDUCED WRITE CURRENT SIGNAL, THIS IS NOT A SERIOUS RESTRICTION. HOWEVER, JUMPERS ARE PROVIDED ON THE BOARD WHICH ALLOW THE PRECOMPENSATION TO BE SELECTED AS ALWAYS ON, ALWAYS OFF, OR TIED TO REDUCED WRITE CURRENT. THE NORMAL POSITION IS TIED TO REDUCED WRITE CURRENT. THIS JUMPER APPLIES TO BOTH DRIVES. SEE TABLE 3-1.

The ACB-4070 does not require reduced write current or write precompensation.

The landing zone position is used with the START/STOP command to indicate the direction and number of cylinders from the last (or first) data cylinder to the shipping position. The most significant bit indicates the direction with a zero meaning that the landing zone is beyond the highest innermost track, and a one indicates the landing zone is outside track zero. The low seven bits gives the number of cylinders. The default is zero (land on inner most track).

The step pulse output rate code specifies the timing of seek steps. Three options are currently available:

- 00 = Non-Buffered Seek - 3.0 mS rate - ST506
- 01 = Buffered Seek - 28 uS rate - ST412
- 02 = Buffered Seek - 12 uS rate

Bytes 0A and 0B of the drive parameter list are only required with hard-sectored or removable drives. In byte 0A, bit 2 = 0, soft-sector drive; bit 2 = 1, hard-sector drive; bit 3 = 0, removable media; bit 3 = 1, fixed media. The number of sectors per track specified in byte 0B must be same as specified by hard-sector drive manufacturer.

### 5.3.14 MODE SENSE (1A hex)

This command is used to interrogate the ACB-4000A and ACB-4070 device parameter table to determine the specific characteristics of any disk drive currently attached. The attached drive must have been formatted by an ACB-4000A or ACB-4070 for this to be a legal command.

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	0	0	0	1	1	0	1	0
01	Logical Unit Number			Reserved (0)				
02	Reserved (0)							
03	Reserved (0)							
04	Number of Bytes Returned							
05	Reserved (0)							

FIGURE 5-21. MODE SENSE COMMAND

Byte 04 of the command specifies the number of data bytes to be returned from the command. A minimum of 12 bytes (0C hex) must be specified. If the drive parameter list is required, the count should be 22 bytes (16 hex) for soft-sectored fixed drives and 24 bytes (18 hex) for the hard-sectored and removable drives.

The return information will be the four-byte parameter list, the extent descriptor list and the drive parameter list (if requested). These lists take the exact format of those in the MODE SELECT command. Please reference that command for exact detail.

Valid Errors:

<u>Error</u>	<u>Error Code</u>
Bad Argument	24 hex
Cartridge Changed	28 hex
Counter Overflow	2C hex

### 5.3.15 START/STOP UNIT (1B hex)

Byte 04, bit 00 of this command should be set to 01 if this is a START command, and 00 for a STOP command.

This command is designed for use on drives with a designated shipping or landing zone.

A STOP command will position the head to the landing zone position. See the MODE SELECT command for description of the landing zone value.

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	0	0	0	1	1	0	1	1
01	Logical Unit Number			Reserved (0)				
02	Reserved (0)							
03	Reserved (0)							
04	Reserved (0)							ST/STP
05	Reserved (0)							

FIGURE 5-22. START/STOP UNIT COMMAND

Valid Errors:

<u>Error</u>	<u>Error Code</u>
Drive Not Ready	04 hex
Bad Argument	24 hex
Cartridge Changed	28 hex
Counter Overflow	2C hex

### 5.3.16 RECEIVE DIAGNOSTIC (1C hex)

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	0	0	0	1	1	1	0	0
01	Logical Unit Number			Reserved (0)				
02	Reserved (0)							
03	(MSB)			Data Length				
04	Data Length					(LSB)		
05	Reserved (0)							

FIGURE 5-23. RECEIVE DIAGNOSTIC COMMAND



This command sends analysis data to the host after completion of a SEND DIAGNOSTIC command. Bytes 3 and 4 designate the size of the available buffer (in bytes).

RECEIVE DIAGNOSTIC is used to transfer data to the host and must immediately follow a SEND DIAGNOSTIC command which initiates the dump action. Otherwise, the command will be rejected.

The data length specified should be 104 hex or more. If a smaller buffer is provided, only that much data will be transferred and the command will terminate normally.

The data buffer received as a result of a dump will be formatted as shown in Figure 5-24.

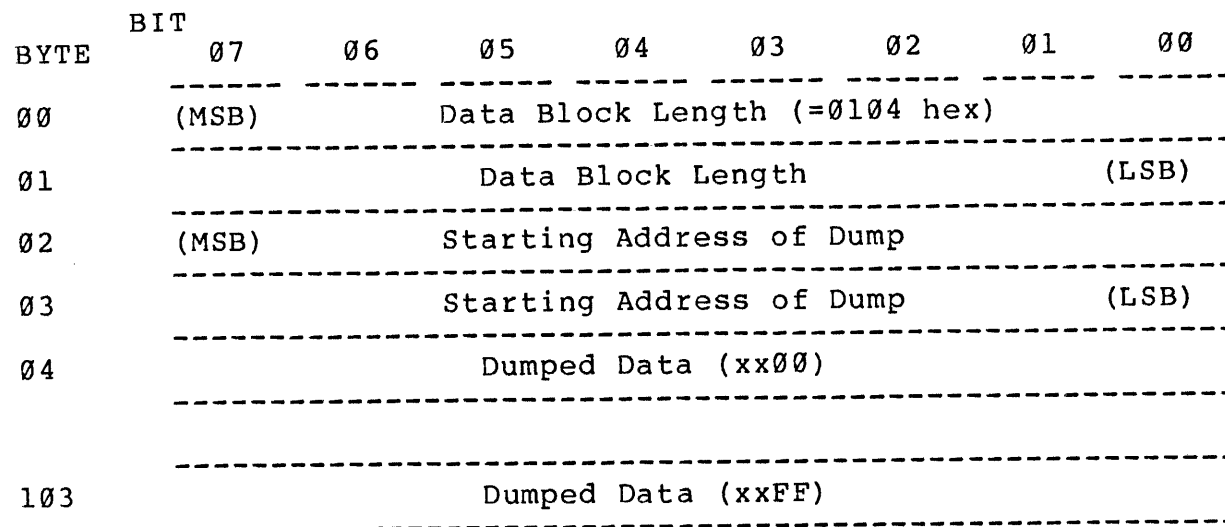


FIGURE 5-24. RECEIVE DIAGNOSTIC DATA

Valid Errors:

<u>Error</u>	<u>Error Code</u>
Bad Argument	24 hex
Cartridge Changed	28 hex
Counter Overflow	2C hex

5.3.17 SEND DIAGNOSTIC (1D hex)

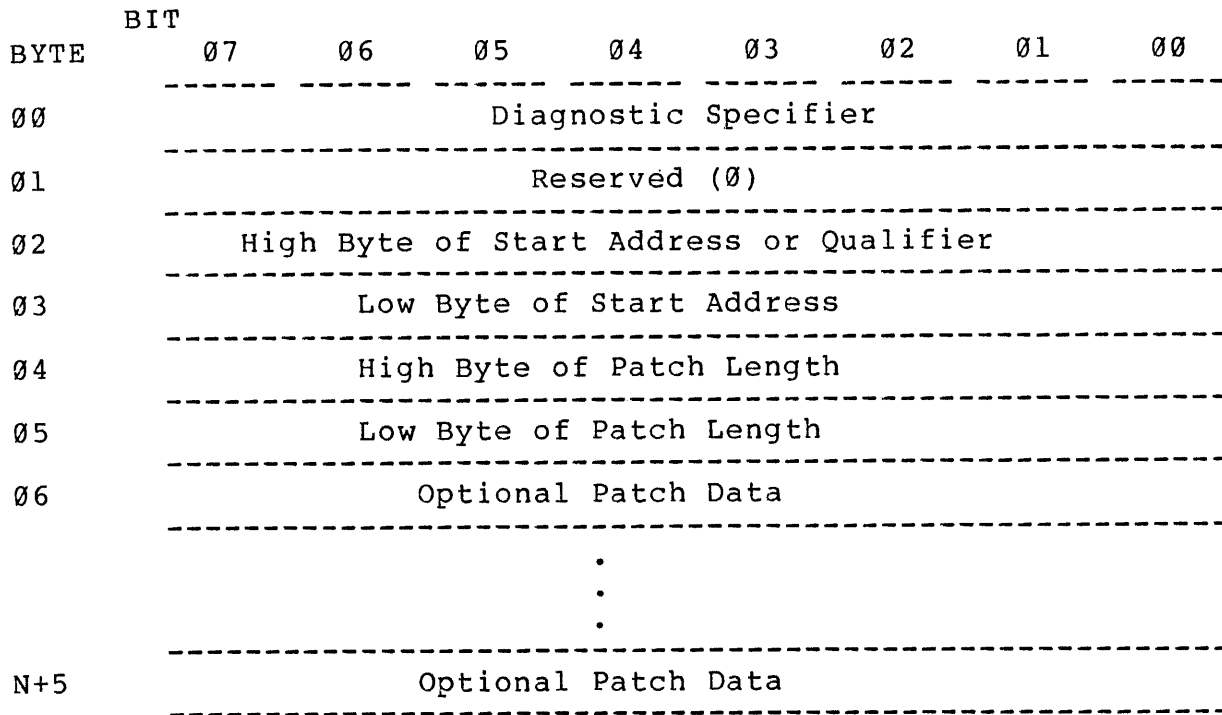
BYTE	BIT							
	07	06	05	04	03	02	01	00
00	0	0	0	1	1	1	0	1
01	Logical Unit Number			Reserved (0)				
02	Reserved (0)							
03	(MSB)			Data Length				
04				Data Length (LSB)				
05	Reserved (0)							

**FIGURE 5-25. SEND DIAGNOSTIC COMMAND**

This command sends data to the Controller to specify the execution of diagnostic functions tests for Controller and peripheral units.

Bytes 3 and 4 specify the length of the data to be sent.

The data length specified in the command must be at least four bytes long and should be equal to the length of the data block to be passed over to the controller. If the length specified is longer than needed, the excess is ignored and not transferred.



**FIGURE 5-26. SEND DIAGNOSTIC PARAMETER FORMAT**

Byte 00 of the Parameter List specifies the particular diagnostic function being requested. The following options are presently available.

- 60 -- Reinitialize Drive
- 61 -- Dump Hardware Area (4000-40FF)\*
- 62 -- Dump RAM (8000-80FF, C000-C0FF)
- 63 -- Patch Hardware Area\*
- 64 -- Patch RAM\*
- 65 -- Set Read Error Handling Options

\*Standard diagnostics do not require these functions.

The detailed format for each of these options is described below.

DIAGNOSTIC 60 -- REINITIALIZE DRIVE

The selected drive runs through its initialization procedure, rezeroing, reading the drive characteristic parameters, determining the maximum capacity, and capturing defect skipping parameters. The parameter list is:

<u>Byte</u>	<u>Contents</u>
0	60 hex
1	30 hex
2-5	00 hex

No RECEIVE DIAGNOSTIC information is available as a result of this diagnostic option.

DIAGNOSTIC 61 -- DUMP HARDWARE AREA

The area specified by the dump address is transferred by the RECEIVE DIAGNOSTIC command immediately following this SEND DIAGNOSTIC command. The parameter list is:

<u>Byte</u>	<u>Contents</u>
0	61 hex
1	00 hex
2	40 hex
3	Low Order Address of Hardware Area.
4-5	Length of Transfer to be Performed.

DIAGNOSTIC 62 -- DUMP RAM AREA

The area specified by the dump address is transferred by the RECEIVE DIAGNOSTIC command immediately following this SEND DIAGNOSTIC command. The parameter list is:

<u>Byte</u>	<u>Contents</u>
0	62 hex
1	00 hex
2	80 hex or E0 hex
3	Low Order Address of RAM area.
4-5	Length of Transfer to be Performed.

DIAGNOSTIC 63 -- PATCH HARDWARE AREA

This option is for Adaptec use only.

DIAGNOSTIC 64 -- PATCH RAM AREA

These commands will provide special diagnostic tools for analysis of certain very complex system interactions. No use should be made of these commands without contacting an Adaptec applications engineer, since temporary unavailability or loss of critical data may occur.

DIAGNOSTIC 65 - SET READ ERROR HANDLING OPTIONS

The selected drive is set in the special error recovery mode established by the contents of byte 02. The error handling mode is set to the default value by a hard SCSI reset condition, a power on reset, and by a SEND DIAGNOSTIC command specifying the default error handling value.

<u>Byte</u>	<u>Contents</u>
0	65 hex
1	00 hex
2	Error Handling Option
3	00 hex

The Error Handling options are specified below:

00 hex (Default Value) - ECC and Retries Enabled

A correctable error will be corrected and any data transfer will be completed. No check status will be presented. If the error is not correctable, the controller will transfer the uncorrected data and post an error code of 11 hex with the address valid bit set. The address will be the logical block address of the bad block.

01 hex - Disable ECC and Retries

If an ECC error occurs on the first read of a data field, the data transfer operation will be halted after transfer of the bad data block. A check condition will be presented. The error code will be 18 with the address valid bit set. The failing block address will be in the logical block address field of the sense information.

02 hex - ECC and Four Retries Enabled Plus Stop on Correction

A correctable error will be corrected and the corrected data transferred. The operation will then stop and present check status and an error code of 18 hex as described in option 01. An uncorrectable error will be handled as in option 00.

See Appendix G for an example of using these options.

#### 5.4 CLASS 1 COMMAND DESCRIPTION

The Class 1 commands supported by the ACB-4000 Series Controllers is shown in Table 5-4.

TABLE 5-4. CLASS 01 COMMAND CODE SUMMARY

OP CODE	COMMAND	PAGE
25	Read Capacity	5-36
28	Read	5-37
2A	Write	5-38
2E	Write and Verify	5-39
2F	Verify	5-40
31	Search Data Equal	5-41

### 5.4.1 READ CAPACITY (25 hex)

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	0	0	1	0	0	1	0	1
01	Logical Unit Number			Reserved (0)				
02	(MSB)			Logical Block Address				
03				Logical Block Address				
04				Logical Block Address				
05				Logical Block Address (LSB)				
06				Reserved (0)				
07				Reserved (0)				
08				Full or Partial Media Indicator				
09				Reserved (0)				

FIGURE 5-27. READ CAPACITY COMMAND

If the Partial Media Indicator (PMI) is 00 hex, this command will return the address of the last block on the unit. It is not necessary to specify a starting block address in this command mode. If PMI is 01 hex, this command will return the address of the block (after the specified starting address) at which a substantial delay of time in data transfer will be encountered (e.g., a cylinder boundary). Any value other than 00 hex or 01 hex in byte 08 will cause Check Status with an error code of 24 hex for an invalid argument.

In both cases, an eight-byte data field is returned. The first four bytes are defined as the block address and the last four bytes are the block size.

#### Valid Errors:

<u>Error</u>	<u>Error Code</u>
All Class 0 Errors	00-05 hex
I.D. ECC Error	10 hex
I.D. AM Not Found	12 hex
Record Not Found	14 hex
Seek Error	15 hex
Bad Argument	24 hex
Cartridge Changed	28 hex
Counter Overflow	2C hex

5.4.2 READ (28 hex)

This command is an extended address command which is identical to the class 00 READ (08 hex) command. The large Logical Block Address and Number of Blocks fields are provided for accessing very large devices. A maximum of 65K blocks can be transferred.

Valid Errors: (See Section 5.3.5)

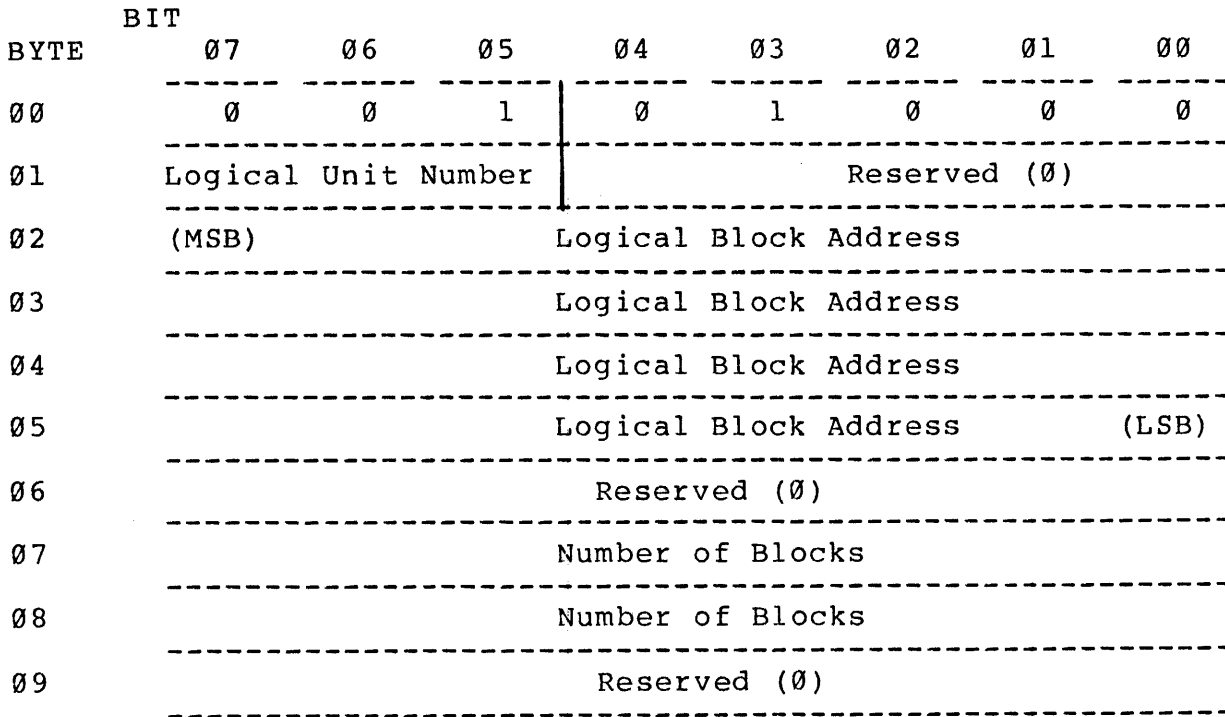


FIGURE 5-28. READ COMMAND



### 5.4.3 WRITE (2A hex)

This command is an extended address command identical to the class 00 WRITE (0A hex) command. The Logical Block Address and Number of Blocks fields have been expanded for larger devices. A maximum of 65K blocks can be transferred.

Valid Errors: (See Section 5.3.6)

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	0	0	1	0	1	0	1	0
01	Logical Unit Number			Reserved (0)				
02	(MSB)			Logical Block Address				
03				Logical Block Address				
04				Logical Block Address				
05				Logical Block Address (LSB)				
06				Reserved (0)				
07				Number of Blocks				
08				Number of Blocks				
09				Reserved (0)				

FIGURE 5-29. WRITE COMMAND

#### 5.4.4 WRITE AND VERIFY (2E hex)

This command is similar to the traditional "read after write" function. It is an extended address command which operates like a WRITE command over the specified number of blocks and then verifies the data written on a block by block basis. The verify function transfers no data to the host and only checks the ECC to be correct.

Since no data is transferred to the host during verify, correctable data checks will be treated in the same manner as uncorrectable data checks.

Valid Errors:

<u>Error</u>	<u>Error Code</u>
Read Operation Errors	
Write Operation Errors	
ECC Error During Verify	19 hex

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	0	0	1	0	1	1	1	0
01	Logical Unit Number			Reserved (0)				
02	(MSB)			Logical Block Address				
03				Logical Block Address				
04				Logical Block Address				
05				Logical Block Address (LSB)				
06				Reserved (0)				
07				Number of Blocks				
08				Number of Blocks				
09				Reserved (0)				

FIGURE 5-30. WRITE AND VERIFY COMMAND

### 5.4.5 VERIFY (2F hex)

This command is similar to the previous WRITE AND VERIFY except that it verifies the ECC of an already existing set of data blocks. No Write Operation is performed. It is up to the host to provide data for rewriting and correcting if an ECC error is detected.

Valid Errors:

<u>Error</u>	<u>Error Code</u>
Read Operation Errors	
Write Operation Errors	
ECC Error During Verify	19 hex

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	0	0	1	0	1	1	1	1
01	Logical Unit Number			Reserved (0)				
02	(MSB)			Logical Block Address				
03				Logical Block Address				
04				Logical Block Address				
05				Logical Block Address (LSB)				
06				Reserved (0)				
07				Number of Blocks				
08				Number of Blocks				
09				Reserved (0)				

FIGURE 5-31. VERIFY COMMAND

#### 5.4.6 SEARCH DATA EQUAL (31 hex)

This powerful extended address command provides for a search and compare on equal of any data on the disk. A starting block address and number of blocks to search are specified and a search argument is passed from the host which includes a byte displacement (not supported) and the data to compare.

The invert bit (Byte 01, Bit 04) inverts the sense of the search comparison operation. This SEARCH DATA NOT EQUAL will cause the controller to stop on a sector not equal to the search data. It will report search satisfied with a status of equal (04 status).

This command allows the host to perform a high-speed data verify. Unlike VERIFY command which only checks for ECC errors, the search data equal will compare a chosen data pattern against data contained in selected blocks on the fly. This feature provides an excellent method of verifying disk integrity after format by searching not equal for a 6C hex or other unique fill character.

When a search is satisfied, it will terminate with an Equal status. A REQUEST SENSE command can then be issued to determine the block address of the matching record. A REQUEST SENSE command following a successful SEARCH DATA command will:

1. Report a Completion Status Byte of Equal if the search was satisfied by an exact match. If the search was satisfied by an inequality, a Sense Key of No Sense (00 hex) is reported.
2. Set the Sense Bytes address valid bit to one.
3. Report the address of the block containing the first matching record in the Sense Bytes.

A REQUEST SENSE command following an unsuccessful SEARCH DATA command will:

1. Report a Completion Status Byte of No Sense (00 hex), provided no errors occurred.
2. Set the Sense Bytes address valid bit to zero.

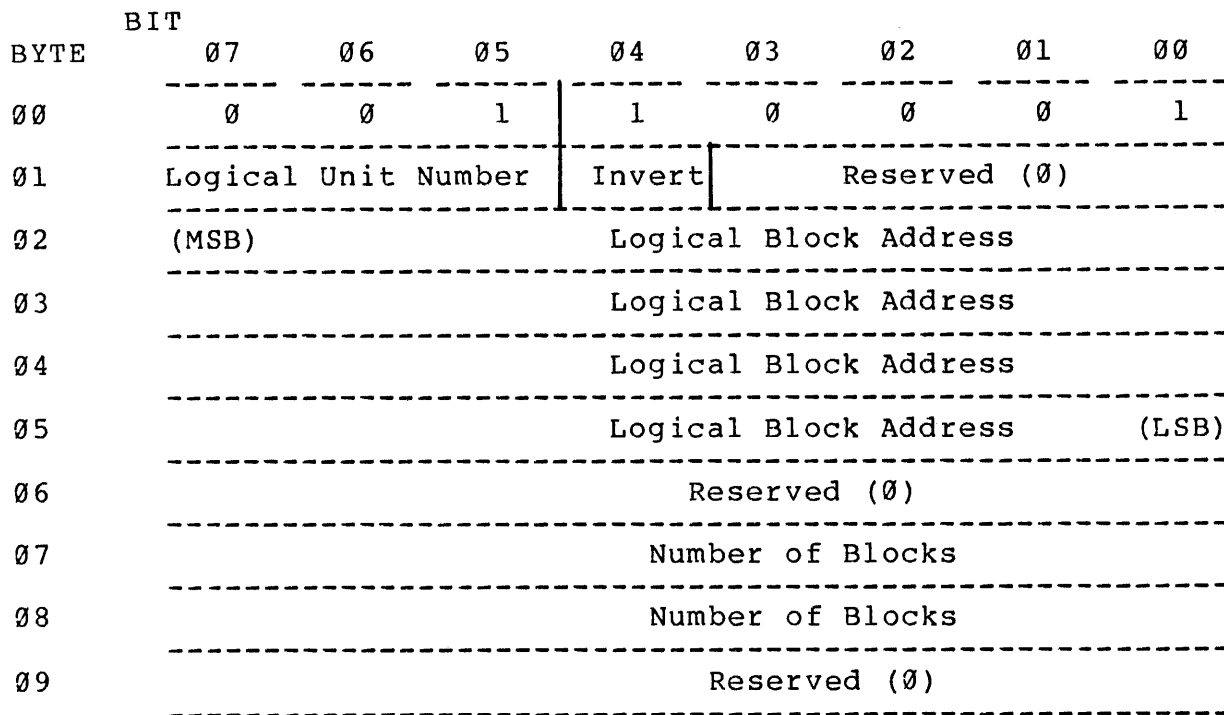


FIGURE 5-32. SEARCH DATA EQUAL COMMAND

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	(MSB)							
01								
02								
03								
04	(MSB)							
05								
06								
07								
08	(MSB)							
09								
10								
11								
12	(MSB)							
13								
14	(MSB)							
15								
16								
17								
18	(MSB)							
19								
20								
M+19								

FIGURE 5-33. SEARCH DATA EQUAL ARGUMENT

A definition of the required data in the SEARCH argument is shown in Table 5-5.

TABLE 5-5. SEARCH DATA EQUAL ARGUMENT

BYTES	PARAMETER
00 to 03	Record Size (Bytes) For the ACB-4000 Series this must equal the block size or zero. Zero will be taken to mean the format block size.
04 to 07	First Record Offset (Bytes) For the ACB-4000 Series this must be zero.
08 to 11	Number of Records For the ACB-4000 Series this must be less than or equal to the number of blocks specified in the command and greater than zero. The search will terminate upon a match or when the smaller of these values is encountered
12 to 13	Search Argument Length (Bytes) The number of bytes in the following search argument. Must equal the pattern length +6.
14 to 17	Search Field Displacement The displacement from the beginning of the record to the first byte to be compared. Must be zero for the ACB-4000 Series controllers.
18 to 19	Pattern Length (M Bytes) The number of bytes in the following data pattern to be compared with a like size in each record. Pattern length must equal block size on the 4000 Series controllers.
20 to M+19	Data Pattern The block of data to be compared.

## 5.5 COMPLETION STATUS BYTE

Status is always sent at the end of a command. Intermediate status is sent at the completion of a linked command. Any abnormal condition encountered during command execution causes command termination and ending status.

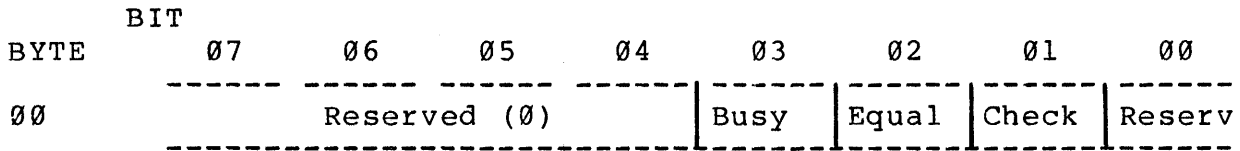


FIGURE 5-34. COMPLETION STATUS BYTE

Bits 0, 4, 5,6 and 7: Always zero

Bit 1: Check condition. Sense is available. See REQUEST SENSE below.

Bit 2: Equal. Set when any Search is satisfied.

Bit 3: Busy. Device is busy or reserved. Busy status will be sent whenever a target is unable to accept a command from a host. This condition occurs when a LUN is seeking or has outstanding sense data.



### 5.5.1 REQUEST SENSE (03 hex)

This command returns unit sense information.

The sense data will be valid for the Check status condition sent to the host and will be saved by the controller until requested. Sense data will be cleared on receiving a subsequent command from the host that received the check condition. Therefore, Check status should always be followed by a SENSE command.

The number-of-blocks field (byte 04) specifies the number of bytes allocated by the host for returned sense. Values of 0 to three bytes will default to four bytes. The controller returns four bytes of sense information in response to this command. Check status will not be sent in response to this command.

BYTE	BIT							
	07	06	05	04	03	02	01	00
00	0	0	0	0	0	0	1	1
01	Logical Unit Number				Reserved (0)			
02	Reserved (0)							
03	Reserved (0)							
04	Number of Bytes							
05	Reserved (0)							

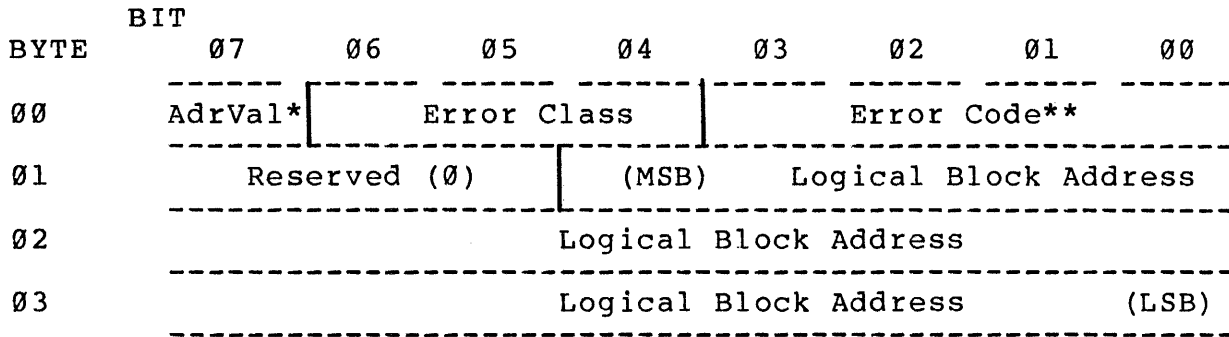
FIGURE 5-35. REQUEST SENSE COMMAND

Valid Errors:

Error  
Bad Argument

Error Code  
24 hex

### 5.5.2 SENSE BYTES



\*The address valid bit (byte 00, bit 07) indicates that the Logical Block Address bytes contain valid information.

\*\*See Section 6.

FIGURE 5-36. REQUEST SENSE DATA

The bit definitions, error codes and probable cause and described in Section 6.4 of Section 6: Troubleshooting.

## 6.0 TROUBLESHOOTING

---

### 6.1 INTRODUCTION

This section describes the procedures needed to troubleshoot problems that may arise when installing the Adaptec ACB-4000 Series Controller boards. These are the most commonly found problems and are not inclusive of every application.

**NOTE:**

WHEN TROUBLESHOOTING PROBLEMS, USE THE MOST BASIC SYSTEM CONFIGURATION. THAT IS, ONE HARD DISK DRIVE ON THE ACB-4000A AND ALL OTHER CONTROLLERS SUCH AS TAPE, ETC., REMOVED. ONCE THE SYSTEM WORKS FOR THE BASIC CONFIGURATION, ADD DRIVES AND CONTROLLERS TO THE SYSTEM ONE AT A TIME AND RETEST AFTER EACH ADDITION.

If these procedures fail to give a solution to your problem, recheck your steps, read the entire manual, document the problem, and check with the technical support department where you bought the controller.

ADAPTEC ACB-4000A/4070 TROUBLESHOOTING CHECKLIST

- CHECK JUMPERS ON THE DISK DRIVE, BE SURE THAT IT IS NOT SET FOR A RADIAL SELECTED DRIVE.
- CHECK JUMPERS ON CONTROLLER, BE SURE THAT JUMPERS AND PULSE OR HANDSHAKING SELECTION (SEE APPENDIX D) HAVE BEEN DONE PROPERLY.
- CHECK CABLES, BE SURE THAT J0 GOES TO DRIVE 0, J1 GOES TO DRIVE 1 AND J2 GOES TO BOTH DRIVES. BE SURE THAT PIN 1 ON THE CONTROLLER IS CONNECTED TO PIN 1 OF THE DRIVE. BE SURE THAT THE SCSI CABLE IS CONNECTED PROPERLY, NOTING THE LOCATION OF PIN 1.
- CHECK THAT THE DRIVE PARAMETERS AND STEP PULSE RATE ARE IN AGREEMENT WITH THE DRIVE VENDOR.
- CHECK THAT THE TERMINATORS ON THE DRIVE AND SCSI BUS ARE SET PROPERLY.
- CHECK THAT THE POWER SUPPLY CAN SUPPORT THE ADDED CURRENT REQUIRED BY THE DRIVE. BE SURE THAT THE +5V AND +12V VOLTAGES ARE CORRECT. VERIFY WITH THE DRIVE VENDOR HIS REQUIREMENTS.
- CHECK THE DRIVE'S WRITE PRE-COMPENSATION AND REDUCED WRITE CURRENT VALUES IF SOFT READ ERRORS OCCUR. ALSO IF THESE ARE NOT BEING USED BE SURE THAT THE CORRECT JUMPER IS INSTALLED. THERE MUST BE A JUMPER FROM "R" TO ANOTHER PIN.
- CHECK TO SEE THAT YOU ARE MEETING THE SCSI BUS SIGNAL PARAMETERS SUCH AS SETUP, HOLD , MINIMUM AND MAXIMUM TIMES.

## 6.2 SELF DIAGNOSTICS

### **NOTE:**

THESE SELF-DIAGNOSTICS ARE FOR TROUBLESHOOTING ON A SYSTEM LEVEL ONLY. THEY ARE NOT INTENDED FOR REPAIR OF THE BOARD.

The ACB-4000 Series Controller has built-in diagnostics. These are performed at power-on time when the O-P jumper is installed. DO NOT connect the cables. If the board is functioning properly the LED will flash continuously with duration of 0.5 - 1.0 seconds per flash. If there is a problem with the controller, the LED will stay on for six seconds, flash once for one second (notating the start of diagnostics), and then flash in 0.5 second bursts. This will then be repeated as long as the O-P jumper is installed. The number of 0.5 second bursts is the error code. These error codes are detailed in Table 6-1.

**TABLE 6-1. ACB-4000 SERIES CONTROLLER SELF-DIAGNOSTICS**

<u>ERROR CODE</u>	<u>PROBABLE PROBLEM AREA</u>
(NUMBER OF 0.5 SECOND BURSTS)	
NONE	8085 SUBSYSTEM
1	8156 RAM
2	FIRMWARE
3	AIC-010 AND RELATED LOGIC
4	AIC-010 AND RELATED LOGIC
5	AIC-300 AND RELATED LOGIC
6	AIC-010 BUS
CONTINUOUS BURSTS OF 1 SEC, 1 SEC, 1/2 SEC (WITHOUT 6 SECOND FLASH AND 1 SECOND FLASH)	CONTROLLER PASSES SELF- DIAGNOSTICS

When a known good drive is correctly connected to the controller, it will seek and read. Error codes are invalid in this mode.

### 6.3 ERROR CODES AND PROBABLE CAUSES

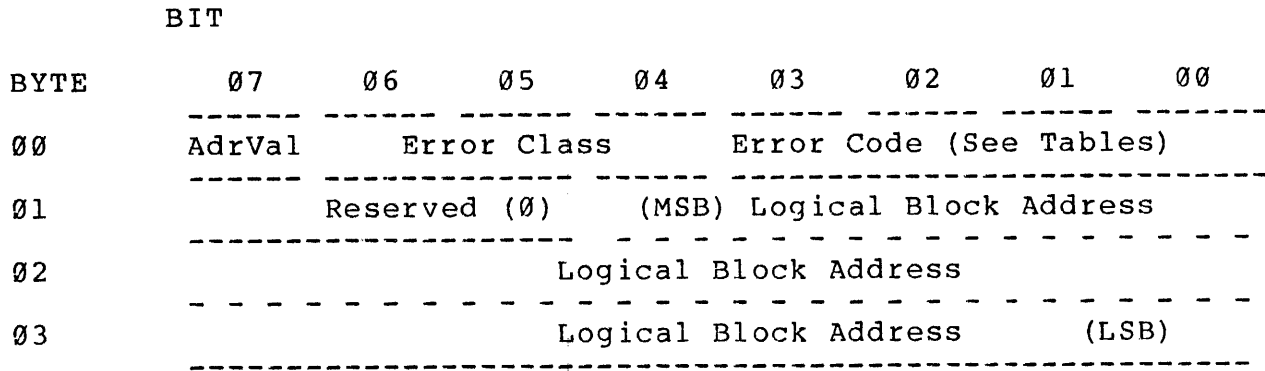
BYTE	BIT							
	07	06	05	04	03	02	01	00
00	0	0	0	0	0	0	1	1
01	Logical Unit Number				Reserved (0)		-	-
02	-	-	-	Reserved (0)		-	-	
03	-	-	-	Reserved (0)		-	-	
04	Allocation Length							
05	Reserved (0)							

FIGURE 6-1. REQUEST SENSE COMMAND

The Request Sense (03 hex) command returns unit sense information. This information is used to troubleshoot invalid operation of the controller, drive or system. The sense data will be valid for the CHECK status condition sent to the Host and will be saved by the controller until requested. Sense data will be cleared on receiving a subsequent command to the LUN related to the check condition from the Host that received the check condition. Other hosts will receive BUSY status to commands for LUN with non-zero sense to report. Therefore, CHECK status should always be followed by a SENSE Command.

The controller returns four bytes of sense information in response to this command. The number of bytes should equal 04, however, values of 00, 01, 02, and 03 will default to 04.

The REQUEST SENSE command is the most important mechanism for informing the host of abnormal states discovered by the controller.



**FIGURE 6-2. REQUEST SENSE DATA**

The AdrVal (Address Valid) bit indicates that the Information Bytes contain a valid logical block address for which the error condition was recorded.

The error class indicates the general type of error detected. Class 0 errors are related to drive state, including ready, seek complete, write fault and similar errors. Class 1 errors are related to data recovery problems. Class 2 errors are related to invalid requests from the host system.

The error code defines precisely the failure that was detected. These codes are described in Table 6-2.

The logical block address is 21 bits long. It contains the address of the logical block for which the failure was detected. If the AdrVal bit is off, the logical block address is not meaningful. A few sense error codes store other information in the logical block address without turning on the AdrVal bit.

TABLE 6-2. CLASS 00 ERROR CODES IN SENSE BYTE (DRIVE ERRORS)

CODE	ERROR	MEANING
00	NO SENSE	No error occurred or error cleared before REQUEST SENSE command.
01	NO INDEX/ SECTOR	No index or sector signal found during rd, wr, or format.
02	NO SEEK COMPLETE	Seek complete signal missing
03	WRITE FAULT	Drive detected failure which disallows writes. Write protect is detected during a write command.
04	DRIVE NOT READY	Drive not ready.
06	NO TRACK 0	Track Zero not found.



**TABLE 6-3. CLASS 01 ERROR CODES IN SENSE BYTE (TARGET ERRORS)**

CODE	ERROR	MEANING
10	ID CRC ERROR	ID Field could not be recovered by retry.
11	UNCORRECTABLE DATA ERROR	Data field error could not be recovered by retry or correction.
12	ID ADDRESS MARK NOT FOUND	Missing ID address mark.
14	RECORD NOT FOUND	Logical block ID not on accessed tracks, but no ID CRC error.
15	SEEK ERROR	Could not seek to track with correct ID.
16-17	NOT ASSIGNED	
18	DATA CHECK IN NO RETRY MODE	See Send Diagnostic Command.

**TABLE 6-4. CLASS 02 ERROR CODES (SYSTEM-RELATED ERRORS)**

CODE	ERROR	MEANING
19	ECC ERROR DURING VERIFY	See Verify command.
1A	INTERLEAVE ERROR	Interleave variable is greater than the number of sectors per track on disk.
1B	NOT ASSIGNED	
1C	UNFORMATTED OR BAD FORMAT ON DRIVE	Format failed, no valid format on drive
1D-1F	NOT ASSIGNED	

TABLE 6-4. CLASS 02 ERROR CODES (SYSTEM-RELATED ERRORS)  
(Continued)

CODE	ERROR	MEANING
20	ILLEGAL COMMAND	Command code is invalid or not implemented.
21	ILLEGAL BLOCK ADDRESS	Block address outside address space by Logical Unit.
22	NOT ASSIGNED	
23	VOLUME OVERFLOW	Illegal block address after first block.
24	BAD ARGUMENT	Reserved bit not zero or invalid parameter.
25	INVALID LOGICAL UNIT NUMBER	Logical Unit greater than 1 addressed.
26	NOT ASSIGNED	
28	CARTRIDGE CHANGED	A disk drive cartridge was installed since the last time a command was executed.
2C	ERROR COUNT OVERFLOW	Posted when error count exceeds specified threshold.

## APPENDIX A. ACB-4000 SERIES SCSI IMPLEMENTATION

---

### A.1 INTRODUCTION

This section describes in detail the SCSI protocol with the extensions which are implemented in the Adaptec ACB-4000A and ACB-4070 controllers.

The extended functions of the SCSI standard not supported by the ACB-4000A and 4070 are command chaining, disconnection and reconnection, arbitration and Parity. When designing systems for the ACB-4000 series, you may go directly from the "bus free" phase to the "selection" phase in the protocol description. See appropriate Hardware and Software Implementation Sections in the Users Manual.

### A.2 GENERAL DESCRIPTION OF SCSI

The SCSI interface provides an efficient method of communication between computers and peripheral I/O devices. The eight-port, daisy-chained I/O bus defined by this specification supports the following features:

- Single or multiple\* host system.
- Multiple peripheral device types.
- Bus contention resolution through arbitration on a prioritized basis.\*
  - Asynchronous data transfer at up to 1.5 MBytes/sec.
  - Host-to-host communication.

\* Supported on ACB-5500.

Communication on the bus is allowed between two bus ports at a time. A maximum of eight bus ports are allowed. Each port is attached to a device (e.g., controller or host adapter).

When two devices communicate with each other on the bus, one acts on an INITIATOR, and the other acts as a TARGET. The INITIATOR (typically a host adapter) begins the operation. The TARGET (typically a controller) executes the operation. A device will usually have a fixed role as an INITIATOR or TARGET, but some devices may be able to assume either role. The basic functions of INITIATOR and TARGET are shown in Table A-1.

**TABLE A-1. BASIC FUNCTIONS**

<u>INITIATOR</u>	<u>TARGET</u>
Arbitrates for the bus. Selects a target.	Requests the transfer of data, command or status.

### A.3 BUS SIGNALS

The nine control signals and eight data signals are described in Table A-2.

TABLE A-2. BUS SIGNALS

<u>TITLE</u>	<u>ASSERTED BY</u>	<u>COMMENTS</u>
Busy (BSY)	Any device	"Or-tied" signal which indicates that the bus is in use.
Select (SEL)	INITIATOR	"Or-tied" signal for device selection (and asserted by TARGET in disconnect/reconnect reselection).
Control/Data (C/D)	TARGET	Signal which indicates whether Control or Data information is on data bus. Assertion (low active) indicates control.
Input/Output (I/O)	TARGET	Signal which indicates direction of data on data bus relative to INITIATOR. Assertion (low active) indicates Input to the INITIATOR.
Message (MSG)	TARGET	Signal indicates message phase.
Request (REQ)	TARGET	Signal indicates a request for a data transfer REQ/ACK handshake.

TABLE A-2. BUS SIGNALS (CONT.)

<u>TITLE</u>	<u>ASSERTED BY</u>	<u>COMMENTS</u>
Acknowledge (ACK)	INITIATOR	Signal indicates an acknowledgement of a data transfer REQ/ACK handshake.
Attention (ATN)	INITIATOR	Signal indicates the ATTENTION condition. INITIATOR has message to send target.
Reset (RST)	Any Device	"Or-tied" signal indicating a RESET condition. Clears SCSI bus of all activities.
Data Bus (DB7-0)	INITIATOR OR TARGET	Signals used for data, control and ID information.

DATA BUS (DB:7-0)

Eight data bit signals comprise the DATA BUS. DB(7) is the most significant bit and has the highest priority during arbitration. (ACB-5500 only). Significance and priority decrease with decreasing bit number.

Each of the eight data signals, DB(7) through DB(0), is uniquely assigned as a TARGET or INITIATOR bus address (i.e., DEVICE ID) which is normally assigned and "strapped" in the device during system configuration. In order to obtain the bus during arbitration, a device asserts its assigned data bit (DEVICE ID) and leaves the other data bits in the passive (non-driven) state.

#### A.4 BUS PHASES

The SCSI bus implemented by the ACB-4000 Series has six distinct operational phases and cannot be in more than one phase simultaneously. See Table A-3.

TABLE A-3. BUS PHASES

<u>BUS PHASE</u>	<u>ASSERTED BY</u>
Bus Free	Initiator
Selection	Initiator
Command	Target
Data	Target
Status	Target
Message	Target

##### A.4.1 BUS FREE PHASE

The BUS FREE Phase, indicating that the bus is available for use, is invoked by the deassertion and passive release of all bus signals. All active devices must deassert and passively release all bus signals (within a BUS CLEAR DELAY) after deassertion of BSY and SEL.

Devices sense BUS FREE when both SEL and BSY are not asserted (simultaneously within a DESKEW DELAY) and the RESET condition is not active.

##### A.4.2 SELECTION PHASE

The SELECTION phase allows an INITIATOR to select a TARGET. The INITIATOR waits a minimum of BUS SETTLE DELAY (after detecting BUS FREE) before driving the DATA bus with the TARGET ID and (optionally) its own ID. After two DESKEW DELAYS, the INITIATOR can assert SEL.

On detecting the simultaneous condition (within one DESKEW DELAY) of SEL, its own ID asserted, and BSY and I/O not asserted, the selected TARGET examines the DATA bus for the INITIATOR ID and responds by asserting BSY.

After a minimum of two DESKEW DELAYS (following the detection of BSY from the TARGET), the INITIATOR deasserts SEL and may change the DATA signals.

The INITIATOR may "time out" the SELECTION phase by deasserting the ID bits on the bus. If (after a SELECTION RESPONSE TIME plus two DESKEW DELAYS) BSY has not been asserted, SEL may be deasserted. The TARGET must drive BSY within a SELECTION RESPONSE TIME of detecting SEL and its own ID.

#### A.4.3 INFORMATION TRANSFER PHASES

The COMMAND, DATA, STATUS, and MESSAGE phases are all used to transfer data or control information through the DATA bus. See Chapter 5 for COMMAND, DATA and STATUS description.

The C/D, I/O and MSG signals are used to differentiate the various INFORMATION TRANSFER phases.

**NOTE:**

THESE SIGNALS ARE NOT VALID WITHOUT REQ ASSERTED. SEE TABLE A-4.

TABLE A-4. INFORMATION TRANSFER PHASE

MSG	SIGNAL C/D	I/O	PHASE NAME	DIRECTION OF INFORMATION TRANSFER
0	0	0	DATA OUT	INIT TO TARG
0	0	1	DATA IN	TARG TO INIT
0	1	0	COMMAND	INIT TO TARG
0	1	1	STATUS	TARG TO INIT
1	0	0	NOT USED	
1	0	1	NOT USED	
1	1	0	MSG OUT	INIT TO TARG
1	1	1	MSG IN	TARG TO INIT

NOTES:

- 0 = Signal Deassertion (High active)
- 1 = Signal Assertion (Low active)
- INIT = INITIATOR
- TARG = TARGET

The INFORMATION TRANSFER phases use the REQ/ACK handshake to control data transfer. Each REQ/ACK allows the transfer of one byte of data. The handshake starts with the TARGET asserting the REQ signal. The INITIATOR responds by asserting the ACK signal. The TARGET then deasserts the REQ signal and the INITIATOR responds by deasserting the ACK signal.

With I/O signal asserted, data will be input to the INITIATOR from the TARGET. The TARGET must insure that valid data is available on the bus (at the INITIATOR port) before the assertion of REQ at the INITIATOR port. The data remains valid until the assertion of ACK by the INITIATOR. The TARGET should compensate for cable skew and the skew of its own drivers.

With the I/O signal not asserted, data will be output from the INITIATOR to the TARGET. The INITIATOR must insure valid data on the bus (at the TARGET port) before the assertion of ACK on the bus. The INITIATOR should compensate for cable skew and the skew of its own drivers. Valid data remains on the bus until the TARGET deasserts REQ.



During each INFORMATION TRANSFER phase, the BSY line remains asserted, the SEL line remains deasserted and the TARGET will continuously envelop the REQ/ACK handshake(s) with the C/D, I/O and MSG signals in such a manner that these control signals are valid for BUS SETTLE DELAY before the REQ of the first handshake and remain valid until the deassertion of ACK at the end of the last handshake.

#### A.4.3.1 COMMAND PHASE

The command phase allows the TARGET to obtain command information from the INITIATOR.

The TARGET asserts the C/D signal and deasserts the I/O and MSG signals during the REQ/ACK handshake(s) of this phase.

#### A.4.3.2 DATA PHASE

The DATA phase includes both the DATA IN phase and the DATA OUT phase.

The DATA IN phase allows the TARGET to INPUT data to the INITIATOR. The TARGET asserts the I/O signal and deasserts the C/D and MSG signals during the REQ/ACK handshake(s) of this phase.

The DATA OUT phase allows the TARGET to obtain OUTPUT data from the INITIATOR. The TARGET deasserts the C/D, I/O and MSG signals during the REQ/ACK handshake(s) of this phase.

#### A.4.3.3 STATUS PHASE

The STATUS phase allows the TARGET to send status information to the INITIATOR.

The TARGET asserts C/D and I/O and it deasserts the MSG signal during the REQ/ACK handshake(s) of this phase.

#### A.4.3.4 MESSAGE PHASE

The MESSAGE phase includes the MESSAGE IN and MESSAGE OUT phases.

The MESSAGE IN phase allows the TARGET to INPUT a message to the INITIATOR. The TARGET asserts C/D, I/O and MSG during the REQ/ACK handshake(s) of this phase.

The MESSAGE OUT phase allows the TARGET to obtain a message from the INITIATOR. The TARGET may invoke this phase only in response to the ATTENTION condition created by the INITIATOR. In response to the ATTENTION condition, the TARGET asserts C/D and MSG and deasserts the I/O signal during the REQ/ACK handshake(s) of this phase.

#### A.4.4 SIGNAL RESTRICTIONS BETWEEN PHASES

When the BUS is between phases, the following restrictions apply to the bus signals:

- The BSY, SEL, REQ and ACK signals may not change.
- The C/D, I/O, MSG and DATA signals may change.
- The ATN and RST signals may change as defined under the description for the ATTENTION and RESET conditions.

## A.5 MESSAGE SYSTEM

The message system allows communication between an INITIATOR and TARGET for purposes of physical path management. This section defines the messages and lists their assigned codes (in HEX).

Normally, the first message sent by the INITIATOR after the SELECTION phase is IDENTIFY (to establish the physical path). After reselection, the TARGET's first message is also IDENTIFY. Under certain conditions, an INITIATOR may send SELECTIVE RESET or BUS DEVICE RESET as the first message.

The ACB-4000A and 4070 controllers only support the COMMAND COMPLETE message and do not respond to the ATN signal except during selection. Only COMMAND COMPLETE need be implemented in an ACB-4000 series environment.

### A.5.1 COMMAND COMPLETE (00H)

Command complete is a single byte message. This code is sent from the TARGET to the completion of command execution to direct the INITIATOR to indicate COMMAND COMPLETE to the host.

**NOTE:**

THIS MESSAGE DOES NOT IMPLY GOOD ENDING STATUS; STATUS MUST BE CHECKED TO DETERMINE END CONDITIONS.

## A.6 BUS CONDITIONS

The bus has two asynchronous conditions: the ATTENTION condition and RESET condition. These conditions cause certain BUS DEVICE actions and can alter the bus phase sequence.

### A.6.1 ATTENTION CONDITION

ATTENTION allows the INITIATOR to signal the TARGET of a waiting DEFINE message. The TARGET may access the message by invoking a MESSAGE OUT phase.

The INITIATOR creates the ATTENTION condition by asserting ATN at any time except during the BUS FREE phase. The TARGET responds when ready with the MESSAGE OUT phase. The INITIATOR keeps ATN asserted if more than one byte is to be transferred.

### A.6.2 RESET CONDITION

The RESET condition, created by the assertion of RST is used to immediately clear all devices from the bus and to reset these devices and their associated equipment.

RESET can occur at any time and takes precedence over all other phases and conditions. Any device (whether active or not) can invoke the RESET condition. On RESET, all devices will immediately (within a BUS CLEAR DELAY) deassert and passively release all bus signals except RST itself. A TARGET capable of continuing an I/O operation after being interrupted by RESET will clear any I/O operation that has not been established.

The RESET condition stays on for at least one RESET HOLD TIME. During the RESET condition, no bus signal except RST can be assumed valid.

Regardless of the prior bus phase, the bus resets to a BUS FREE phase (and then starts a normal phase sequence) following a RESET condition.

## A.7 PHASE SEQUENCING

Phases are used on the bus in a prescribed sequence. In all systems, the RESET condition can interrupt any phase and is always followed by the BUS FREE phase. Also, any other phase can be followed by the BUS FREE phase.

The normal progression is from BUS FREE to SELECTION, and from SELECTION to one or more of the INFORMATION TRANSFER phases (COMMAND, DATA, STATUS or MESSAGE).

There are no restrictions on the sequencing between INFORMATION TRANSFER phases. A phase may even follow itself (e.g., a DATA phase may be followed by another DATA phase).

## A.8 TIMING

A timing chart is provided in Figure A-1. Unless otherwise indicated, the delay time measurements for each device are calculated from signal conditions existing at the device BUS PORT. Delays in the bus cable need not be considered for these measurements. See Table A-5 for minimum and maximum timing values.

### A.8.1 SELECTION ABORT TIME: 200 MICROSECONDS (MAX.)

The maximum delay allowed from SELECT detection until a BSY response is generated by a TARGET (or INITIATOR) during SELECTION. This is not SELECT TIMEOUT.

### A.8.2 BUS CLEAR DELAY: 800 NANoseconds (MAX.)

The maximum time allowed for a device to stop driving all bus signals after the release of BSY when going to BUS FREE.

### A.8.3 BUS SET DELAY: 1.8 MICROSECONDS (MAX.)

The maximum time from detection of BUS FREE until BSY is driven.

A.8.4 BUS SETTLE DELAY: 450 NANOSECONDS (MIN.)

The time to wait for the bus to settle after changing certain control signals.

A.8.5 CABLE SKEW: 10 NANOSECONDS (MAX.)

The maximum difference in propagation time allowed between any two bus signals when measured between any two SCSI devices.

A.8.6 DESKEW DELAY: 45 NANOSECONDS (MIN.)

The time required for deskew of certain signals.

A.8.7 REQ RESPONSE TIMEOUT: 250 MILLISECONDS (MIN.)

The delay allowed between assertion of REQ by the TARGET and time out (due to lack of ACK from the INITIATOR).

A.8.8 RESET HOLD TIME: 25 MICROSECONDS (MIN.)

The minimum time during which RST is asserted. No maximum.

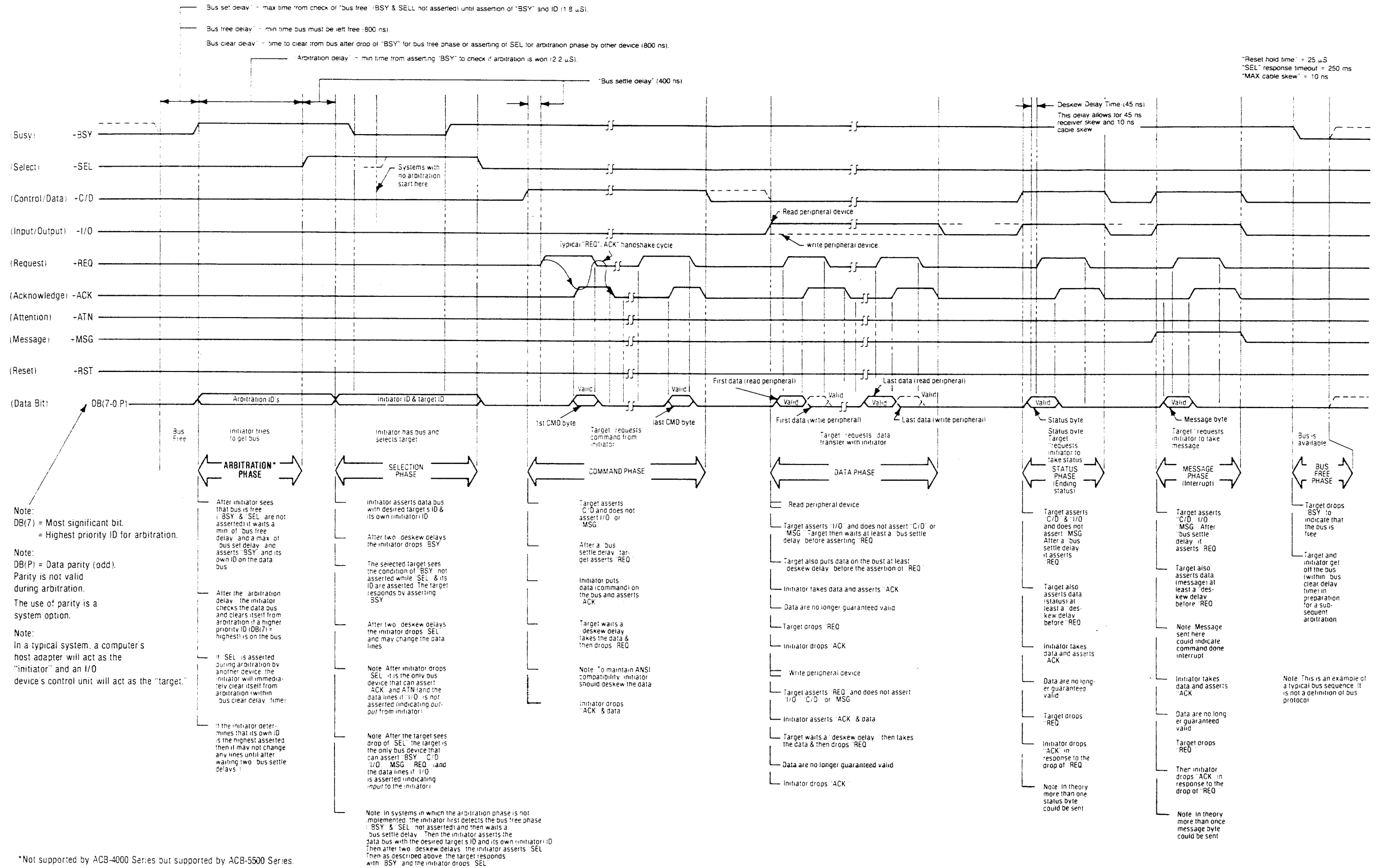
A.8.9 SELECT TIMEOUT: 250 MILLISECONDS (MIN.)

The delay allowed for a BSY response from a TARGET before time out during SELECTION.

The INITIATOR can deassert the ATN signal during the RESET condition, during a BUS FREE phase, or while the REQ signal is asserted and before the ACK signal is asserted during the last REQ/ACK handshakes of a MESSAGE OUT phase.

TABLE A-5. SCSI BUS TIMING

<u>PARAMETER</u>	<u>MINIMUM</u>	<u>MAXIMUM</u>
Aborted Selection Time		200 usec
Bus Clear Delay		800 nsec
Bus Set Delay		1.8 usec
Bus Settle Delay	400 nsec	
Cable Skew		10 nsec
Deskew Delay	45 nsec	
REQ Response Timeout	250 msec	
RST Hold Time	25 usec	
SEL Timeout	250 msec	



\*Not supported by ACB-4000 Series but supported by ACB-5500 Series.

Figure A-1. SCSI Timing



## APPENDIX B. BASIC SCSI HOST ADAPTER

---

The enclosed schematic is a typical example of a SCSI Host Adapter. This example is a S-100 Bus to SCSI Host Adapter.

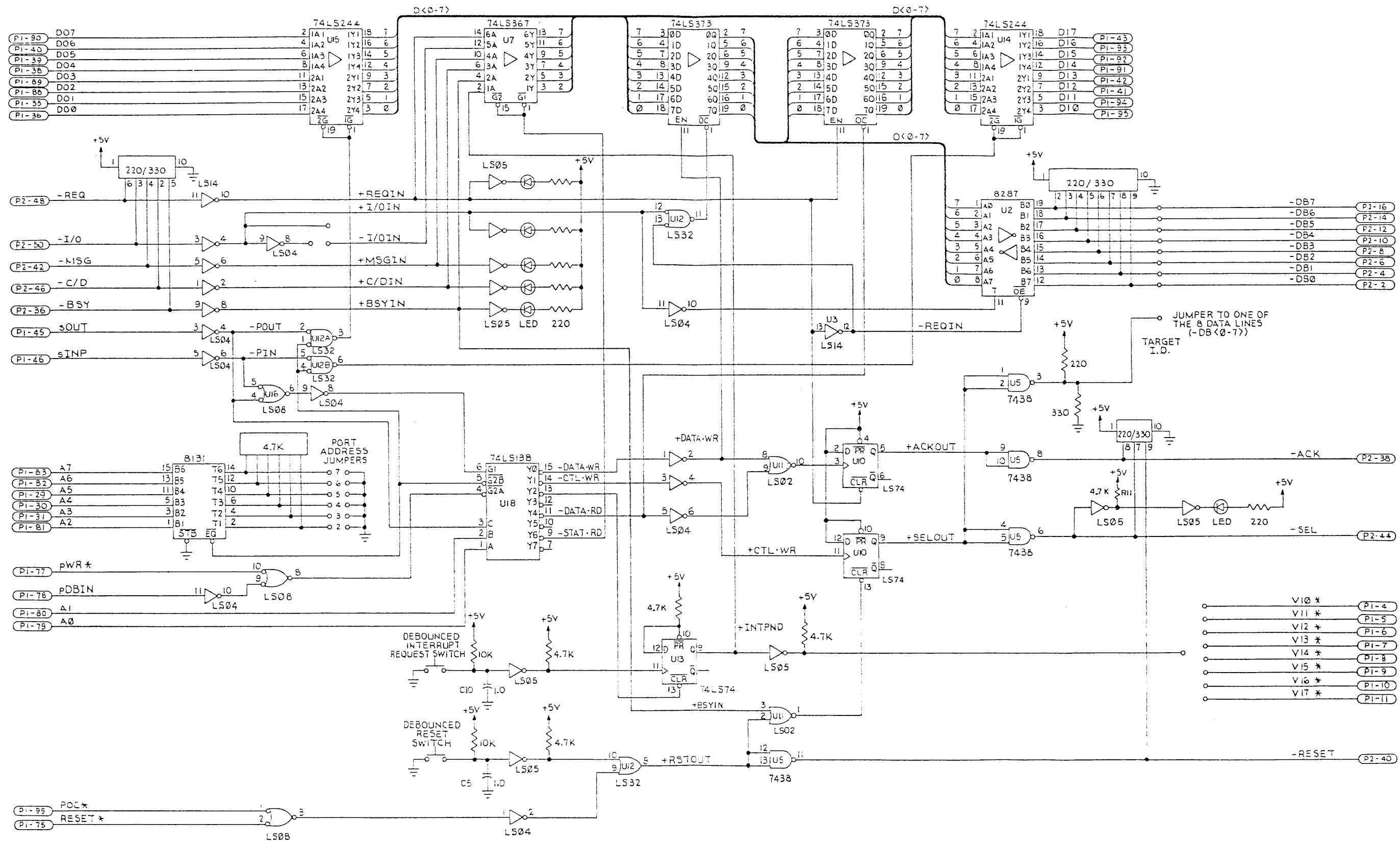


Figure B-1. Example of SCSI Host Adapter

## APPENDIX C. HOW TO WRITE A SCSI I/O DRIVER

---

This appendix details the concepts needed to write a SCSI I/O driver. This structure is recommended when writing an ACB-4000 Series driver.

### C.1 Bus Phase Considerations

An important point to remember in designing a drive routine is that once the controller is started by the host, THE CONTROLLER CONTROLS THE SASI/SCSI BUS. The controller drives the data direction line (I/O), the phase lines (C/D and MSG) and initiates data transfers (REQ). The host driver should make no assumptions about the bus phases or byte counts. In fact, the controller can (and will) change phases between operations. The SCSI spec allows the controller to go through intermediate phases. Thus, the phase lines (C/D and MSG) are only valid when the controller asserts REQ. Do not write your driver or allow your hardware to follow phases when REQ is not active or it may be 'fooled' by phase changes between REQ's. Also, other controllers only support six byte commands, thus some users have set up counters in their software to only send a six-byte command. Since the ACB-4000 Series controller supports 6 and 10-byte commands, the hardware/software should not count out the command bytes but rather should send command bytes as long as the controller requests them. Trust the controller; it 'knows' how many bytes it needs.

The sequence of operations for a single command would be:

1. Select the controller onto the bus (wake it up).
2. Send it command bytes until it changes phase (do not count bytes).
3. If requested, send/receive data until phase changes (do not count bytes; controller will determine data direction).
4. Receive (REQ/ACK cycle) 1 status byte and save for evaluation.
5. Receive (REQ/ACK cycle) 1 message byte (always 00 for ACB-4000 Series) may be ignored.
6. Check status byte. If busy bit set, resend command; if check bit set, send sense command to get error code.

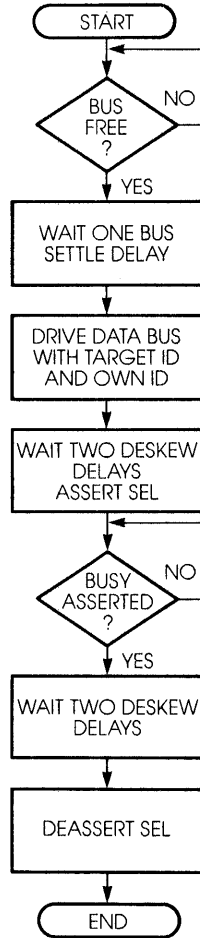
#### **NOTE:**

SOME SASI-TYPE CONTROLLERS DID SUPPORT THE SCSI DEFINED BUSY BIT. HOWEVER, THESE CONTROLLERS DO SET THIS BIT TO ZERO ENABLING THIS PROCEDURE TO WORK WITH THE ACB-4000 SERIES CONTROLLERS AND BE DOWN-LEVEL COMPATIBLE WITH OTHER CONTROLLERS.

## C.2 I/O DRIVER FLOW CHART

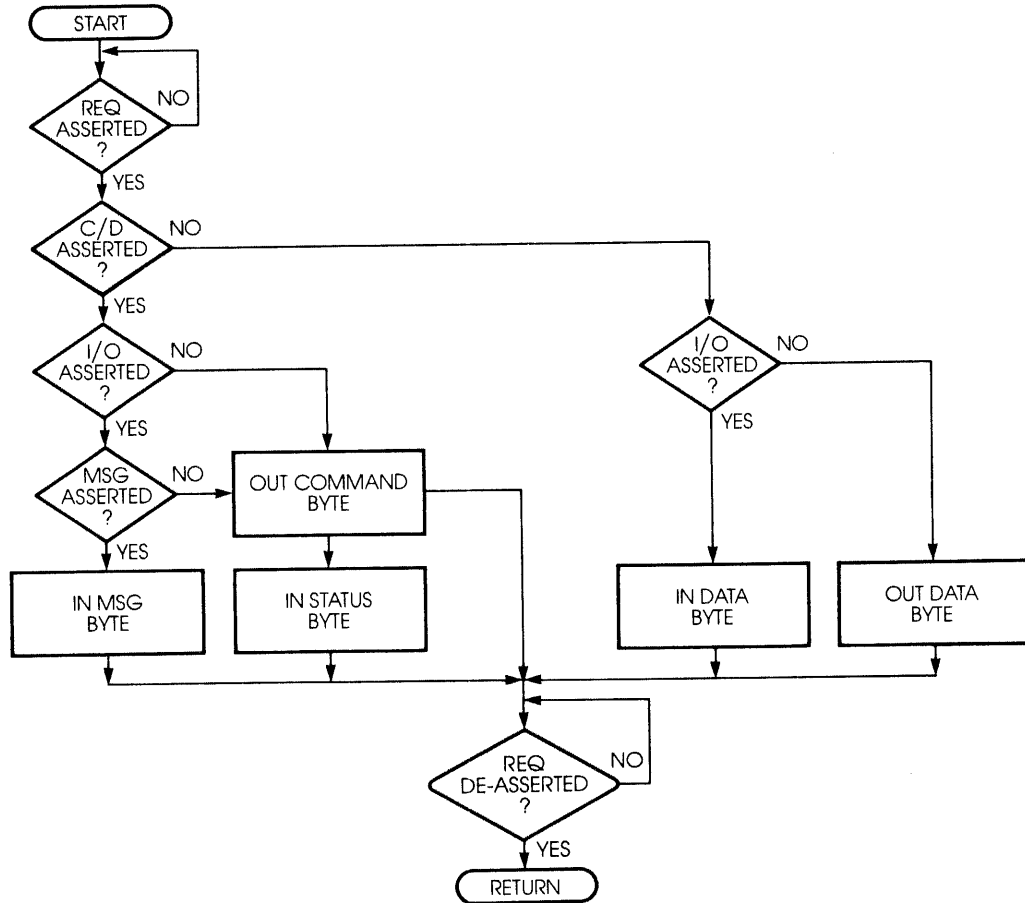
The following flow charts demonstrate how to program the bus phase changes for an ACB-4000 series controller.

### C.2.1 SELECTION PHASE



C.2.2 COMMAND, DATA, STATUS AND MESSAGE PHASES

The following flow charts show how to program all phase changes following selection in the ACB-4000 series controllers. One software routine may be used for the SCSI handshake protocol for each of these phases, and is recommended over the counting of bytes.



This flow chart assumes that an in-port or out-port will generate a host ACK.

### C.3 TIMING CONSIDERATIONS

The controller firmware provides timeouts on all operations which need to be timed out. It is not advisable for the host software to second guess the firmware timeout values.

If software timeouts are used, they should be significantly greater than firmware timeouts in order to prevent minor firmware changes from creating mandatory software changes. When timeout values are used, the following maximum times should be kept in mind. A FORMAT command may take over one hour, a VERIFY command of 64K blocks may take a couple of minutes and a READ/WRITE command with seek recovery and read retries may take a couple of seconds.

### C.4 SAMPLE DRIVER ROUTINE

Following is a sample of the driver routine. The hardware of this sample driver is assumed to be a simple PIO (Programmed Input Output) driver of the data bus and control lines. The only driver function handled in hardware is the ACK signal. We assume that reading/writing the data bus port when REQ is active will cause an automatic ACK response to be sent to the controller.

```

TITLE      *****          SAMPLE SASI DRIVER ROUTINES
CSEG
NAME      ('DRIVER')
.PHASE 100H          ;assemble in CP/M TPA

```

```

;*****
;*
;*          HARDWARE EQUATES.
;*
;*****

```

```

BASE      EQU      0D0H          ;base port address of host adaptor
HADATA    EQU      BASE          ;SASI data bits
HACTRL    EQU      BASE+1        ;enable and SEL output bits
HASTAT    EQU      BASE+2        ;SASI status bits

```

```

;*****
;*
;*          CONTROL REGISTER BIT EQUATES
;*
;*****

```

```

SELECT    EQU      40H          ;asserts SEL to get
                                   ;controller onto bus

```

```

;*****
;*
;*          RETURNED STATUS BYTE BIT EQUATES
;*
;*****

```

```

BSYBIT    EQU      08H          ;LUN is busy
ERROR     EQU      02H          ;error in last operation

```

```

;*****
;*
;*          STATUS REGISTER BIT EQUATES
;*
;*****

```

```

REQ       EQU      80H          ;SASI REQ line      ;asserted)
IO        EQU      40H          ;SASI I/O line (   .nput)
MSG       EQU      20H          ;SASI MSG line (l=asserted)
CD        EQU      10H          ;SASI C/D line (l=command)
BUSY     EQU      08H          ;SASI BUSY line (l=asserted)

```

```

;*****
;*
;*          THE DRIVER ROUTINE ASSUMES THAT THE COMMAND
;*          IS STORED IN LOCATION 'CMD' AND A RAM BUFFER
;*
;*****

```

```

;*      BIG ENOUGH FOR THE DATA HAS BEEN ALLOCATED AT      *
;*      LOCATION 'BUFFER'. IF THE COMMAND COMPLETES          *
;*      WITHOUT ERROR, LOCATION 'ENDSTA' WILL BE ZERO.      *
;*
;*      IF AN ERROR WAS DETECTED, LOCATION 'ENDSTA'         *
;*      WILL CONTAIN 02 AND THE 4 RAM LOCATIONS              *
;*      STARTING AT 'ERCODE' WILL HOLD THE ERROR CODE      *
;*      AND ADDRESS.                                        *
;*
;*****
DRIVER:  PUSH    PSW                ;temp save A/PSW
        PUSH    B                  ;temp save B/C
        PUSH    D                  ;temp save D/E
        PUSH    H                  ;temp save H/L
RETRY:   CALL    WAKEUP             ;get controller onto bus
        LXI     H,CMD              ;point H/L at command to send
        CALL    SEND               ;send command
        CALL    WAITRQ             ;wait for REQ to come active
        IN     HASTAT              ;get bus phase status
        ANI    CD                  ;test for command or data
        JNZ    GETSTA              ;if command, no data so get out
        IN     HASTAT              ;get a new copy of status
        LXI    H,BUFFER
        ANI    IO                  ;test the I/O direction
        JZ     DOWRT               ;if zero, do a write
        CALL   READ                ;otherwise, do a read
        JMP    GETSTA              ;and get status

DOWRT:   CALL    WRITE             ;call the data write routine
GETSTA:  CALL    STATUS            ;get drive status
        JC     RETRY              ;if busy, retry the command
        JZ     RETURN             ;if no error, return

;*****
;*
;*      THERE WAS AN ERROR IN THE LAST COMMAND, GET        *
;*      THE SENSE INFORMATION.                              *
;*
;*****

        CALL    WAKEUP             ;get controller onto bus
        LXI    H,SENSE             ;point H/L at command to send
        CALL    SEND               ;send command
        CALL    WAITRQ            ;wait for REQ to come active
        LXI    H,ERCODE           ;point H/L at error code
        CALL    READ               ;get the error code
        CALL    STATUS             ;get status (will be 00)
        MVI    A,02H              ;load error code into A
        STA    ENDSTA             ;save as status

RETURN:  POP     H                 ;recover H/L
        POP     D                 ;recover D/E
        POP     B                 ;recover B/C
        POP     PSW               ;recover A/PSW
        RET                      ;and return

```



```

;#####
;#
;#          SUPPORT SUBROUTINES          #
;#
;#####

```

```

;*****
;*
;*          THE WAKEUP ROUTINE GETS THE CONTROLLER          *
;*          ONTO THE BUS                                    *
;*
;*
;*****

```

```

WAKEUP: PUSH     PSW           ;temp save A/PSW
TSTBSY: IN       HASTAT        ;get current SASI status
        ANI      BUSY          ;test busy line
        JNZ     TSTBSY        ;is bus is busy, wait in loop
        MVI     A,01          ;controller ID=01
        OUT     HADATA        ;put it onto the data bus
        MVI     A,SELECT      ;activate select bit
        OUT     HACTRL        ;assert SEL line

CKBUSY: IN       HASTAT        ;get current SASI status
        ANI      BUSY          ;test only the BUSY line
        JZ      CKBUSY        ;wait for busy from controller
        MVI     A,00H         ;release the SEL line....
        OUT     HACTRL        ;once BUSY is active
        POP     PSW           ;then recover A/PSW
        RET


```

```

;*****
;*
;*          THE SEND ROUTINE SENDS THE COMMAND POINTED     *
;*          BY THE H/L REGISTER TO THE CONTROLLER.         *
;*
;*
;*****

```

```

SEND:   PUSH     PSW           ;temp save A/PSW
        PUSH     H             ;temp save H/L
SEND1:  CALL     WAITRQ        ;wait for REQ from controller
        IN       HASTAT        ;get SASI status
        ANI      CD           ;test command/data bit
        JZ      RET1          ;if data phase, get out
        IN       HASTAT        ;get status again
        ANI      IO           ;test the direction line
        JNZ     RET1          ;if command in phase, get out
        MOV     A,M           ;if command out, get next byte
        OUT     HADATA        ;put on data bus
        INX     H             ;bump pointer
        JMP     SEND1         ;and loop back

```

```

RET1:  POP    H           ;recover H/L
       POP    PSW        ;recover A/PSW
       RET                      ;and return

```

```

;*****
;*
;*   THE READ ROUTINE RECEIVES THE DATA AND SAVES
;*   IN THE BUFFER POINTED BY THE H/L REGISTER
;*
;*****

```

```

READ:  PUSH    PSW        ;temp save A/PSW
       PUSH    H          ;temp save H/L
READ1: CALL    WAITRQ     ;wait for REQ from controller
       IN      HASTAT     ;get SASI status
       ANI     CD         ;test command/data bit
       JNZ     RET2       ;if command, we are done
       IN      HADATA     ;if still data, get a byte
       MOV     M,A        ;save it in ram
       INX    H          ;bump pointer
       JMP     READ1      ;and loop till command phase

```

```

RET2:  POP    H           ;recover H/L
       POP    PSW        ;recover A/PSW
       RET                      ;and return

```

```

;*****
;*
;*   THE WRITE ROUTINE SENDS THE DATA IN THE BUFFER
;*   POINTED BY THE H/L REGISTER
;*
;*****

```

```

WRITE: PUSH    PSW        ;temp save A/PSW
       PUSH    H          ;temp save H/L
WRITE1: CALL    WAITRQ     ;wait for REQ from controller
       IN      HASTAT     ;get SASI status
       ANI     CD         ;test command/data bit
       JNZ     RET3       ;if command, we are done
       MOV     A,M        ;if still data, get buffer byte
       OUT    HADATA     ;send to controller
       INX    H          ;bump pointer
       JMP     WRITE1     ;and loop till command phase

```

```

RET3:  POP    H           ;recover H/L
       POP    PSW        ;recover A/PSW
       RET                      ;and return

```

```

;*****
;*

```

```

;*      THE STATUS ROUTINE GETS THE STATUS AND MESSAGE      *
;*      BYTES FROM THE CONTROLLER. THE ROUTINE RETURN      *
;*      WITH THE ZERO BIT SET IF THERE WAS NO ERROR AND    *
;*      THE CARRY BIT SET IF THE CONTROLLER WAS BUSY      *
;*
;*****
STATUS: CALL    WAITRQ      ;wait for request
        IN      HADATA     ;get the status byte
        STA     ENDSTA     ;save in ram
        CALL    WAITRQ     ;wait for request
        IN      HADATA     ;get (and ignore message)
        LDA     ENDSTA     ;get end status
        ANI     BSYBIT     ;test the busy bit
        JZ      NOTBSY    ;if not busy, jump around
        STC     ;if busy, set carry
        JMP     RET4       ;and return
NOTBSY: LDA     ENDSTA     ;if not busy, get status again
        ANA     A          ;test value
RET4:   RET          ;and return

```

```

;*****
;*
;*      THE WAITRQ ROUTINE WAITS FOR THE CONTROLLER      *
;*      TO ASSERT THE REQ LINE OF THE SASI BUS.          *
;*
;*****

```

```

WAITRQ: PUSH    PSW        ;temp save A/PSW
WAITLP: IN      HASTAT    ;get current SASI status
        ANI     REQ       ;look at the REQ line
        JZ      WAITLP    ;loop till request active
        POP    PSW       ;recover A/PSW
        RET          ;and return

```

PAGE

```

;*****
;*
;*      BUFFERS AND CONSTANTS                            *
;*
;*****

```

```

SENSE:  DB      03,00,00      ;sense command for errors
        DB      00,00,00
CMD:    DB      00,00,00,00,00 ;10 byte command area
        DB      00,00,00,00,00
ENDSTA: DB      00           ;ending status
ERCODE: DB      00,00,00,00  ;error code
BUFFER: DS      256         ;256 byte data buffer

```

END

## APPENDIX D. SASI TO SCSI CONVERSION AND DIFFERENCES

---

### D.1 QUICK INSTALLATION OF THE ACB-4000 SERIES CONTROLLERS

An Adaptec 4000 Series Controller will easily replace a SASI-like Xebec or WD controller by following these three simple steps.

- 1) Install a jumper across the I-J jumper position of the 4000 series controller board. This jumper enables the optional command set.
- 2) When looking at the board with the power connector in the lower right-hand corner, locate the three square solder pads in the upper center of the board. Turn the board over and cut the trace between H and the center jumper and install a trace between P and the center jumper. This diverges from the SCSI specification by providing for a pulse select rather than a level select. See Figure 3-1.
- 3) Reformat the disk by sending the MODE SELECT command followed by the FORMAT command. Format utilities which use the FORMAT TRACK command will have to be changed to use the MODE SELECT and FORMAT commands instead. Software drivers will be compatible, with the exception of the FORMAT TRACK command.

### D.2 INTRODUCTION

The Adaptec ACB-4000 Series Controller offers several advantages over similar controllers offered by competition such as Xebec or Western Digital. Some of the advantages are as follows:

\* Higher Performance: The Adaptec controller supports non-interleaved operation, which allows a full track to be read in one revolution of the disk. Other controllers typically take three or four revolutions.

Even when running with an interleave, the Adaptec controller still offers better performance over competition due to its double-buffering capability.

\* Device Independence: Since the controller puts the drive related information (such as number of cylinders, heads, step rate, etc.) on the drive at format time, the operating system need not worry about the physical characteristics of the drive during normal operation. The system integrator or OEM is therefore free to change the drives without having to worry about its impact on his software.

\* Defect Handling: The Adaptec controller handles defects by mapping out the problem at the sector level. This maximizes the usable drive capacity, and maximizes performance by eliminating seeks to alternate tracks. The Xebec or WD controller formats out an entire track, and sets up an alternate track usually on the inside of the disk. This maps out an entire track for just one bad sector, and also reduces performance.

\* Search Capability: The Adaptec ACB-4000 Series has the ability to Search (Scan) the drive for an "Equal" or "Not Equal" compare between sectors on the drive, and data in the sector buffer. This can be used by the operating system to speed up a search based on a key field, or to verify a disk format operation on a byte for byte basis.

\* Verify Capability: The Adaptec controller has the ability to verify information, either immediately after it is written or written earlier. This improves performance by eliminating the need to read information into the host for the purpose of verifying it.

\* ANSI Standard: The Adaptec controller supports the ANSI standard SCSI bus without deviation. This allows the OEM or system integrator to add on other SCSI controllers to the bus.

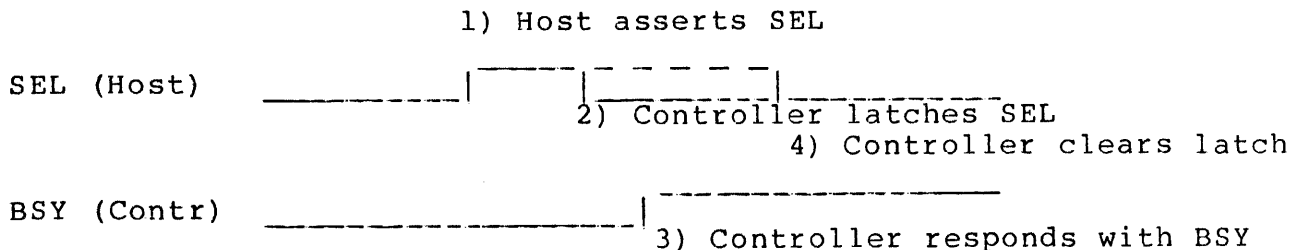
### D.3 HARDWARE

In order to design in the Adaptec ACB-4000 Series Controller board into systems based around the Xebec S1410 or S1410A, or the WD 1002-SHD controller, the following modifications will have to be made:

#### 1) Controller Selection

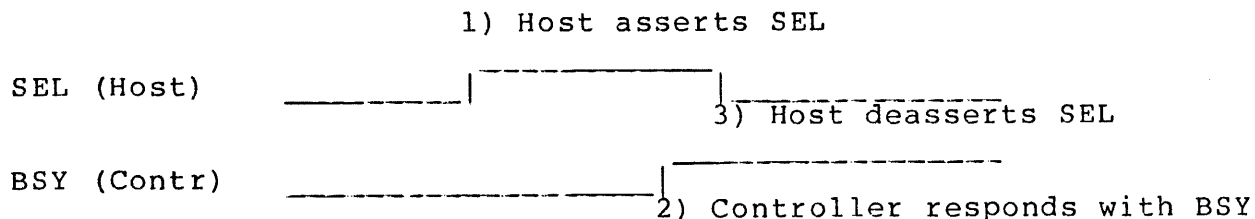
In order to communicate with the controller, the host system has to first select the appropriate controller. This is done by the host, by asserting the SEL line. In addition, the host also asserts the data bit corresponding to the controller's ID. The selected controller responds with an appropriately timed BSY signal.

The Xebec or WD controller will respond to a "pulse" select technique. In such a setup, the host puts out a small pulse on the SEL line, which is latched by the controller. The controller then responds with a BSY signal. This is shown below.



Pulse Selection

The Adaptec controller, in keeping with the SCSI standard, responds to a "level" select. In this case, after the host asserts the SEL line, it has to wait until the controller responds with BSY, before deasserting SEL. This is shown below.



Level Selection

If the host system uses a pulse select technique, then a change has to be made to either the host adapter or the ACB-4000 Series Controller. In most cases the host adapter usually uses programmed I/O for the selection process, and so this is a trivial software change. In the PIO-based selection process, the host has to keep SEL asserted, in software, until the controller asserts BSY. After this the SEL line can be deasserted.

If this is not possible, then the ACB-4000 Series controller has to be modified by means of one cut and one jumper, as detailed in Step 2 above.

**NOTE:**

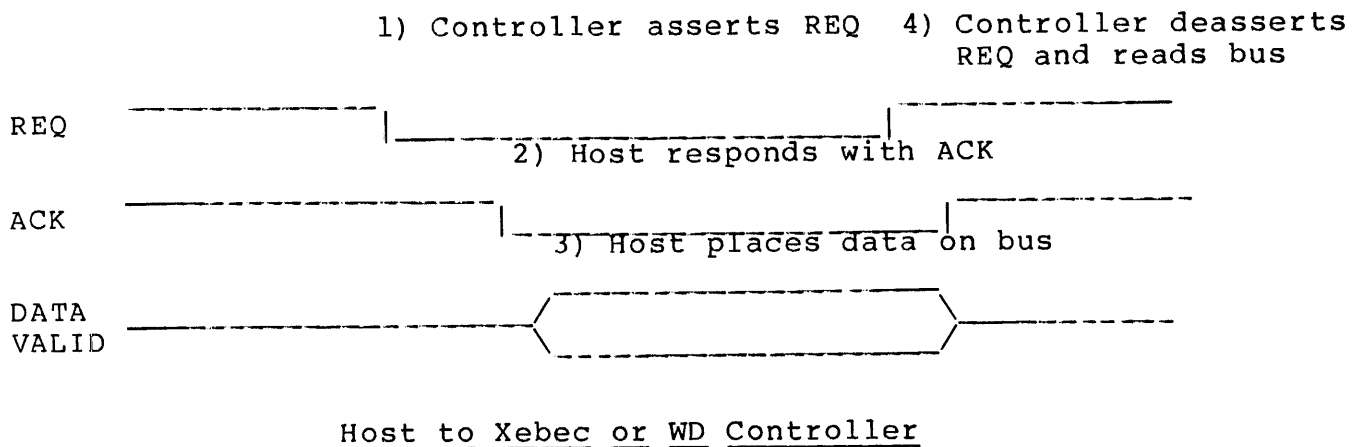
LEVEL SELECT IS THE CORRECT WAY TO SELECT A SCSI DEVICE, AND THUS ONCE THE ADAPTEC BOARD HAS BEEN MODIFIED TO ACCEPT PULSE SELECT, IT MUST BE THE ONLY CONTROLLER ON THE BUS. THIS MAY NOT BE A MAJOR PROBLEM SINCE THIS IS THE CASE IN MANY SYSTEMS.

2) SCSI Handshaking

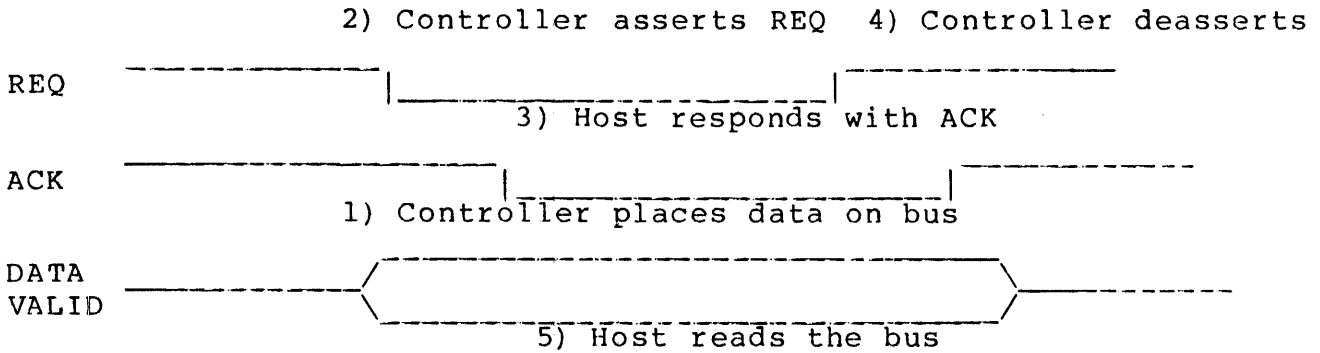
The Adaptec controller supports commands where the Command Data Block (CDB) can be six bytes (class 0) or 10 bytes (class 1). Therefore it relies on a full REQ/ACK handshake for the transfer of each byte. Since Xebec and WD support only commands which are six bytes long, the CDB can be transferred by the host counting bytes. This is not proper however, and the host should always trust the controller, by checking the Bus Phase and REQ. Based on the design, this may be a hardware fix, or if based on programmed I/O, it may be a minor software fix in the host adapter code.

According to the ANSI SCSI specification, data transfers should begin with the controller asserting REQ, requesting the data transfer. The host then appropriately places data on, or reads the data from, the bus. The transfer is then concluded by the host asserting ACK, acknowledging the data transfer. The ACB-4000 fully complies with this specification, whereas the Xebec or WD controllers violate this by allowing the host to assert ACK first, presumably concluding the data transfer, and then placing data on or reading data from the bus.

For example, with a host to controller transfer the Xebec or WD controller asserts REQ, initiating the transfer. The host responds by asserting ACK and then places one byte of data on the SASI bus. The controller then deasserts REQ and reads the bus. This is shown below.

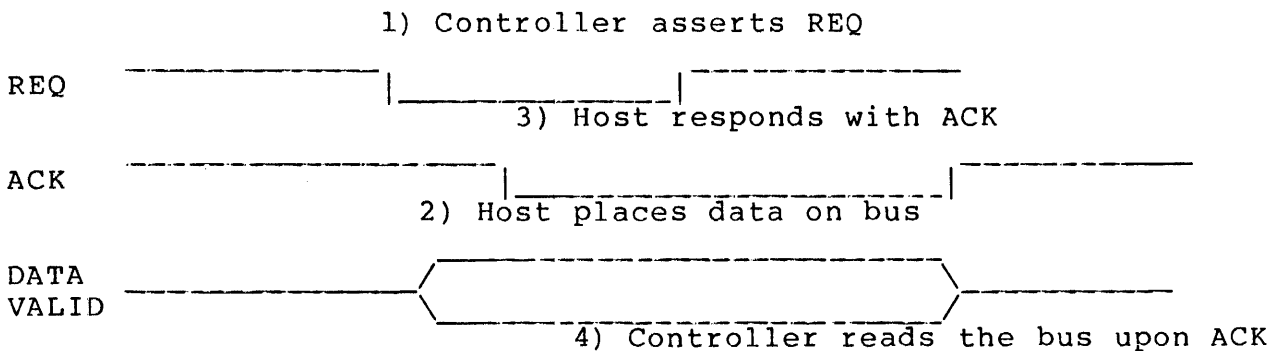


Similarly, with a controller to host transfer, the controller asserts REQ, and places the data on the bus. The host now responds with an ACK. The controller deasserts REQ, causing the host to read the bus.



Xebec or WD Controller to Host

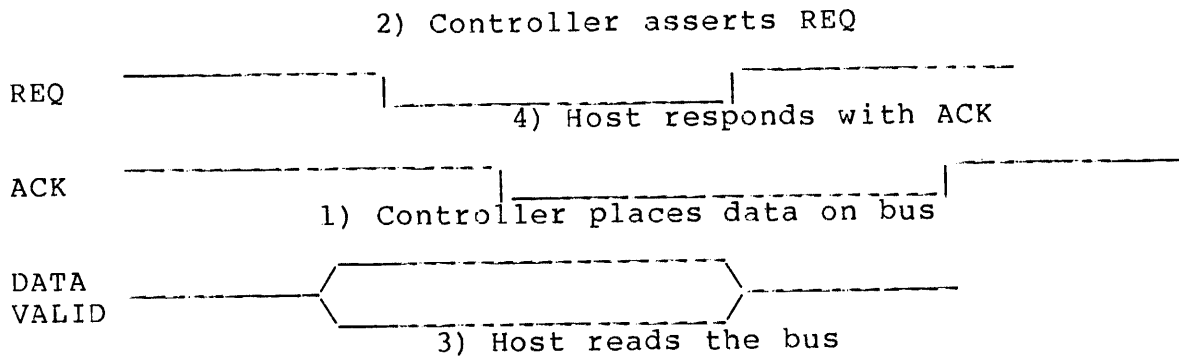
The ACB-4000 Series, keeping true to the SCSI specification, assumes the data transfer is concluded when the host asserts acknowledge. Therefore, to assure a proper data transfer from the host to the ACB-4000 Series, the host must respond to the controller request by placing data on the bus before asserting acknowledge.



Host to Adaptec (SCSI) Controller



Likewise, to transfer data from the ACB-4000 Series to the host the host must respond to the controller REQ by reading from the bus and then asserting acknowledge.



Adaptec (SCSI) Controller to Host

Any attempts by the host to read or write data after asserting acknowledge, will cause missing bytes, double-counting bytes and misread bytes.

3) Drive Cabling

Xebec allows the use of the radial cable for a drive to be plugged into either connector J0 or J1. In the case of the ACB-4000 Series, Drive 0 must be connected to J0, and Drive 1 must be connected to J1.

4) SCSI Cabling

On the 50-pin SCSI connector Pin 1 is reversed with respect to some SASI type controllers.

#### D.4 SOFTWARE

In most cases, driver software written for a Xebec or WD controller will easily carry over to the Adaptec controller. There are, however, a few areas, where some minor changes have to be made. The format drive routine, which is usually a separate utility, is the one that is affected the most. The critical areas are described below:

1) The Format routine has to be rewritten. This is because the Xebec or WD controller uses Format Track, Format Bad Track and Format Alternate Track, to handle media defects. The Adaptec controller optimizes performance and disk capacity by using an extended Format Drive command to do sector level defect handling. A Mode Select command is used before format, to enable the information to be written on the drive at format time.

This is described in the installation section of the ACB-4000 Series Users Guide Manual. A more detailed write up on the defect handling strategies of the Adaptec family of controllers is given in Section 2.

2) In the case of Xebec or WD, the host has to initialize the controller after power-up, by telling it the drive characteristics. Since the Adaptec controller reads drive parameters from the drive at power on, a "0CH" command (Initialize Characteristics) is not required, and hence is not supported.

3) Since the Step Pulse code information is stored on the drive, and retrieved after reset, this information should not be included with the command, as is necessary with a Xebec or WD controller. The sixth byte in the CDB is reserved, and always should be zero.

4) There are a few commands available on the Xebec or WD controller, which are not supported by the ACB-4000 Series. By installing the extended option jumper (jumper I-J), the controller will accept all these commands, but will not act on them. Instead it will come back immediately with command complete status. This jumper also allows the sixth byte and other reserved bytes to be non-zero. This is described in detail in Section 5 of the Manual.

With these minimal changes, the Adaptec controller can easily replace a Xebec or WD controller, and offer several benefits to the user in the form of increased performance and easier software maintenance.

From this starting point, the system integrator or OEM can enhance his system by using several of the class 0 and class 1 SCSI commands available on the ACB-4000 Series. A brief summary of the commands supported by the ACB-4000 Series follows:

<u>COMMAND</u>	<u>CLASS</u>	<u>OP CODE</u>
Test Ready	0	00
Rezero	0	01
Request Status	0	03
Format Drive	0	04
Read	0	08
Write	0	0A
Seek	0	0B
Translate	0	0F
Write Data Buffer	0	13
Read Data Buffer	0	14
Mode Select	0	15
Mode Sense	0	1A
Start/Stop	0	1B
Receive Diagnostics	0	1C
Send Diagnostics	0	1D
Read Capacity	1	25
Read Extended	1	28
Write Extended	1	2A
Write and Verify	1	2E
Verify	1	2F
Search Equal	1	31

The following Xebec/WD commands (Op codes) are not supported by the ACB-4000:

<u>COMMAND</u>	<u>CLASS</u>	<u>OP CODE</u>
Check Track	0	05
Format Track	0	06
Format Bad Track	0	07
Set Parameters	0	0C
Last Corrected Burst Length	0	0D
Format Alternate Track	0	0E
Write Sector Buffer	0	0F
Read Sector Buffer	0	10
Ram Diagnostic	7	E0
Drive Diag	7	E3
Controller Diag	7	E4
Read Long	7	E5
Write Long	7	E6

## D.5 SUMMARY

As can be seen, the changes necessary to incorporate an Adaptec ACB-4000 or ACB-4070 in place of a Xebec S1410 or S1410A, or a WD-1002SHD, is fairly simple. In fact, in most cases, these changes involve modifying the host adapter firmware slightly. Once these changes are done, and the user converts over to the SCSI bus, all the performance benefits of the ACB-4000 Series are available to him.

In addition, the user can further enhance his system's performance by upgrading to the other products in Adaptec's ACB-4000 family or the ACB-5500 family.

## APPENDIX E. HOST AND DRIVE INTERFACES

---

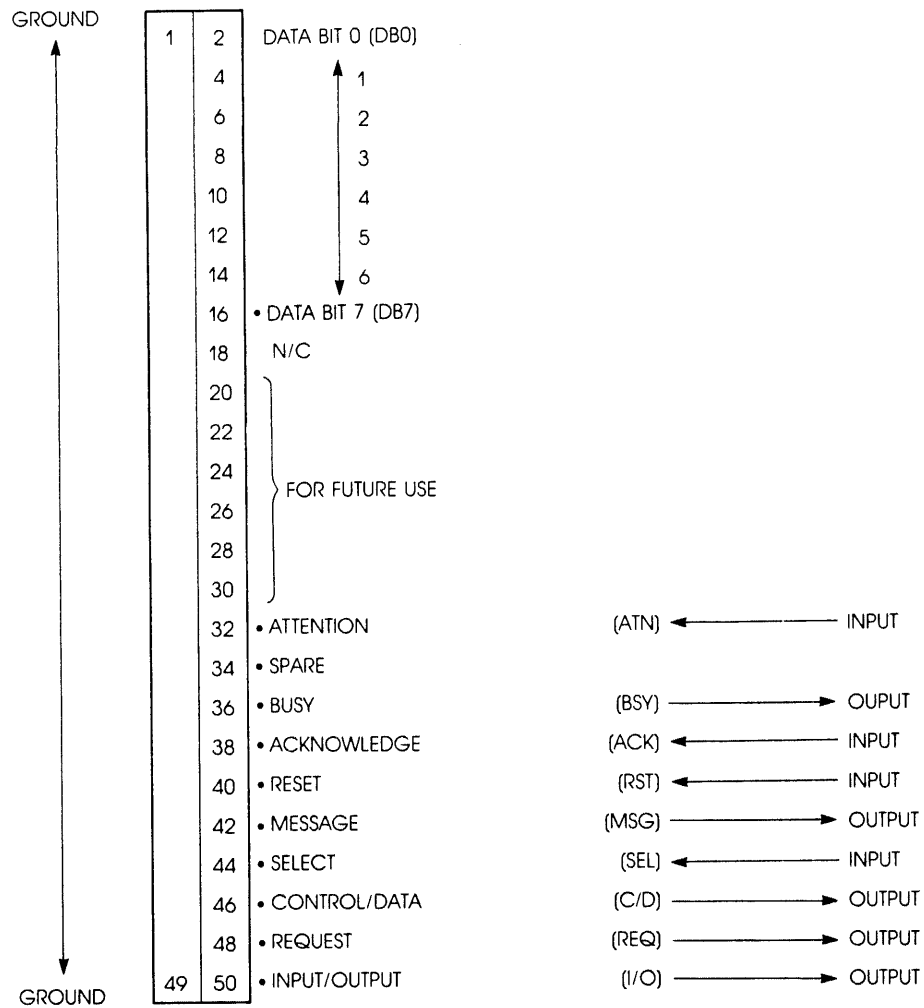
### E.1 HOST ADAPTER INTERFACE

The Adaptec ACB-4000A and ACB-4070 controllers interface to a host adapter according to the proposed ANSI X3T9.2 Standard (SCSI). The data bus is a bidirectional 8-bit parallel interface.

#### E.1.1 HOST ADAPTER INTERFACE - PHYSICAL

A 50-pin flat ribbon connector is provided at J4. The 3M P/N 3425-3000 cable connector is recommended.

Single ended drivers and receivers allow a maximum cable length of 20 feet (six meters) between the host adapter and the controller. All signals are low true. All odd pins are grounded. Figure E-1 shows the SCSI bus pin assignments.



**FIGURE E-1. SCSI BUS PIN ASSIGNMENTS**

## E.1.2 HOST ADAPTER INTERFACE - ELECTRICAL

All signals are low true and use open collector drivers terminated with 220 ohms to +5 volts (nominal) and 330 ohms to ground at each end of the cable.

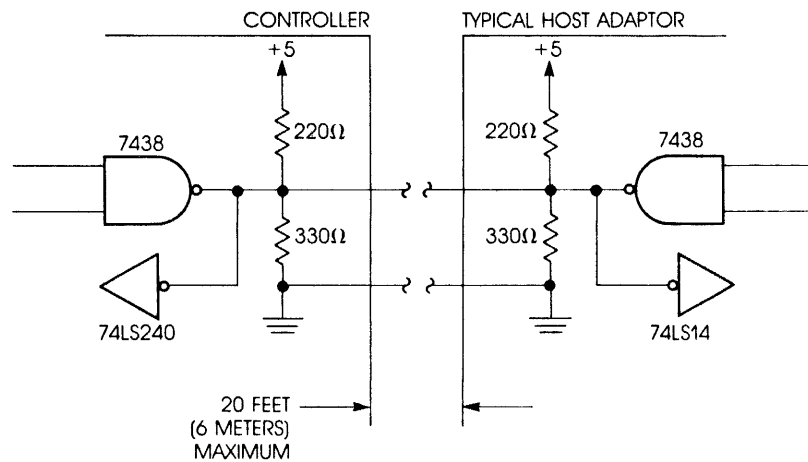
Each signal driven by the controller has the following output characteristics:

True (Signal Assertion) = 0.0 to 0.4 VDC @ 48 mA (max.)

False (Signal Non-Assertion) = 2.0 to 5.25 VDC

A 74LS14 receiver with hysteresis meets this specification.

Figure E-2 shows an example of proper bus termination.

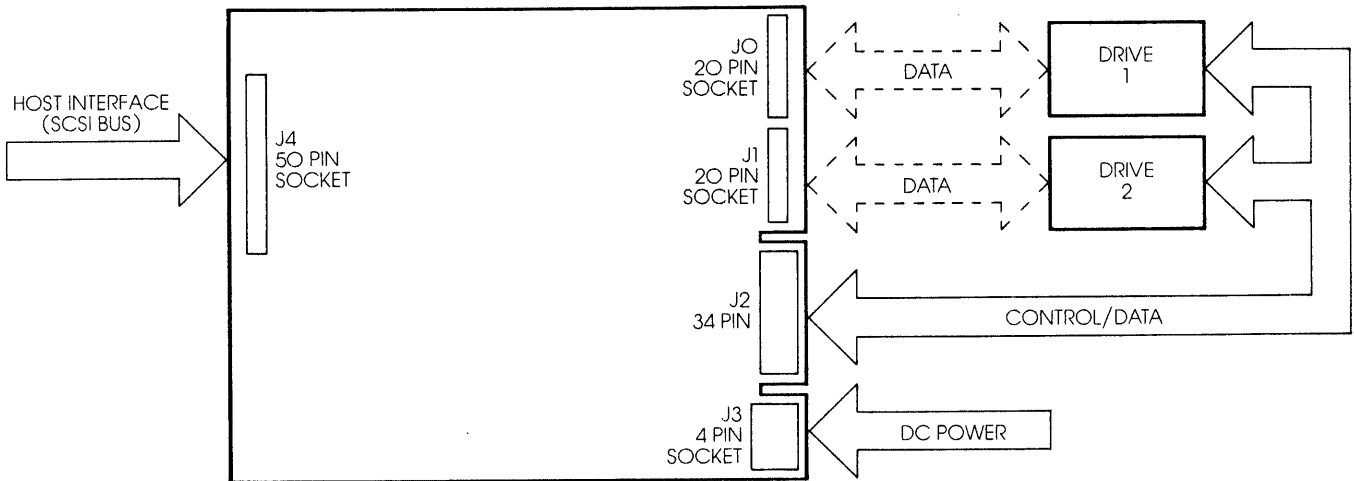


**FIGURE E-2. HOST ADAPTER BUS TERMINATION**

## E.2 DISK DRIVE INTERFACE

The ACB-4000A and ACB-4070 controllers comply with the standard ST506/412 interface.

A system interconnect diagram is shown in Figure E-3. See Section 3.4 for location of the connectors.



**FIGURE E-3. SYSTEM INTERCONNECT DIAGRAM**

### E.2.1 DISK DRIVE INTERFACE - PHYSICAL

J2 is a 34-pin edge connector to which all drive control lines are daisy-chained. Maximum cable length is 20 feet (6 meters). The suggested mating connector for this ribbon cable is 3M P/N 3402-E0000.

The pins are numbered one through 34 with the even pins located on the component side of the controller board. Pin 2 is the pin closest to the power connector (J3). Table E-1 shows pin assignments for connector J2.



J0 and J1 are the radial data connectors to each disk drive. Maximum cable length should not exceed 20 feet (six meters). Suggested mating sockets for these connectors is 3M P/N 3421 Series. Table E-2 shows pin assignments for connectors J0 and J1.

Table E-1. J2 Connector Pin Assignment

GND RTN PIN	SIGNAL PIN	SIGNAL NAME
1	2	Reduced Write Current/Head Select 2 <sup>3</sup>
3	4	Head Select 2 <sup>2</sup>
5	6	Write Gate
7	8	Seek Complete
9	10	Track 0
11	12	Write Fault
13	14	Head Select 2 <sup>0</sup>
15	16	Reserved/Sector*
17	18	Head Select 2 <sup>1</sup>
19	20	Index
21	22	Ready
23	24	Step
25	26	Drive Select 1
27	28	Drive Select 2
29	30	Drive Select 3
31	32	Drive Select 4
33	34	Direction In

\*ACB-4000A controller only.

Table E-2. JO and J1 Connector Pin Assignment

GND RTN PIN	SIGNAL PIN	SIGNAL NAME
2	1	Drive Selected
4	3	Reserved
6	5	Reserved
8	7	Reserved
	9,10	Reserved
12	11	GND
	13	MFM Write Data/2,7 RLL Write Data*
	14	MFM Write Data/2,7 RLL Write Data*
16	15	GND
	17	MFM Read Data/2,7 RLL Read Data*
	18	MFM Read Data/2,7 RLL Read Data*
20	19	GND

\*ACB-4070 controller only.

### E.2.2 DISK DRIVE INTERFACE - ELECTRICAL

The last physical drive on the control bus daisy chain must be terminated with a resistor pack provided by the drive manufacturer. The control signal driver/receiver electrical specifications are shown in Figure E-4.

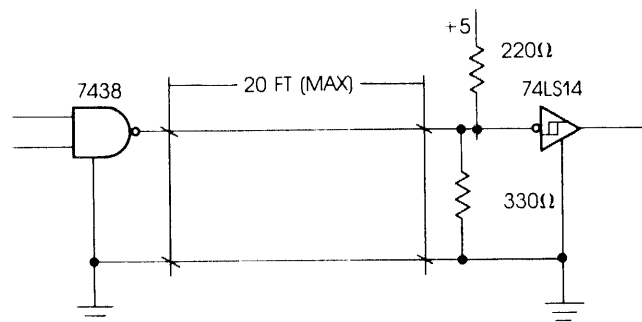


FIGURE E-4. CONTROL DRIVER/RECEIVER LINES

The Control signals are specified at:

True = 0.0 VDC to 0.4 VDC @ I = 48 mA (Max.)

False = 2.5 VDC to 5.25 VDC @ I = +250 uA (open collector)

The read and write MFM data lines are differential signals, present on connectors J0 and J1. The Adaptec receiver/driver pairs meet the required RS-422 specifications. Figure E-5 shows these lines for the ACB-4000A and ACB-4070.

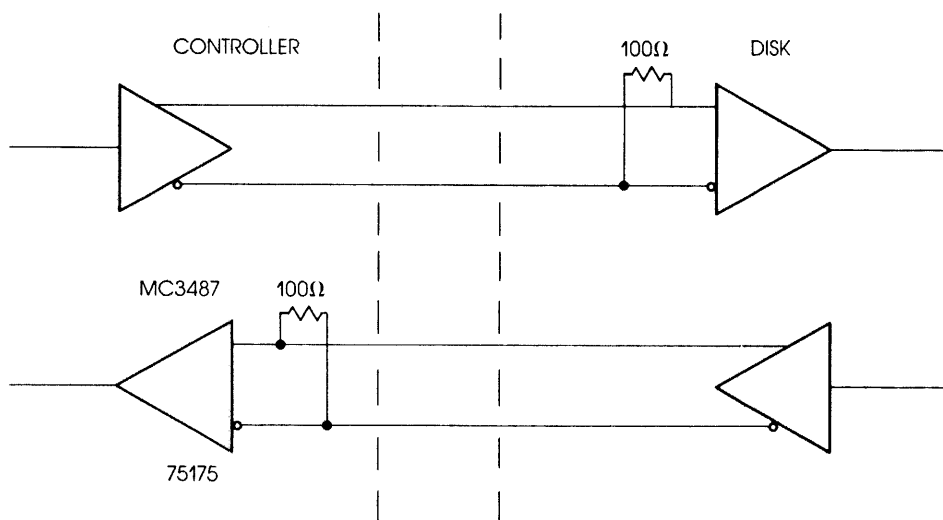


FIGURE E-5. ACB-4000 SERIES DATA RECEIVER/DRIVER PAIRS

## APPENDIX F. DISK FORMAT

---

The following diagrams show an expanded track format for the ACB-4000A and ACB-4070. Note that the ACB-4000A and ACB-5500 have the same track format.

Adaptec ACB-4000 and ACB-5500 Disk Format

	GAP 1	SYNC	PRE ID		CYL	HD	SEC	FLAG	ECC	GAP 2		PRE DATA DATA		DATA FIELD	ECC GAP 3			GAP 4	TOTAL BYTES PER SCTR
			AM	AM						AM	AM	X	00		00	AM	AM		
256 Byte Sector 1:1 interleave 32 Sectors/Trk	10	12	1	1	1	1	1	1	4	3	12	1	1	256	4	2	19	166	320
> 2:1 interleave 33 Sectors/Trk	10	12	1	1	1	1	1	1	4	3	12	1	1	256	4	2	9	176	310
512 Byte Sector 1:1 interleave 17 Sectors/Trk	10	12	1	1	1	1	1	1	4	3	12	1	1	256	4	2	19	614	576
> 2:1 interleave 18 Sectors/Trk	10	12	1	1	1	1	1	1	4	3	12	1	1	256	4	2	9	218	566
1024 Byte Sector 1:1 interleave 9 Sectors/Trk	10	12	1	1	1	1	1	1	4	3	12	1	1	256	4	2	19	614	1088
> 2:1 interleave 9 Sectors/Trk	10	12	1	1	1	1	1	1	4	3	12	1	1	256	4	2	9	704	1078

Adaptec ACB-4070 Disk Format

	GAP 1	PRE ID SYNC	AM	AM	CYL	HD	SEC	FLAG	ECC	GAP 2	PRE DATA AM	DATA AM	DATA FIELD	ECC	GAP 3	GAP 4	TOTAL BYTES PER SECTOR
	33	00	5E	A1	X	X	X	X	X	00 00	5E	A0	X	X	00 33	33	
256 Byte Sector 1:1 interleave 46 Sectors/Trk	11	16	1	1	1	1	1	1	4	6 16	1	1	256	4	2 21	295	333
> 2:1 interleave 47 Sectors/Trk	11	16	1	1	1	1	1	1	4	6 16	1	1	256	4	2 11	432	323
512 Byte Sector 1:1 interleave 25 Sectors/Trk	11	16	1	1	1	1	1	1	4	6 16	1	1	512	4	2 21	888	589
> 2:1 interleave 26 Sectors/Trk	11	16	1	1	1	1	1	1	4	6 16	1	1	512	4	2 11	559	579
1024 Byte Sector 1:1 interleave 14 Sectors/Trk	11	16	1	1	1	1	1	1	4	6 16	1	1	1024	4	2 21	199	1101
> 2:1 interleave 14 Sectors/Trk	11	16	1	1	1	1	1	1	4	6 16	1	1	1024	4	2 11	339	1091

## APPENDIX G. ADVANCED EXAMPLES

---

The following examples show the power of the ACB-4000 Series Command Set.

### G.1 USING DATA SEARCH NOT EQUAL TO VERIFY AFTER FORMAT

DATA SEARCH EQUAL is a quick and powerful method for verifying a disk after format on a byte-for-byte basis without relying only on ECC. This command may also be used by system software to speed the search for a particular data field. This command will search the drive for a field "Equal" or "Not Equal" to the data in the sector buffer.

To verify a FORMAT using the DATA SEARCH NOT EQUAL follow the steps shown in this example.

- 1) The FORMAT command must be issued first specifying the desired data pattern. This pattern must be noted in order to do a DATA SEARCH NOT EQUAL for the same pattern. In this example the default pattern '6C' will be used.
- 2) Following the FORMAT command, DATA SEARCH EQUAL may be sent. The following is an example of DATA SEARCH EQUAL starting at logical block 0 and searching to logical block FFFFH with the invert bit set.

<u>BYTE</u>	<u>CONTENTS</u>	<u>MEANING</u>
00	31	DATA SEARCH EQUAL
01	10	Logical unit 0 with the invert bit set
02	00	Logical block address, high
03	00	Logical block address, second byte
04	00	Logical block address, third byte
05	00	Logical block address, low
06	00	Reserved
07	FF	Number of blocks, high
08	FF	Number of blocks, low
09	00	Reserved

Byte 1 bit 4 is the Invert bit. This bit inverts the sense of the search operation. In this example, the search not equal will cause the controller to stop on a sector not equal to the search data and report search satisfied with a status byte of equal (04 status). Otherwise, the search will be terminated by a search length equal to the number of blocks specified in the data bytes or the number of blocks specified in the command block. When terminated by the search block count a no sense status will be reported.

## Data block

<u>BYTE</u>	<u>CONTENTS</u>	<u>MEANING</u>
00	00	Record size, high byte
01	00	Record size, second byte
02	01	Record size, third byte
03	00	Record size, low byte
04	00	First record offset (must be zero for the ACB-4000A)
05	00	First record offset (must be zero for the ACB-4000A)
06	00	First record offset (must be zero for the ACB-4000A)
07	00	First record offset (must be zero for the ACB-4000A)
08	00	Number of records, high byte
09	00	Number of records, second byte
10	98	Number of records, third byte
11	FF	Number of records, low byte
12	01	Search argument length, high byte
13	06	Search argument length, low byte
14	00	Search field displacement, byte 1
15	00	Search field displacement, byte 2
16	00	Search field displacement, byte 3
17	00	Search field displacement, byte 4
18	01	Pattern length, high byte
19	00	Pattern length, low byte
20	6C	Data pattern, first byte
21	6C	Data pattern, second byte
.		
.		
.		
22	6C	Data pattern, last byte (256 for this example)

See Section 5.4.6 for details.

The COMPLETE STATUS BYTE will return an EQUAL status when data is matched. The logical block address will be valid and can be used in the TRANSLATE command to determine the physical location of the defect. The use of the TRANSLATE command is described in the next example.

### G.2 TRANSLATE, FORMAT AND INTERLEAVED OPERATION

When the manufacturer's defect list is not available or when the disk has been in use and a defect is grown, the following steps should be performed in order to map out all known media defects:

- 1) Backup the entire disk if any valuable data is contained on the disk.



- 2) Determine where all media defects are by physical location. If defects are identified by a logical location, the TRANSLATE command must be used to determine the sector's physical location. The TRANSLATE command will return the physical cylinder, head and the number of bytes from index to the defect. The number of bytes from index is given as SECTOR NUMBER x TOTAL BYTES PER SECTOR + 150 (see Section 2 for the total bytes per sector).

If the TRANSLATE command fails due to the error being in an ID field, the TRANSLATE command must be performed on the physical sector prior to, and the physical sector following, the desired sector. The physical sector desired can then be determined. This exercise becomes more difficult when an interleave value other than one is used. See the examples which follow for an example of interleaved and non-interleaved drives.

- 3) Send the MODE SELECT command. This comand specifies the format parameters and should always precede the FORMAT command. See the previous examples for a demonstration of this command.
- 4) Send the FORMAT command with an appended defect map, not greater than 1024 bytes in length, with all defects listed in ascending order. This defect map must be of the format specified in Section 5 for FORMAT command data bytes.

The "bytes from index" value in the appended defect map indicates the number of bytes from index to where the defect is located. The entire sector in which the defect resides will then be marked as unusable.

When the disk is being formatted for the first time, the sequence of operations is somewhat simplified. In this case, the method of mapping out defective sectors using the Adaptec defect handling would be as follows:

Issue the TRANSLATE command to determine the physical location of a defective sector when the interleave is equal to one. An example of an interleave value of one with 256-byte sectors follows:

```
P - 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18
F - 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18
```

```
P - 19 20 21 22 23 24 25 26 27 28 29 30 31
F - 19 20 21 22 23 24 25 26 27 28 29 30 31
```

P = Physical sector count  
F = Formatted sector location

For the purposes of this example, we will assume there is a media defect in physical sector eight, head seven of cylinder 19. The equivalent logical block address is calculated by the following algorithm:

$$\text{Logical Address} = (\text{Sctrs/Trk})(\text{hds})(\text{Cyl \#}) + (\text{Sctrs/Trk})(\text{head \#}) + \text{Sctr \#}$$

Sctrs/Trk = # of data sectors per track(32 for this example)  
hds = # of data heads on the drive (one relative)  
Cyl # = cylinder # (0 relative)  
Sctr# = sector # (0 relative)  
Head # = data head #, 0 relative

For this example: (32)(8)(19) + (32)(7) + 8 = 5096d =13E8h

<u>BYTE</u>	<u>CONTENTS</u>	<u>MEANING</u>
00	0F	TRANSLATE
01	00	High byte of Logical Block Address
02	13	Middle byte of Logical Block Address
03	E8	Low byte of Logical Block Address
04	00	Reserved
05	00	Reserved

After formatting the drive with the defective sector mapped out and the interleave of one, the sector numbers would appear as follows:

```
P - 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18
F - 00 01 02 03 04 05 06 07 D 08 09 10 11 12 13 14 15 16 17

P - 19 20 21 22 23 24 25 26 27 28 29 30 31
F - 18 19 20 21 22 23 24 25 26 27 28 29 30
```

D = Defect location, sector not used.

Now an example using an interleave factor of three with 256-byte sectors. The formatted sector location would appear as follows:

```
P - 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18
F - 00 11 22 01 12 23 02 13 24 03 14 25 04 15 26 05 16 27 06

P - 19 20 21 22 23 24 25 26 27 28 29 30 31 32*
F - 17 28 07 18 29 08 19 30 09 20 31 10 21 32
```

In this example, we will again assume a media defect in the same physical location.

For this example: (33)(8)(19) + (33)(7) +24 = 5271d = 1497h

<u>BYTE</u>	<u>CONTENT</u>	<u>MEANING</u>
00	0F	TRANSLATE
01	00	High byte of Logical Block Address
02	14	Middle byte of Logical Block Address
03	97	Low byte of Logical Block Address
04	00	Reserved
05	00	Reserved

After reformatting the drive with an interleave of three, and a bad block map which will map out physical sector eight of head seven, cylinder 19, the track format will be:

```
P - 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18
F - 00 11 22 01 12 23 02 13 D 03 14 24 04 15 25 05 16 26 06

P - 19 20 21 22 23 24 25 26 27 28 29 30 31 32*
F - 17 27 07 18 28 08 19 29 09 20 31 10 21 31
```

\* Notice the one extra sector per track assigned when the interleave value is any value other than one.

### G.3 CREATING A DEFECT LIST FROM A DRIVE USING TRANSLATE

A defect list can be created by using the TRANSLATE command in the following manner:

- 1) Format the drive with a known interleave (preferably one).
- 2) Perform a test on the drive media. This may be done through use of the VERIFY, WRITE AND VERIFY or DATA SEARCH EQUAL commands. The test may also be performed by any other media test method which provides a physical or logical defect location.
- 3) Use the TRANSLATE command to determine the actual physical locations of the media defects. The translate command gives the physical cylinder, head and bytes from index information required for the defect map. This information will be used in the FORMAT command as the appended defect list.
- 4) When the TRANSLATE command fails due to a bad ID field, the following algorithm may be used to determine the defective cylinder, head, and byte displacement.

#### DEFINITION OF VARIABLES

N- Logical sector number where the translate command fails  
BD- Byte displacement from index to the bad sector  
CYL- Cylinder number containing the bad sector  
HD- Head number of the bad sector  
C\_HD- Number of heads per cylinder  
D\_BD- Defect descriptor, bytes from index  
D\_CYL- Defect descriptor, cylinder number  
D\_HD- Defect descriptor, head number  
S(0; K-1)-Array of Sector ID's for K sectors per track

K=	33	for	blocksize	of	256	with	interleave	
	32	"	"	"	256	w/o	"	
	18	"	"	"	512	with	"	
	17	"	"	"	512	w/o	"	
	9	"	"	"	1024	with or without	interleave	

For the ACB-4070 2,7 RLL controller

K=	47	for	blocksize	of	256	with	interleave	
	46	"	"	"	256	w/o	"	
	26	"	"	"	512	with	"	
	25	"	"	"	512	w/o	"	
	14	"	"	"	1024	with or without	interleave	

R- Recording size of a sector

```

R=  310 for blocksize of 256 with interleave
    320 "      "      " 256 w/o      "
    566 "      "      " 512 with     "
    576 "      "      " 512 w/o      "
   1078 "      "      " 1024 with     "
   1088 "      "      " 1024 w/o      "

```

For the ACB-4070 2,7 RLL Controller

```

R=  323 for blocksize of 256 with interleave
    333 "      "      " 256 w/o      "
    579 "      "      " 512 with     "
    589 "      "      " 512 w/o      "
   1091 "      "      " 1024 with     "
   1101 "      "      " 1024 w/o      "

```

D(0;W-1)- Array of defect descriptor used in format. Each entry has three elements: D\_CYL, D\_HD, & D\_BD  
W- Total number of defects.  
I,J,L- work variables

STEP 1: INITIALIZATION

```

CYL, HD, L = 0;
INITIALIZE R & K according to block size
Read the first defect descriptor to set D_CYL, D_HD, & D_BD
If there is no defect descriptor then set D_CYL to 64000

```

STEP 2: FIND THE DEFECTIVE SECTORS ON A TRACK

```

a.  S(0; K-1) = 0, I = K      /* Initialize Sector Map */
/* check if the defect is at the current track */
b.  If D_CYL > CYL then goto STEP 3
c.  If D_HD > HD then goto STEP 3
/* the defect descriptor is pointing to current track */
d.  J = D_DB / R      /* defective sector is after the */
    If J >= K then goto step 2.f      /* last sector */
e.  If S(J) = 0 then
    S(J) = 80, I = I - 1 /* designates a bad sector*/
    Else /*in this case, the sector is already marked bad*/
f.  Get the next defect descriptor
    If there are no more defect descriptors, set D_CYL to
    64000.
g.  Goto STEP 2.b

```

STEP 3: FILL IN THE GOOD SECTORS FOR A TRACK

```
a. M = 0
b. For J = 1 to I      /* I has the # of good sectors */
    /* 0 denotes an available sector */
    If S(M) = 0 then
        S(M) = 80      /* mark the sector as used */
        if L = N then /* found the desired sector */
            Goto STEP 4
        L = L + 1      /* inc the logical sector # */
        M = M + interleave factor
                    /* next physical sector */
    Else /* Sector used, try the next one */
        M = M + 1

        /* start the sector number over */
        If M >= K then M = M - K

    /* End of For loop */

c. HD = HD + 1 /* try the next track */
d. If HD = C_HD, then HD = 0, CYL = CYL + 1
e. Goto STEP 2
```

STEP 4: FIND THE BYTE DISPLACEMENT

```
a. BD = M * R + 150 /* M is passed in from step 3 */
b. BD and CYL and HD (passed in from step 3) give the
    defect descriptor of the bad sector where the bad ID
    error occurred.
```

This algorithm assumes the defect descriptors are sorted into ascending order and that there is at least one good sector per track.

#### G.4 WRITE PROTECTING A DRIVE AND RECOVERY FROM BLOWN FORMAT

Hard disk usage in some environments may necessitate the need to write protect a disk drive. This is easily accomplished with Adaptec controllers by sending the MODE SELECT command.

The MODE SELECT command is useful in three situations. The first is to enable device independence by following MODE SELECT with a FORMAT command. The second situation is when one wishes to write protect a drive. The third use for the MODE SELECT command is to recover data on a disk with a blown format. This may be used when there is a problem with the format on track 0 causing the drive parameters to be returned incorrectly or not at all. The MODE SELECT will then reinitialize the controller enabling the ability to read data therefore, backing up the disk. When the MODE SELECT command is not followed by a FORMAT command, it is necessary to write protect the disk in case the MODE SELECT command did not send the correct disk parameters. When the drive

is write protected through the MODE SELECT command, the drive will remain write protected until power to the controller is cycled.

#### G.5 USING SEND DIAGNOSTICS COMMAND TO TURN OFF ECC AND RETRIES

This command sends data to the controller to specify diagnostic tests for the controller and drive.

The following example sets the error handling options for no retries or error correction. This error handling scheme is recommended only for testing a drive.

<u>BYTE</u>	<u>CONTENT</u>	<u>MEANING</u>
00	1D	SEND DIAGNOSTIC
01	00	Logical unit 0, 20 for drive 1
02	00	Reserved
03	00	Data length
04	04	Data length
05	00	Reserved

Bytes 3 and 4 give the data length. The data length must be at least four bytes long. For the Patch Ram and Patch Hardware Area, the data length must be equal to the length of the data block to be passed. If the length specified is longer than needed, the excess is ignored and not read.

The following example gives the data bytes required to change the error handling option.

<u>BYTE</u>	<u>CONTENT</u>	<u>MEANING</u>
00	65	Diagnostic specifier (set read error handling options)
01	00	Diagnostic Option or Coded Release Level
02	01	Set read error handling option
03	00	Patch data length or reserved

Byte 0 is the Diagnostic Specifier. The supported functions are:

- 60H - Indicates Re-initialize drive
- 61H - Dump Hardware Area (4000-40ff)
- 62H - Dump RAM (8000-80ff)
- 63H - Patch Hardware Area
- 64H - Patch RAM
- 65H - Set Read Error Handling Options

Functions 60-64 are reserved for vendor field support.

Byte 1 provides a safety mechanism for the patch options (63H, 64H) to prevent inadvertent patches.

Byte 2 specifies the low byte of the starting patch address for options 63H and 64H. For option 65H, byte 2 determines the read error handling option. These options, once set, stay in effect until the next reset, START UNIT command, RE-INITIALIZE option or SET READ ERROR HANDLING OPTIONS. A controller reset, START UNIT or RE-INITIALIZE selects the default state which is the same as option 0.

The read error handling options are:

00 - Correctable errors will be corrected and no check status will be issued. Uncorrectable errors will be transferred and the transfer halted with check status set with an error code of 91H.

The 00 option is the default operation and is the mode of operation which should be used under normal operating conditions. This option provides for invisible retries and correction, except for the case that the error was uncorrectable, therefore, the host must be notified.

01 - No retries or error correction. An ECC error will halt the operation and check status will be issued by the controller. The error code will always be a 98H.

The 01 option should be selected when testing a drive for quality of media or locating media defects to provide a defect list for the FORMAT command. It should be noted in this case especially, that ECC errors use the same error handling scheme for both READ commands and the VERIFY commands.

02 - Correctable errors will be corrected and data transferred, but the operation will be terminated and a check status sent. The subsequent REQUEST SENSE will give a 98H error code. If the error is not correctable, the controller will transfer the uncorrected data halt the operation, and set check status with an error code of 91H.

The 02 option should be used only when the user wishes to halt on any ECC error (including correctable errors). This option performs four retries for any ECC error.

Adaptec controllers use an intelligent retry scheme which will improve system performance as well as data reliability. When an ECC error occurs, the error syndrome will always be checked after the first retry. Oftentimes the syndrome will be the same on the retry as on the original read. When the syndrome coincides on two consecutive reads, error correction will occur immediately, rather than waiting for the full number of retries to occur.

Byte 3 is reserved and must be zero for all SEND DIAGNOSTICS options except the patch options (63H and 64H) in which this byte specifies the number of bytes to be overwritten.



## APPENDIX H: ACB-4000A TO ACB-4070 CONVERSION

---

In order to replace an ACB-4000, ACB-4010 or ACB-4000A controller with an ACB-4070, the following considerations must be addressed. Note that the ACB-4070 accepts all commands, without change, that the ACB-4000A accepts. If the same drive and host adapter are used, reformatting on the ACB-4070 is all that is required.

### H.1 DRIVE CAPABILITY

The ACB-4000A controller uses MFM (Modified Frequency Modulation) encoding on ST506/412 drives. This method has the following characteristics:

- 1) 5 Megabit per second (5.0 MHz) transfer rate on the ST506/412 interface.
- 2) 100 nanosecond data window (+/- 50 ns from center).

The ACB-4070 controller uses 2,7 RLL (Run Length Limited) encoding of ST506/412 drives. This method has the following characteristics:

- 1) 7.5 Megabit per second (7.5 MHz) transfer rate on the ST506/412 interface.
- 2) 66.6 nanosecond data window (+/- 33.3 ns from center).

These characteristics place different requirements on the disk drive.

- 1) The drives read/write channel must be able to handle the different frequency range.
- 2) The tolerances contributed to the data window by the drive must be tighter, typically +/- 27 ns.

Not all drives meet these requirements. Based on Adaptec's experience in testing drives, many drives that use thin film plated and sputtered media meet these requirements. Most drives that use oxide media do not.

CONSULT YOUR DRIVE VENDOR TO DETERMINE IF THEIR DRIVE MEETS THESE REQUIREMENTS AND SUPPORT USING 2,7 RLL.

## H.2 HOST ADAPTER TRANSFER RATE

The ACB-4000A controller accepts MFM data from the drive at a rate of 5MHz or a sector burst rate of 625 Kbytes per second. This is the rate that data can be put into the data buffer. This rate, plus the interleave factor, determines the rate that the host can take data from the controller.

The ACB-4070 controller accepts 2,7 RLL data from the drive at a rate of 7.5 Mhz or a sector burst rate of 900 Kbytes per second. This increase in speed allows a faster rate that the host can take data from the controller. The host adapter can be improved to accept this faster transfer rate and take full advantage of the added speed. If the host adapter is not upgraded, the interleave factor should be changed to compensate for the faster transfer rate.

One example is to run the ACB-4000A at a 2-to-1 interleave. If replaced with an ACB-4070 and the same host adapter, the interleave factor may be changed to 3-to-1. 2-to-1 may lose revolutions of the disk since the host may not be able to take data from the buffer quickly enough.

## H.3 DEFECT HANDLING

Drive vendors usually give a defect list for the drive in bytes from index. These are MFM bytes. When using the ACB-4070, these defects must be converted to 2,7 RLL bytes.

$$\text{MFM BYTES FROM INDEX} \times 1.5 = \text{2,7 RLL BYTES FROM INDEX}$$

If the result is a fraction, round up to the next integer, e.g. 7.5 rounds up to 8. This 2,7 RLL information must be used at format time to determine the location of defects on the disk. If using the TRANSLATE command to determine the location of defects on the disk, the bytes from index value returned is in 2,7 RLL encoding.



**adaptec, inc.**

580 Cottonwood Drive, Milpitas, California 95035 Telephone (408) 946-8600

400003-00A/AB/SB/BOFORS Printed in U.S.A.