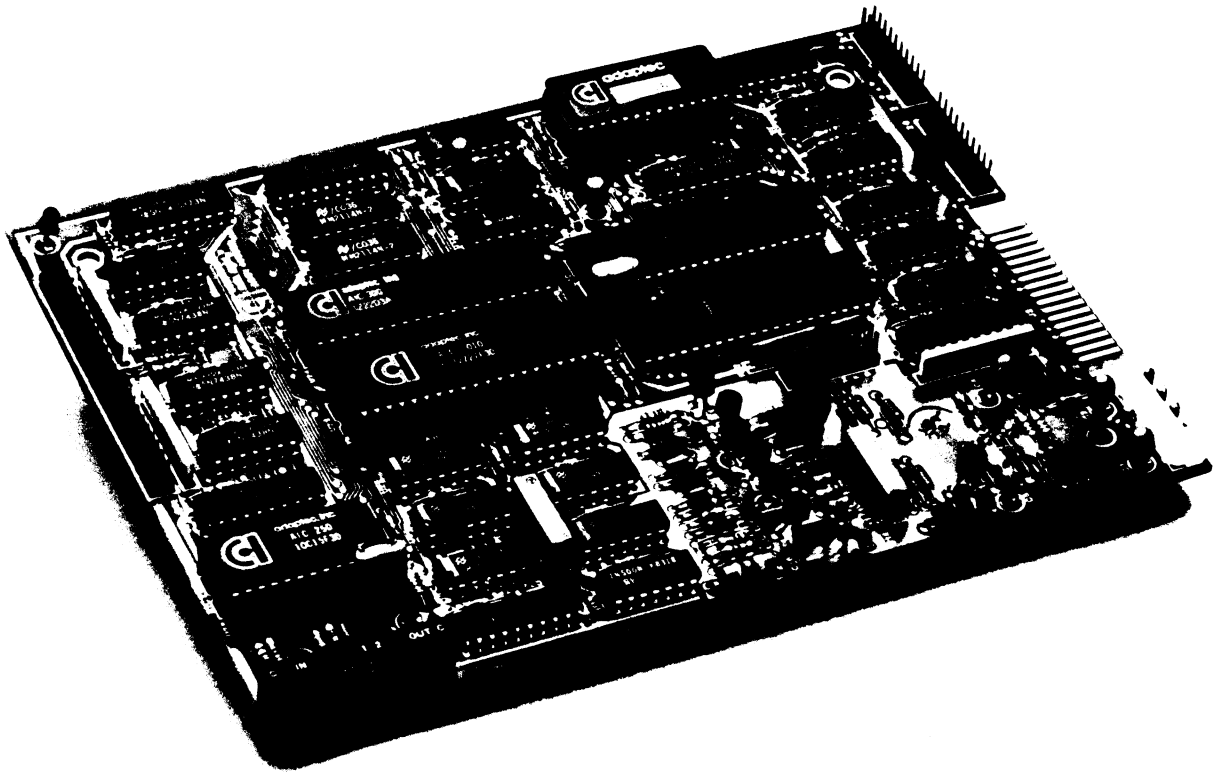
ADAPTEC

ACB-4000 WINCHESTER DISK CONTROLLER

OEM MANUAL

Adaptec, Inc. 580 Cottonwood Drive Milpitas California 95035

PRODUCT PICTURE



ADAPTEC

ACB-4000 SERIES DISK CONTROLLERS

OEM MANUAL

1.0 INTRODUCTION

The ADAPTEC 4000 Series Disk Controller boards are products which interface Winchester disk drives to any ANSC X3T9.2 SCSI (Small Computer System Interface) standard host adapter interface.

1.1 BASIC DESCRIPTION

The 4000 series supports standard SCSI features plus extensions and controls two Seagate ST-506/412 or equivalent Winchester drives. Most of the currently available drives are supported through the Mode Select Command.

1.2 FEATURE SET

A) All ADAPTEC controllers have a 1KByte FIFO data buffer which is dual ported for rapid data transfers. No sector interleaving is required.

B) Controllers offer complete host software device independence.

C) Disk defect handling is on a sector level and is transparent to the host. No spare tracks are required, therefore this disk space may be used for data. Long seeks to alternate tracks are eliminated.

D) All controllers utilize a 32 bit ECC and provide correction of single burst errors of 8 bits. All ID and data fields are ECC protected.

E) ADAPTEC controllers use logical sector addressing and variable sector lengths are programmable at format time.

F) High speed data search is supported.

G) The ACB-4000 controllers support multiple host and multiple controller systems, but are not arbitrating SCSI bus devices.

2.Ø PHYSICAL SPECIFICATIONS

2.1 SIZE

Length	7.75 inches (19.7 cm)
Width	5.75 inches (14.6 cm)
Height	.75 inches (1.9 cm)
Weight	1 pound, with packaging

2.2 POWER REQUIREMENTS

+5VDC ± 5% at 1.5 Amps (Max)
+12VDC ± 1Ø% at 3ØØ mA (Max)

Power is applied through J3, 4 pin AMP connector. The recommended mating connector is AMP P/N 1-48Ø424-Ø. J3 pins are numbered as shown in Figure 2-1.

+ 12 V
+ 12 V return
+ 5 V return
+ 5 V

Figure 2-1. Connector J3 Pin Assignments

2.3 ENVIRONMENTAL LIMITS

	Operating	Storage
Temperature F/C	32/Ø to 131/55	-4Ø/-4Ø to 167/75
Humidity(non-cond)	1Ø% to 95%	1Ø% to 95%
Altitude, ft.	Sea Level to 1Ø,ØØØ	Sea Level to 2Ø,ØØØ

3.Ø HOST AND DRIVE INTERFACES

3.1 HOST ADAPTER INTERFACE - Signals

The ADAPTEC 4ØØØ Series controllers interface to a host adapter according to the proposed ANSC X3T9.2 Standard (SCSI). The data bus is a bidirectional 8 bit parallel interface.

A 5Ø pin flat ribbon connector is provided at J4. The 3M P/N 3425-3ØØØ cable connector is recommended.

Single ended drivers and receivers allow a maximum cable length of 2Ø feet (6 meters) between the host adapter and the controller. All signals are low true. All odd pins are grounded. Figure 3-1 shows the SCSI bus pin assignments.

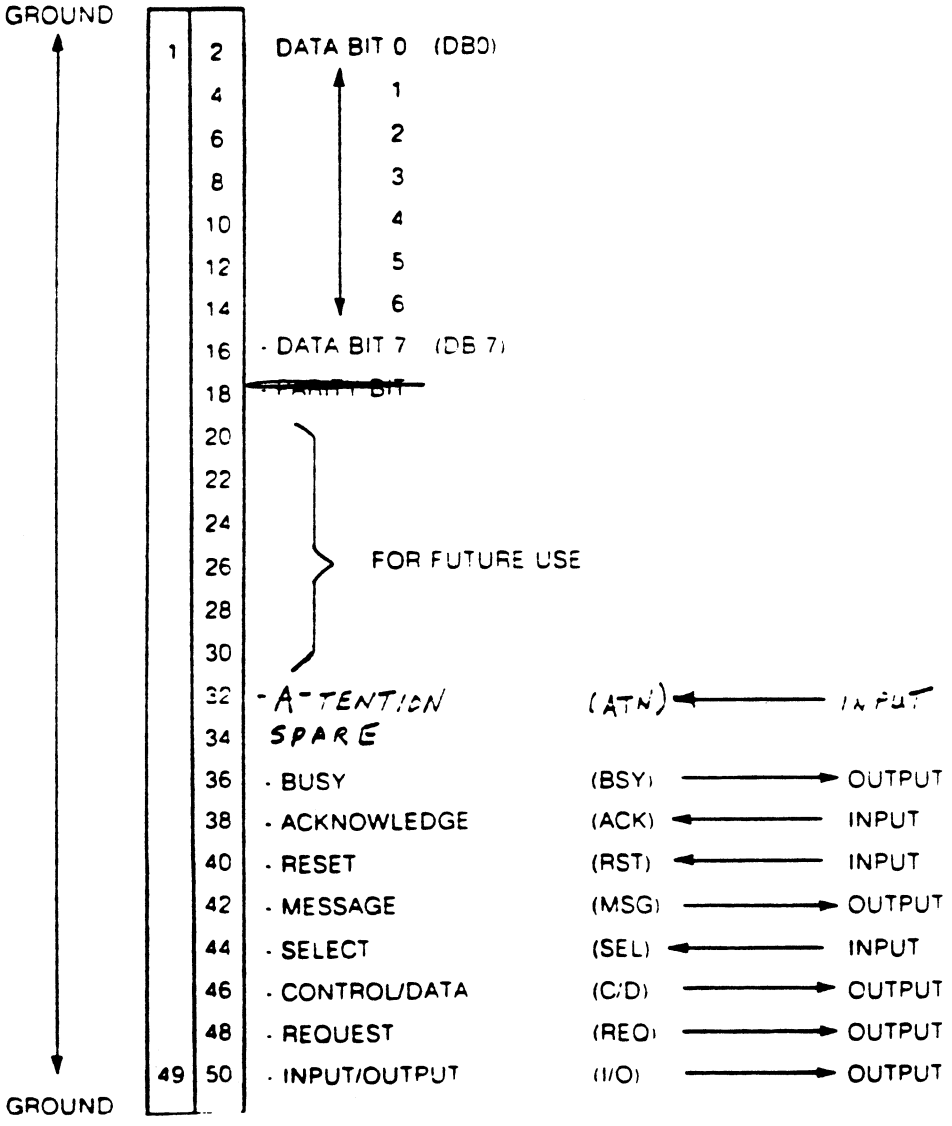
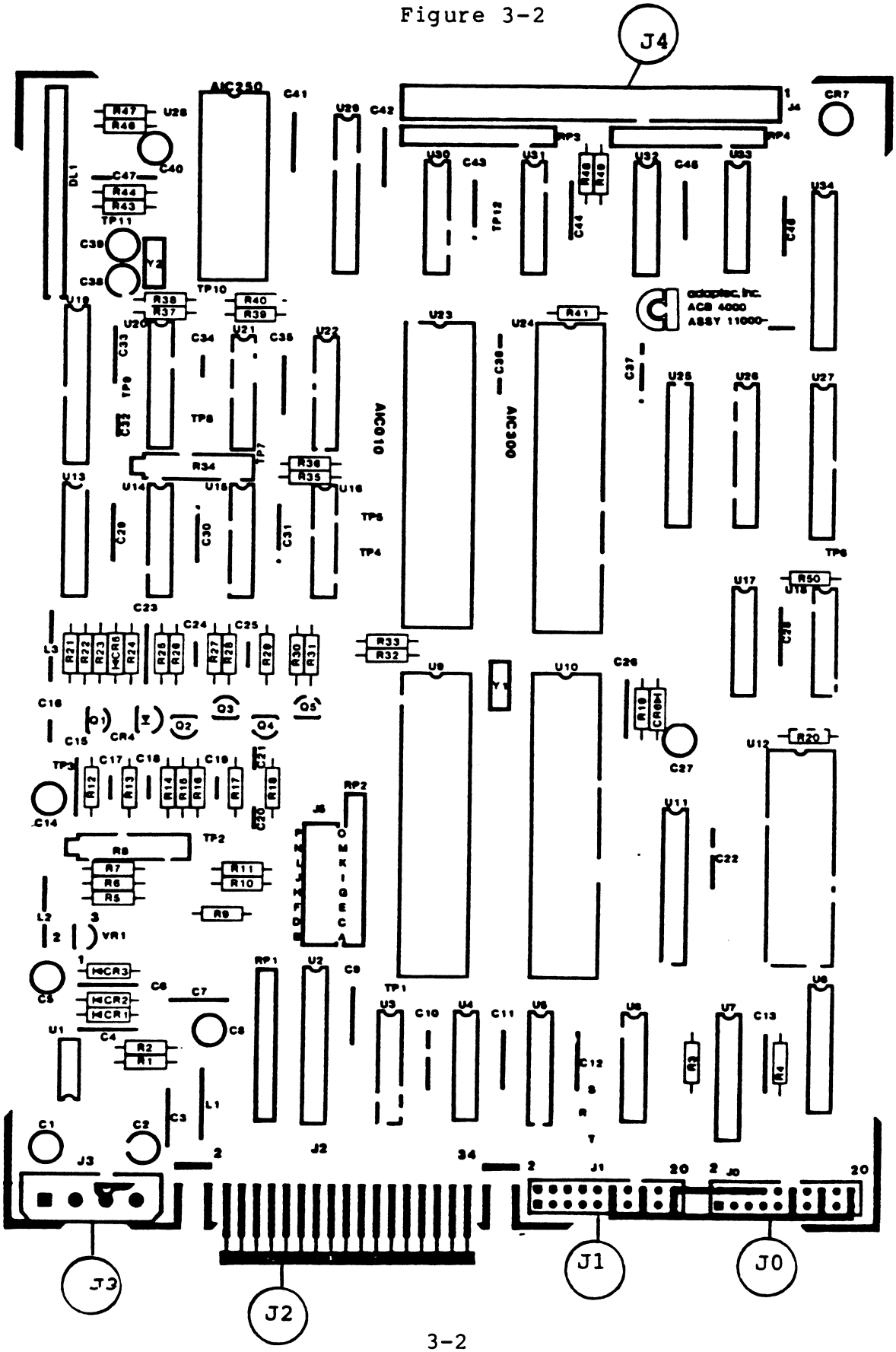


FIGURE 3-1. SCSI Bus Pin Assignments

ACB-4000 Board Layout and Connectors

Figure 3-2



3.2 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(sinking)
False = Signal Non-Assertion = 2.5 to 5.25 VDC

ADAPTEC controllers use a 7438 open collector driver to meet this specification.

Each signal from the host to the controller must have the following characteristics:

True = Signal Assertion = 0.0 to 0.8 VDC @ .4 mA (max)
False = Signal Non-Assertion = 2.0 to 5.25 VDC

A 74LS14 receiver with hysteresis meets this specification.

Figure 3-3 shows an example of proper bus termination.

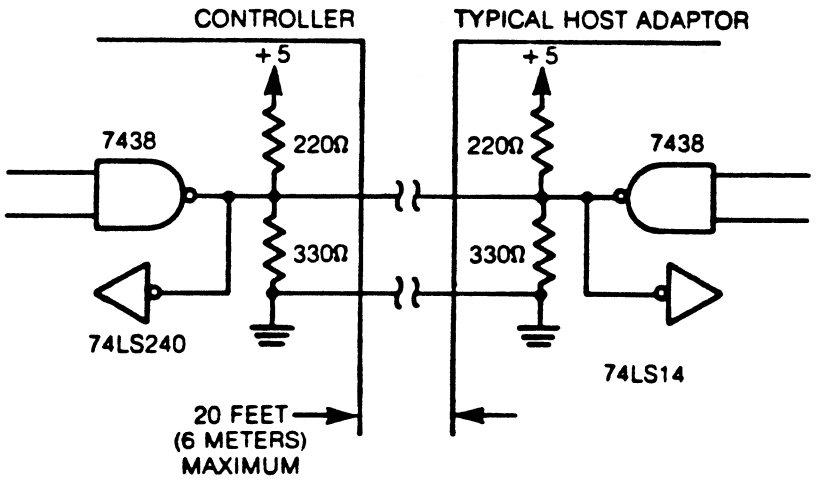


Figure 3-3. Host Adapter Bus Termination

3.3 DISK DRIVE INTERFACE - Signals

ACB-4000 controllers comply with the standard ST-506/412 interface.

A system interconnect diagram is shown in Figure 3-4. Board layout for connector positioning is shown in Figure 3-2.

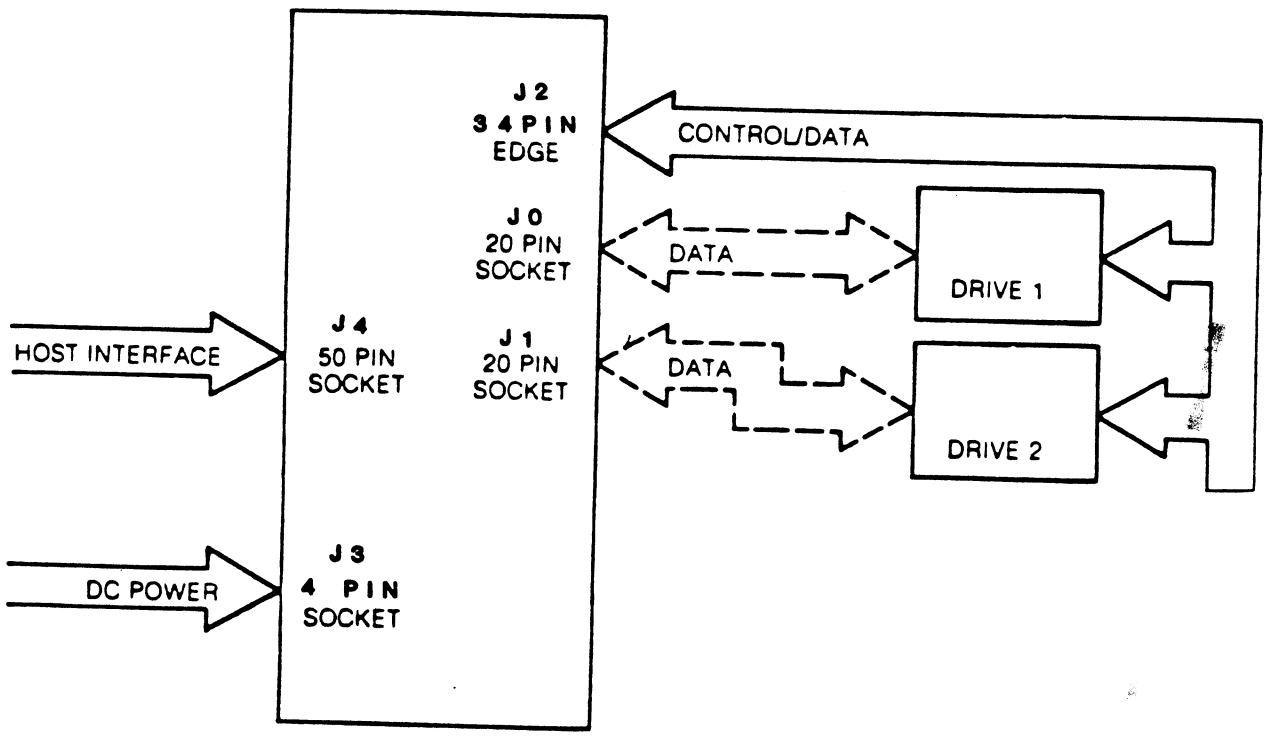


Figure 3-4. System Interconnect Diagram

3.3.1 DISK DRIVE CONNECTIONS

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 3402-0000.

The pins are numbered 1 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 (J6). Table 3-1 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 (6 meters). Suggested mating sockets for these connectors is 3M P/N 3421 Series. Table 3-2 shows J0 and J1 data bus pin assignments.

Table 3-1 CONNECTOR PIN ASSIGNMENT

GND RTN PIN	SIGNAL PIN	SIGNAL NAME
1	2	-RED WR CUR/HD 2[3]
3	4	-HEAD SELECT 2[2]
5	6	-WRITE GATE
7	8	-SEEK COMPLETE
9	10	-TRACK 0
11	12	-WRITE FAULT
13	14	-HEAD SELECT 2[0]
15	16	RESERVED
17	18	-HEAD SELECT 21
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

Table 3-2 -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
	14	- MFM WRITE DATA
16	15	GND
	17	+ MFM READ DATA
	18	- MFM READ DATA
20	19	GND

3.4 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 3-5.

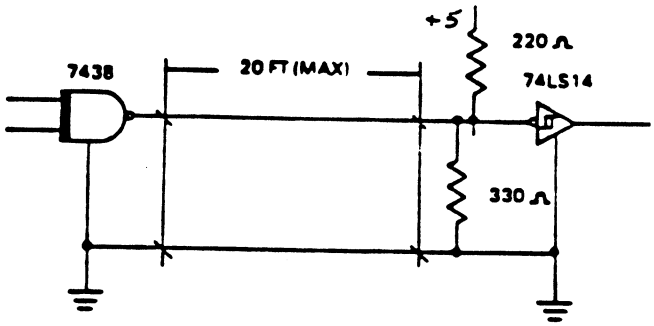


Figure 3-5. Control Driver/Receiver Lines

The control signals are specified at:

True = Ø.Ø VDC to Ø.4 VDC @ 1 = -48 mA (Max)
False = 2.5 VDC to 5.25 VDC @ 1 = +25Ø uA (Open Collector)

The read and write MFM data lines are differential signals, present on connectors JØ and J1. The ADAPTEC receiver/driver pairs meet the required RS 422 specifications. Figure 3-6 shows these lines for the ACB-4ØØØ.

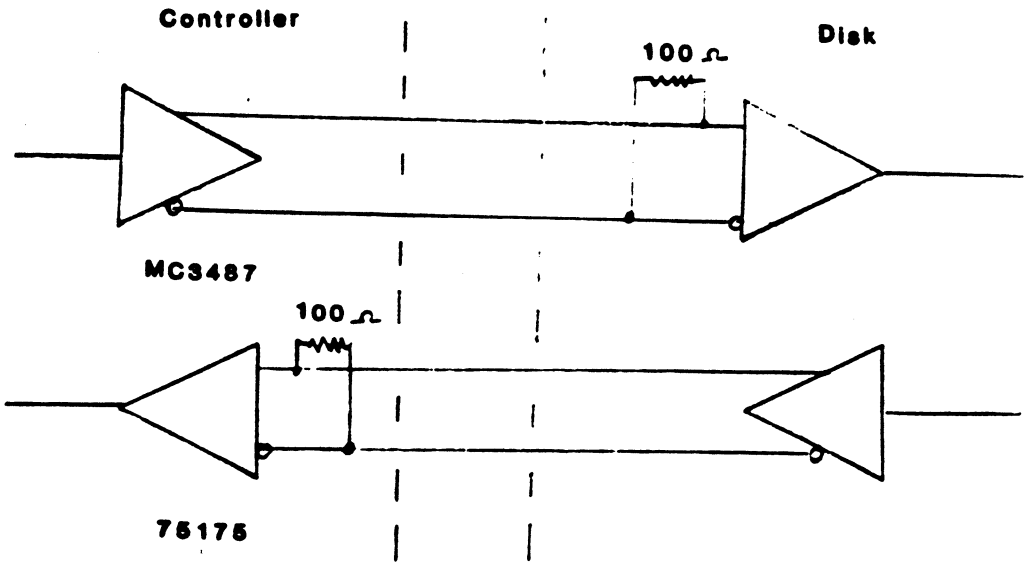


Figure 3-6 ACB 4ØØØ Data Receiver/Driver Pairs

4.0 HOST INTERFACE PROTOCOL

4.1 ACB-4000 SCSI FEATURES

This section describes in detail the SCSI protocol with the extensions which are implemented in the ADAPTEC ACB-4000 controller. The main deviations from the SCSI Standard are that the ACB-4000 does not support command chaining or disconnection and reconnection. When designing systems for the 4000 series, you may go directly from the "bus free" phase to the "selection phase" in the protocol description. The ADAPTEC ACB-5000 series controllers support the full ANSI SCSI protocol, while the ACB-4000 is designed for single host environments.

4.2 GENERAL DESCRIPTION OF SCSI

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

* Single or multiple host system.
* Multiple peripheral devices and 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.

Communication on the bus is allowed between two bus ports at a time. A maximum of eight (8) 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 as an INITIATOR and the other acts as a TARGET. 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.

An INITIATOR may address up to two (2) peripheral I/O devices that are connected to an ACB-4000 TARGET. The TARGET will provide a "virtual controller" for each of these devices, appearing to the system as two separate controller/device pairs.

Certain bus functions are assigned to the INITIATOR and certain bus functions are assigned to the TARGET. The INITIATOR may arbitrate for the bus and select a particular TARGET. The TARGET may request the transfer of COMMAND, DATA, STATUS or other information on the bus.

Data transfers on the bus are asynchronous and follow a defined REQUEST/ACKNOWLEDGE HANDSHAKE protocol. One eight-bit byte of information may be transferred with each handshake.

4.3 BUS SIGNALS

The 9 control signals and 8 data signals are described below:

4.3.1 BUSY (BSY)

BSY is an "or-tied" signal which indicates that the bus is in use.

4.3.2. SELECT (SEL)

SEL is an "or-tied" signal used by an INITIATOR to select a TARGET or by a TARGET to reselect an INITIATOR.

4.3.3. CONTROL/DATA (C/D)

C/D is a TARGET-driven signal to indicate whether CONTROL or DATA information is on the data bus. Assertion indicates CONTROL.

4.3.4. INPUT/OUTPUT (I/O)

I/O is a TARGET-driven signal which controls the direction of data movement on the data bus relative to an INITIATOR. Assertion indicates INPUT to the INITIATOR.

4.3.5. MESSAGE (MSG)

MSG is a TARGET-driven signal indicating the MESSAGE phase.

4.3.6. REQUEST (REQ)

REQ is a TARGET-driven signal indicating a request for a REQ/ACK data transfer handshake.

4.3.7. ACKNOWLEDGE (ACK)

ACK is an INITIATOR-driven signal indicating acknowledgment of a REQ/ACK data transfer handshake.

4.3.8. ATTENTION (ATN)

ATN is an INITIATOR-driven signal indicating the ATTENTION condition.

4.3.9. RESET (RST)

RST is an "or-tied" signal indicating the RESET condition.

4.3.10. 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. 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 I.D.) 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 I.D.) and leaves the other data bits in the passive (non-driven) state.

4.4 BUS PHASES

The bus has eight (8) distinct operational phases and cannot be in more than one phase simultaneously.

* BUS FREE PHASE
* SELECTION PHASE
* INFORMATION TRANSFER PHASES
* COMMAND PHASE
* DATA PHASES (DATA IN/OUT)
* STATUS PHASE

4.4.1. 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 (simulataneously within a DESKEW DELAY) and the RESET condition is not active.

4.4.2. 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 I.D. and (optionally) its own I.D. After two DESKEW DELAYS, the INITIATOR can assert SEL.

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

4.4.3. INFORMATION TRANSFER PHASES

The COMMAND, DATA, STATUS and MESSAGE phases are all used to transfer data or control information through the DATA bus. The actual contents of the information is beyond the scope of this section.

The C/D, I/O and MSG signals are used to differentiate the various INFORMATION TRANSFER phases. Note that these signals are not valid without REQ asserted. See Table 4-1.

TABLE 4-1: INFORMATION TRANSFER PHASES

SIGNAL				DIRECTION OF
MSG	C/D	I/O	PHASE NAME	INFORMATION XFER
Ø	Ø	Ø	DATA OUT PHASE	(INIT to TARG)
Ø	Ø	1	DATA IN PHASE	(INIT from TARG)
Ø	1	Ø	COMMAND PHASE	(INIT to TARG)
Ø	1	1	STATUS PHASE	(INIT from TARG)
1	Ø	Ø	* Not Used	
1	Ø	1	* Not Used	
1	1	Ø	MSG OUT PHASE	(INIT to TARG)
1	1	1	MSG IN PHASE	(INIT from TARG)

Notes: Ø = SIGNAL DEASSERTION
1 = SIGNAL ASSERTION
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 ensure 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.

4.4.3. INFORMATION TRANSFER PHASES (Continued)

With the I/O signal not asserted, data will be output from the INITIATOR to the TARGET. The INITIATOR must ensure 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 a 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.

4.4.3.1. 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.

4.4.3.2. DATA PHASES (DATA IN/DATA OUT)

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.

4.4.3.3. 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.

4.4.3.4. MESSAGE PHASES (MESSAGE IN/MESSAGE OUT)

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.

4.4.4. 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 descriptions for the ATTENTION and RESET conditions.

4.5. BUS CONDITIONS

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

4.5.1. ATTENTION CONDITION

ATTENTION allows the INITIATOR signal the TARGET of a waiting IDENTIFY MESSAGE. The TARGET may access the messsage 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.

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 handshake of a MESSAGE OUT phase.

4.5.2. 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 as defined in the controller specification.

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. TARGETS 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.

4.6. 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. (Any other phase can also 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).

4.9. TIMING

A timing chart is provided in Figure 4-2. 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.

* ABORTED SELECTION 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.

* BUS CLEAR DELAY: 650 nanoseconds (maximum)

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

* BUS SET DELAY: 1.1 microseconds

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

* BUS SETTLE DELAY: 450 nanoseconds (minimum)

* CABLE SKEW: 10 nanoseconds (maximum)

The maximum difference in propogation time allowed between any two bus signals when measured between any two bus ports.

* DESKEW DELAY: 45 nanoseconds (minimum)

* REQ RESPONSE TIMEOUT: 250 milliseconds

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

* RESET HOLD TIME: 25 microseconds (minimum)

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

* SELECT TIMEOUT: 250 milliseconds

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

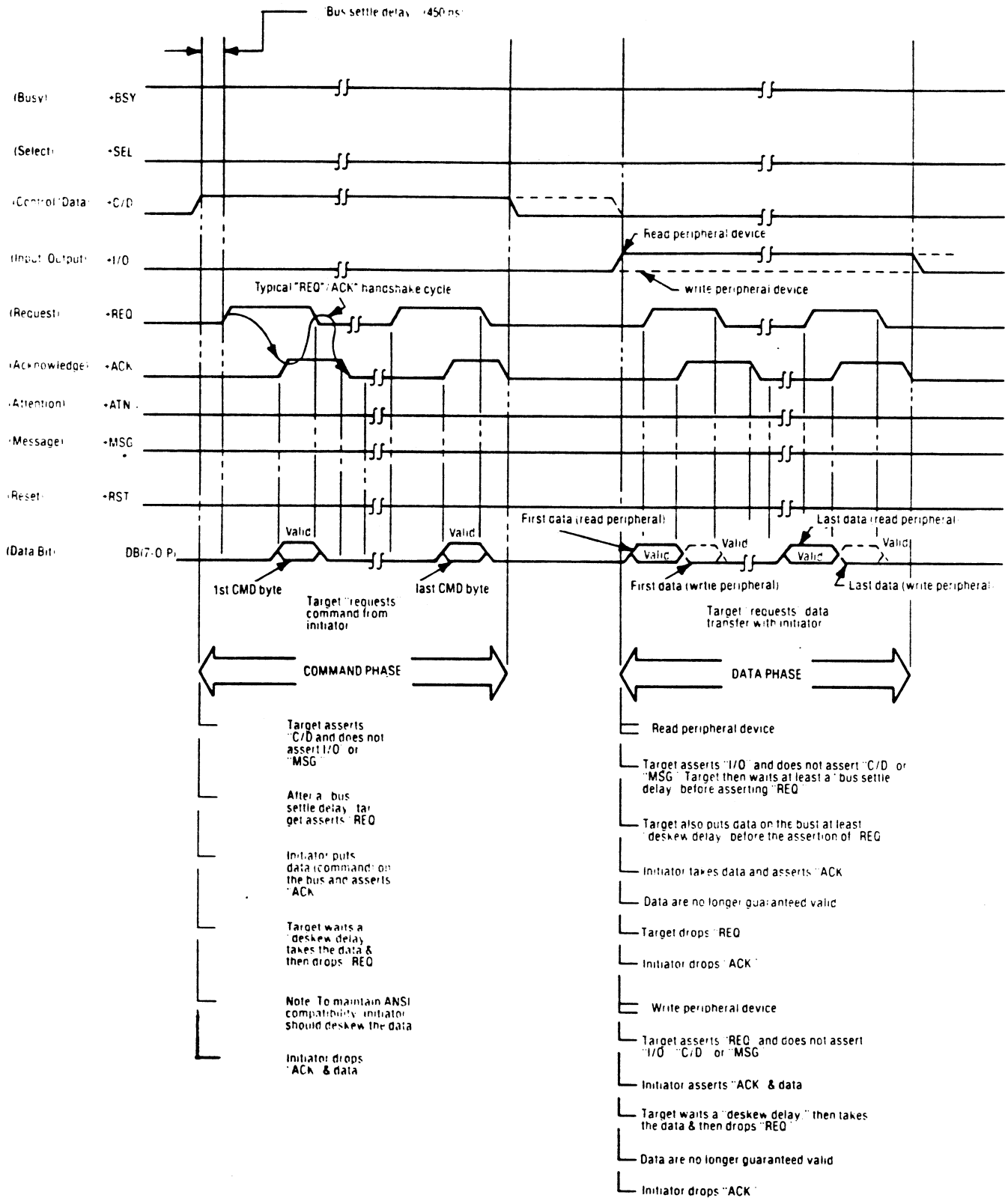
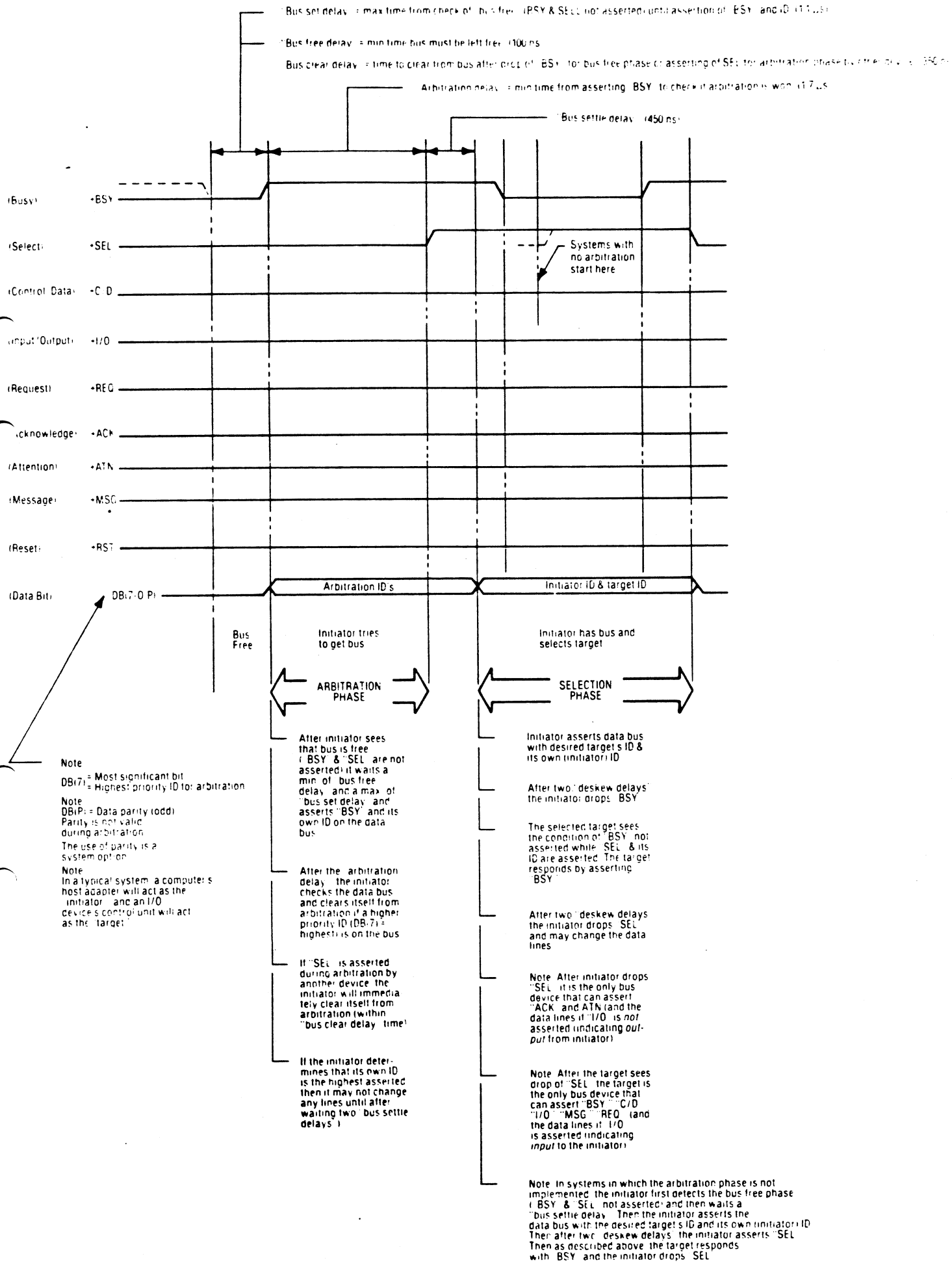
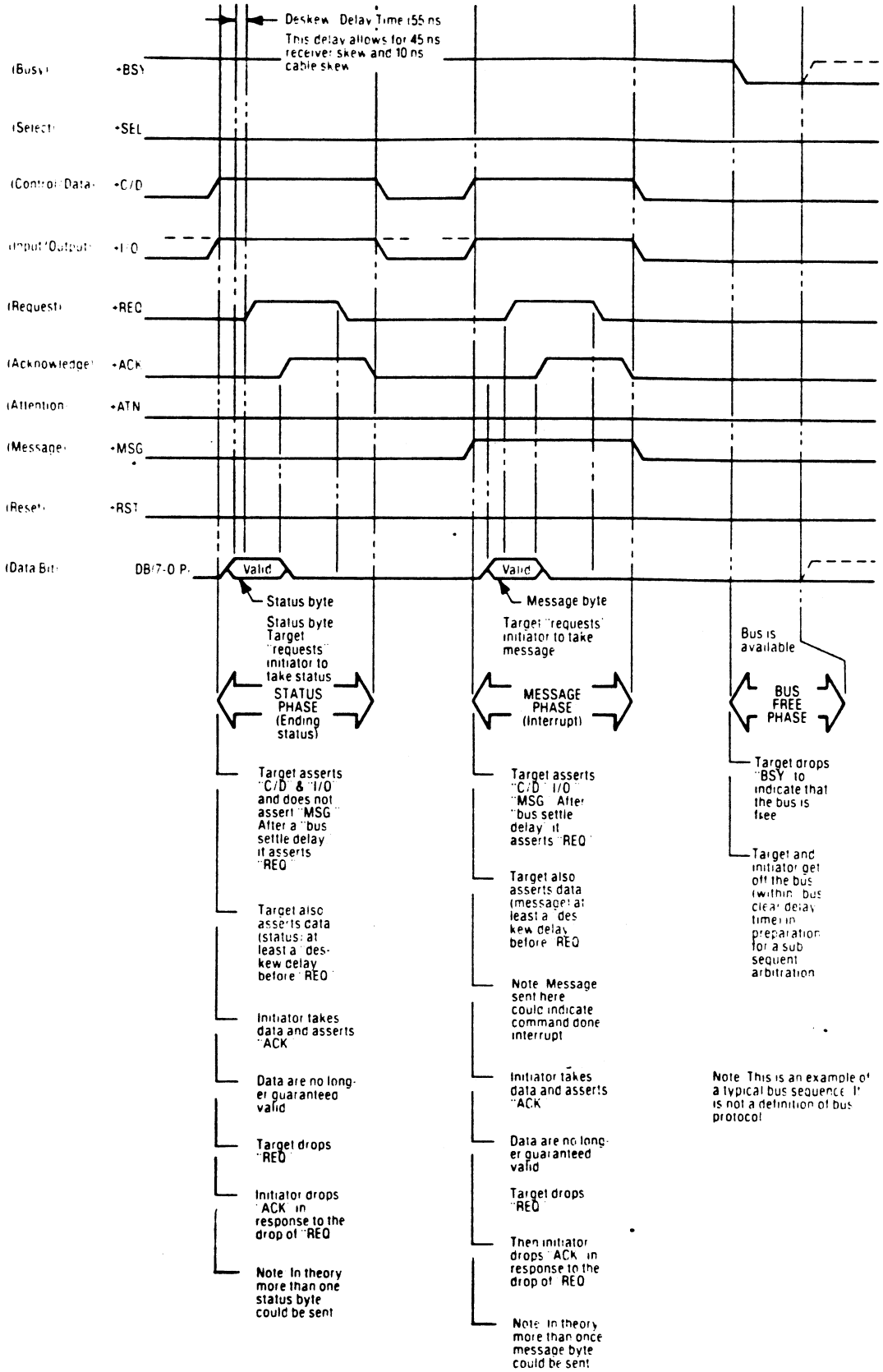


Figure 4-2
Bus Timing Chart
4-10


"Bus set delay" = max time from check of "bus free" ("BSY" & "SEL" not asserted) until assertion of "BSY" and ID
"Bus free delay" = min time bus must be left free (100 ns)
Bus clear delay = time to clear from bus after drop of "BSY" for bus free phase or asserting of "SEL" for arbitration phase
Arbitration delay = min time from asserting "BSY" to check if arbitration is won (1.7 µs)
"Bus settle delay" (450 ns)
(Busy) +BSY
(Select) +SEL
(Control Data) +C/D
(Input/Output) +I/O
(Request) +REQ
(Acknowledge) +ACK
(Attention) +ATN
(Message) +MSG
(Reset) +RST
(Data Bit) DB(7-0,P)
Systems with no arbitration start here
Arbitration ID's
Initiator ID & target ID
Bus Free
Initiator tries to get bus
Initiator has bus and selects target
ARBITRATION PHASE
SELECTION PHASE
Note
DB(7) = Most significant bit = Highest priority ID for arbitration
Note
DB(P) = Data parity (odd) Parity is not valid during arbitration
The use of parity is a system option
Note
In a typical system a computer's host adapter will act as the "initiator" and an I/O device's control unit will act as the "target"
After initiator sees that bus is free ("BSY" & "SEL" are not asserted) it waits a min. of "bus free delay" and a max. of "bus set delay" and asserts "BSY" and its own ID on the data bus
After the "arbitration delay" the initiator checks the data bus and clears itself from arbitration if a higher priority ID (DB-7) is highest ID on the bus
If "SEL" is asserted during arbitration by another device the initiator will immediately clear itself from arbitration (within "bus clear delay" time)
If the initiator determines that its own ID is the highest asserted then it may not change any lines until after waiting two "bus settle delays"
Initiator asserts data bus with desired target's ID & its own (initiator) ID
After two "deskew delays" the initiator drops "BSY"
The selected target sees the condition of "BSY" not asserted while "SEL" & its ID are asserted. The target responds by asserting "BSY"
After two "deskew delays" the initiator drops "SEL" and may change the data lines
Note: After initiator drops "SEL" it is the only bus device that can assert "ACK" and ATN (and the data lines if "I/O" is *not* asserted indicating *output* from initiator)
Note: After the target sees drop of "SEL" the target is the only bus device that can assert "BSY" "C/D" "I/O" "MSG" "REQ" (and the data lines if "I/O" is asserted indicating *input* to the initiator)
Note: In systems in which the arbitration phase is not implemented, the initiator first detects the bus free phase ("BSY" & "SEL" not asserted) and then waits a "bus settle delay". Then the initiator asserts the data bus with the desired target's ID and its own (initiator) ID. Then after two "deskew delays" the initiator asserts "SEL". Then as described above, the target responds with "BSY" and the initiator drops "SEL"


Deskew Delay Time (55 ns
This delay allows for 45 ns receiver skew and 10 ns cable skew
(Busy) -BSY
(Select) -SEL
(Control/Data) -C/D
(Input/Output) -I/O
(Request) -REQ
(Acknowledge) -ACK
(Attention) -ATN
(Message) -MSG
(Reset) -RST
(Data Bit) DB(7-0,P)
Valid
Valid
Status byte
Message byte
Status byte Target "requests" initiator to take status
Target "requests" initiator to take message
Bus is available
STATUS PHASE (Ending status)
MESSAGE PHASE (Interrupt)
BUS FREE PHASE
Target asserts "C/D" & "I/O" and does not assert "MSG" After a "bus settle delay" it asserts "REQ"
Target also asserts data (status) at least a "des-kew delay" before "REQ"
Initiator takes data and asserts "ACK"
Data are no longer guaranteed valid
Target drops "REQ"
Initiator drops "ACK" in response to the drop of "REQ"
Note In theory more than one status byte could be sent
Target asserts "C/D" "I/O" "MSG" After "bus settle delay" it asserts "REQ"
Target also asserts data (message) at least a "des-kew delay" before "REQ"
Note Message sent here could indicate command done interrupt
Initiator takes data and asserts "ACK"
Data are no longer guaranteed valid
Target drops "REQ"
Then initiator drops "ACK" in response to the drop of "REQ"
Note In theory more than once message byte could be sent
Target drops "BSY" to indicate that the bus is free
Target and initiator get off the bus (within "bus clear" delay time) in preparation for a subsequent arbitration
Note This is an example of a typical bus sequence It is not a definition of bus protocol

5.0 MESSAGE SPECIFICATION

5.1 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-4000 controllers support the COMMAND COMPLETE, MESSAGE REJECT and IDENTIFY messages and does not respond to the ATN signal except during selection. Only COMMAND COMPLETE need be implemented in a ACB-4000 environment.

5.1.1. SINGLE BYTE MESSAGES

Command Complete (00H)

This code is sent from the TARGET, at the completion of command execution (or at the end of a series of linked commands), to direct the INITIATOR to indicate COMMAND COMPLETE to the host.

This message does not imply good ending status; STATUS must be checked to determine end conditions.

Message Reject (07H)

This code is sent from the INITIATOR or TARGET if the message received was inappropriate or not implemented.

The INITIATOR will assert the ATN signal prior to its release of ACK for the REQ/ACK handshake of the message that will be rejected. When the TARGET sends this message, it will change to MESSAGE IN Phase and send this MESSAGE prior to requesting additional message bytes.

Identify (80 TO FF)

This code is sent by either the INITIATOR or TARGET to establish the physical path connection between the INITIATOR and TARGET for a particular LUN. Initiators signal thier support for the message system by asserting ATTENTION during the selection phase. The ACB-4000 expects an IDENTIFY message in this case.

If an IDENTIFY message is received the LUN specified there is used in lieu of the LUN field (Byte 01) of the command.

Bit-7 is always set to identify this message.

Bit-6 is set by the INITIATOR to indicate its capability to accommodate disconnection and reconnection. (Not used in ACB-4000.)

Bits-5, 4, and 3 are reserved.

Bits-2, 1, and 0 specify a LUN address in a TARGET.

6.0 COMMAND SPECIFICATIONS

6.1 GENERAL DESCRIPTION

This section of the ADAPTEC Controller Manual includes the software command set and the specific status information related to the commands.

By defining a fixed block structure using a simple, logical address scheme, the I/O interface can support device independence. In addition, by including the logical block address as a component of the command structure, physical requirements (such as SEEK) can be imbedded within the basic READ and WRITE requests.

This interface, despite its simplicity, is capable of providing the high level of performance required in multi-host/multi-task environments. Powerful functions, such as search, are included to enhance random access applications, and single-command, multi-block transfers are included to simplify sequential operations.

The ACB-4000 series controllers support a majority of the proposed ANSC SCSI command set. It is important to note that all ADAPTEC controllers require that reserved bit and byte positions in commands be zero. Commands which violate this standard will be rejected. Therefore, as a rule, all reserved and vendor unique portions of commands should be zero unless their use is specifically stated in this document.

6.2 COMMAND AND STATUS STRUCTURE

6.2.1. COMMAND DESCRIPTION BLOCK (CDB)

An I/O request to a device is made by passing a Command Description 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 in ADAPTEC controllers:

Class Ø: 6-Byte commands

Class 1: 1Ø-Byte commands.

Tables 6-1 and 6-2 show typical command descriptor block formats.

Table 6-1: CLASS ØØ COMMANDS (6-BYTE COMMANDS)
(Such as READ or WRITE)

BYTE \ BIT	7	6	5	4	3	2	1	Ø
ØØ	Class Code				Opcode			
Ø1	Logical Unit Number			(MSB) Logical Block Address				
Ø2	Logical Block Address							
Ø3	Logical Block Address							(LSB)
Ø4	Number of Blocks							
Ø5*	Reserved (Ø)							

* Control Byte

Table 6-2: CLASS Ø1 COMMANDS (1Ø BYTE EXTENDED BLOCK ADDRESS)

BYTE	BIT 7	6	5	4	3	2	1	Ø
ØØ	Class Code				Opcode			
Ø1	Logical Unit Number				Command Specific Bits			
Ø2	(MSB) Logical Block Address							
Ø3	Logical Block Address							
Ø4	Logical Block Address							
Ø5	Logical Block Address							(LSB)
Ø6	Reserved							
Ø7	Number of Blocks							
Ø8	Number of Blocks							
Ø9*	Reserved (Ø)							

*Control Byte

6.2.2. CLASS CODE

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

6.2.3. OPERATION CODE

The operation code for each class allows 32 commands (ØØ to $1F_H$).

6.2.4. LOGICAL UNIT NUMBER

Logical unit numbers allow 8 devices per Controller. The ACB-4ØØØ accomodates 2 devices per Controller which must be devices Ø and 1.

6.2.5. COMMAND SPECIFIC BITS

Byte Ø1, bits Ø1 - Ø4 specify options which depend upon the particular command.

6.2.6. LOGICAL BLOCK ADDRESS

Class Ø commands contain 21 bit starting block addresses while Class 1 supports 32 bit block addressing.

The "block" concept implies that the Host and Controller have "preset" the number of bytes of data to be transferred. You will note that the concept of sector is replaced by block.

6.2.7. NUMBER OF BLOCKS

A variable number of blocks may be transferred under a single command. Class ØØ commands may transfer up to 256 blocks, while Class Ø1 commands may transfer up to 64K blocks. A zero block number count defaults to the maximum value.

6.2.8. CONTROL BYTE (Last Byte in All Commands)

All bits in the control byte are reserved and must be zero.

6.3 COMMAND DESCRIPTIONS

The following section describes the complete command set and associated formats for ACB-4000 controllers. In most cases, ADAPTEC has followed the proposed ANSI SCSI command specifications to the letter, deviating only in degree of implementation.

6.3.1. CLASS 00 COMMAND DESCRIPTIONS

The following is a series of command descriptions.

Table 6-5: CLASS 00 COMMAND CODE SUMMARY

OP CODE	COMMAND	OP CODE	COMMAND
00	TEST UNIT READY	0F	TRANSLATE
01	REZERO UNIT	13	WRITE BUFFER
03	REQUEST SENSE	14	READ BUFFER
04	FORMAT UNIT	15	MODE SELECT
08	READ	1A	MODE SENSE
0A	WRITE	1B	START/STOP UNIT
0B	SEEK	1C	RECEIVE DIAGNOSTIC
		1D	SEND DIAGNOSTIC

TEST UNIT READY (00_H)

BYTE \ BIT	7	6	5	4	3	2	1	0
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)							

* Control Byte

This command returns zero status if the requested unit is powered on and ready. If not ready, a check condition will be set in the status byte. Possible errors are Drive Not Ready (04_H) and Write Fault (03_H). This is not a request for self-test.

REZERO UNIT (01_H)

BYTE	BIT 7	6	5	4	3	2	1	0
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)							

* Control Byte

This command sets the selected drive to track zero and then sends completion status. Possible error returns are No Seek Complete (02_H), Drive Not Ready (04_H) and No Track Zero (06_H).

REQUEST SENSE (03_H)

See Paragraph 6.4 for details of the complete command as well as a complete discussion of returned sense data.

FORMAT UNIT (04_H)

The control unit will write from index to index all ID and DATA fields with a block size as specified by an immediately previous MODE SELECT command. If no MODE SELECT command has been executed, the previous data block size will be used. On unformatted disks or those whose format is determined bad (sense byte error code $1C_H$ returned following a READ), a MODE SELECT command is required prior to the format command. Data fields are completely written with $6C_H$ unless otherwise specified in the format command.

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)							

* Control Byte.

The ID fields will be interleaved as specified in bytes 3 and 4 of the CDB (byte 4, bit 0 LSB). Under normal conditions, ADAPTEC controllers do not require interleaving because of their high speed buffer control. An interleave number of 1 results in sequential ID fields being written on the disk. Any interleave number greater than 1 and one less than the total sectors per track result in interleaved formatting. A 0 in this field will cause the default interleave factor of 2 to be used. By using an interleave of 2, ADAPTEC controllers can format 33 256-byte sectors per track rather than the normal 32 sectors. (See Appendix A for details) Note that byte 3 must always be zero and also that the value in byte 4 must not exceed the number of sectors per track minus one. An error code of 24_H (Bad Argument) will be returned if either of these rules are violated.

The interleave number is equivalent to the number of disk revolutions required to sequentially read one track. An example of an interleave number of 3 follows:

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

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

where P=physical sectoring and F=new formatted addresses.

Bits 0 through 4 of byte 1 in the CDB specify the format of the bad block list for defect skipping.

When the Data Bit (04) is set, the controller expects a list of known bad areas in the data portion of the command. If this bit is zero, the defect list is not read and defect skipping is not performed.

Bit 03 is the Complete List bit and specifies that all of the known defects on the drive are contained in the list. The list itself must be less than 1024 bytes since it must fit in the available buffer space.

Bit 02 of Byte 01, if set, indicates that the next two bits (Byte 01, Bits 01 and 00) will be used to define the format. A zero indicates default. The next Format List bit (Byte 01, Bit 01) if set indicates that the data pattern in Byte 02 is to be used to format. A zero indicates default. A zero in bit 00 indicates that a Cylinder /Head/Byte Count format is used in the data list. The following table defines the use of the Data and List Format bits:

Fmt Data	Bit 02	Bit 00	Definition
0	0	0	Format with no user-supplied error information.
1	1	0	Error information is in Cyl., Head and Displacement format.

All other combinations of these bits will be rejected.

Bit 01	Definition
0	Format with default fill byte ($6C_H$)
1	Use format command byte 02 for fill data.

The following is the defect list format supported by ADAPTEC controllers. The list includes the physical coordinates of known media flaws in assending order of cylinder, head, and bytes from index. All defects must be listed in ascending order when presented to the controller.

FORMAT DATA (BYTE FORMAT)

BYTE	BIT 07	06	05	04	03	02	01	00
00	Reserved							
01	Reserved							
02	Length of							
03	Defect List in Bytes (8N)							
04	(MSB) Cylinder Number of Defect #1							
05	Cylinder Number of Defect #1							
06	Cylinder Number of Defect #1							(LSB)
07	Head Number of Defect #1							
08	(MSB) Bytes From Index							
09	Bytes From Index							
10	Bytes From Index							
11	Bytes From Index							(LSB)
.								
.								
.								
8N-4 to 8N+3	Nth Defect							

See example in Appendix A.

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

If, in time, other defects appear on a drive, the contents of the entire drive should be backed up and a new format operation performed. To identify the physical locations of the troublesome blocks use the TRANSLATE command. The new defect locations must then be added and sorted into the complete list.

ADAPTEC's defect skipping technique is at the sector level and does not require time-consuming seeks to spare track locations. Therefore, the tracks specified by a drive manufacturer as "spare" may be utilized for data, increasing the effective capacity of the device.

READ (08_H)

BYTE	BIT 7	6	5	4	3	2	1	0
00	0	0	0	0	1	0	0	0
01	Logical Unit Number			(MSB) Logical Block Address				
02	Logical Block Address							
03	Logical Block Address							(LSB)
04	Number of Blocks							
05*	Reserved (0)							

* Control Byte

This command transfers (to the Host) the specified number of blocks starting at the specified logical starting 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 field 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 re-read up to 5 times to establish a solid error syndrome. Only then is correction attempted. Correction is done directly into the data buffer, transparent to the host.

WRITE ($0A_H$)

BYTE	BIT 7	6	5	4	3	2	1	0
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	Number of Blocks							
05*	Reserved (0)							

* Control Byte

This command transfers (to the Target Device) the specified number of blocks starting 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.

(ADAPTEC controllers also support corresponding extended READ and WRITE commands using the Class 01 CDB format.)

SEEK ($0B_H$)

BYTE	BIT 7	6	5	4	3	2	1	0
00	0	0	0	0	1	0	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)							

* Control Byte

This command causes the selected drive to seek to the specified starting address. The ACB-4000 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. When the seek is complete, the controller reconnects to the host and responds with completion status.

All ACB products use an implied seek on READ, WRITE and SEARCH commands obviating the need for issuance of SEEK commands with each operation.

TRANSLATE ($0F_H$)

BYTE	BIT 7	6	5	4	3	2	1	0
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)							

* Control Byte

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.

WRITE DATA BUFFER (13_H)

BYTE	BIT 7	6	5	4	3	2	1	0
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)							

* Control Byte

This command serves buffer RAM diagnostic purposes. The controller will fill the buffer with 1K bytes of data from the host. There is no guarantee that this data will not be overwritten by other operations initiated by other INITIATORS.

READ BUFFER RAM (14_H)

BYTE	BIT 7	6	5	4	3	2	1	0
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)							

* Control Byte

Read Buffer will pass the host 1K of data from the buffer. It is intended for RAM diagnostic purposes. The same caveat applies to this as to write buffer. In addition, although data remains in the buffer after normal data operations the ordering of the data found there is undefined.

MODE SELECT (15_H)

This command is used in ACB controllers to specify FORMAT parameters and should always preceed 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. Then the drive should be backed up and reformatted.

MODE SELECT COMMAND

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

* Control Byte

Byte 4 of the command specifies the number of information bytes to be passed with the command. A minimum of twelve bytes ($0C_H$) must be specified. If drive parameters are being specified the count should be 22 bytes (16_H).

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

MODE SELECT PARAMETER LIST

BYTE	BIT 07	06	05	04	03	02	01	00
00	Reserved							
01	Reserved							
02	Reserved							
03	Length of Extent Descriptor List = 08_H							

EXTENT DESCRIPTOR LIST

BYTE	BIT 07	06	05	04	03	02	01	00
00	Density Code							
01	Reserved							
02	Reserved							
03	Reserved							
04	Reserved							
05	(MSB) Block Size							
06	Block Size							
07	Block Size							(LSB)

Byte 0 of the extent descriptor list specifies the data density of the drive. Current ACB products support only MFM and a value of 00 in this byte is required. Bytes 1, 2 and 3 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.

The extent descriptor list and following drive parameter list are a single large data block which follows the command.

The ACB-4000 must be set up with a value 256, 512 or 1024 bytes.

Any violation of the above constraints will result in Check Status with a Error Code of 24_H, indicating an invalid argument in parameter data.

DRIVE PARAMETER LIST

BYTE	BIT 07	06	05	04	03	02	01	00
00	List Format Code == 01							
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							

The Drive Parameter list includes all the data necessary to specify a drive. It is optional, but if present must be complete 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.

The Cylinder Count is the number of data cylinders on the drive. Due to the in-line defect skipping formatting cylinders normally set aside as spares may be included in this total. The minimum is one. 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 1. The minimum is 1; maximum is 16. A drive with 9 or more heads will use the reduced write current line as the high order head select. The default value is 2.

The Reduced Write Current Cylinder is the cylinder number beyond which the controller will assert the reduced write current line. Minimum value is Ø; maximum is 2Ø47. The default value is cylinder 15Ø. Note that reduced write current assumes a different meaning on drives with more than 8 heads.

The Write Precompensation Cylinder is the cylinder beyond which the controller will compensate for inner track bit shift. The specs for this function agree with those of most disk manufacturers. Minimum value is Ø; maximum is 2Ø47.

> NOTE: On the ACB-4ØØØ 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 for those drives with more than 8 heads 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. (For Maxtor drives set the jumper to the always off position. See Appendix A.)
>
> For drives which do not require reduced write current or write precompensation, the user must specify the maximum cylinder address +1 in these two parameters to prevent the controller from asserting the reduced write current signal.

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 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:

```
ØØ == Non Buffered Seek -- 3.Ø mS rate -- ST-5Ø6
Ø1 == Buffered Seek -- 28 uS rate -- ST-412
Ø2 == Buffered Seek -- 12 uS rate
```

MODE SENSE ($1A_H$)

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

MODE SENSE 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				
02	Reserved (0)							
03	Reserved							
04	Number of Bytes Returned							
05*	Reserved							

* Control Byte

Byte 4 of the command specifies the number of data bytes to be returned from the command. A minimum of 12 bytes ($0C_H$) must be specified. If the drive parameter list is required, the count should be 22 bytes (16_H).

The returned 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.

START/STOP UNIT ($1B_H$)

Byte Ø4, bit ØØ of this command should be set if this is a START command, otherwise it is 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.

BYTE	BIT 07	06	05	04	03	02	01	00
ØØ	Ø	Ø	Ø	1	1	Ø	1	1
Ø1	Logical Unit Number			Reserved (Ø)				
Ø2	Reserved							
Ø3	Reserved							
Ø4	Reserved							St/Stp
Ø5*	Reserved							

* Control Byte

RECEIVE DIAGNOSTIC RESULT ($1C_H$)

BYTE	BIT 7	6	5	4	3	2	1	0
ØØ	Ø	Ø	Ø	1	1	1	Ø	Ø
Ø1	Logical Unit Number			Reserved (Ø)				
Ø2	Reserved (Ø)							
Ø3	(MSB) Data Length							
Ø4	Data Length							(LSB)
Ø5*	Reserved (Ø)							

* Control Byte

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

READ 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_H or more, although, 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 follows:

BYTE	BIT 7	6	5	4	3	2	1	0
ØØ	(MSB) Data Block Length (=Ø1Ø4$_H$)							
Ø1	Data Block Length							(LSB)
Ø2	(MSB) Starting Address of Dump							
Ø3	Starting Address of Dump							
Ø4	Dumped Data (xxØØ)							
1Ø3	Dumped Data (xxFF)							

SEND DIAGNOSTIC ($1D_H$)

BYTE	BIT 7	6	5	4	3	2	1	0
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)							

* Control Byte

This command sends data to the Controller to specify diagnostic 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 4 bytes long and should be qual 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 read.

The first byte of the data block specifies the particular function being requested. The options available at this time, along with their associated codes are:

60_H -- Re-initialize Drive
61_H -- Dump Hardware Area (4000-40FF)
62_H -- Dump RAM (8000-80FF)
63_H -- Patch Hardware Area
64_H -- Patch RAM
65_H -- Set Read Error Handling Options

Of these options, only the patch options require a data block longer than 4 bytes.

The second byte specifies a subtest or qualifiers specific to the test selected by the first byte. (Because of the potential danger in patching controller programs, this byte provides a safety mechanism to prevent obsolete patches). For further details, contact ADAPTEC Customer Support. The second byte is not checked if the 65_H option code is specified.

The third byte specifies the starting address in RAM or the memory-mapped registers to gbe patched. The high byte of the address is implicit in the diagnostic specified. Therefore, a Patch RAM operation with a third byte of $A1_H$ will overwrite an area of RAM starting with $80A1_H$.

The fourth byte gives the number of bytes to be overwritten. This can range from 1 to 256, with a zero yielding 256. The data block for the Send Diagnostic Command is as follows:

BYTE	BIT 7	6	5	4	3	2	1	0
00	Diagnostic Specifier							
01	Diagnostic Option or Coded Release Level							
02	Low Byte of Patch Starting Address or Qualifier							
03	Patch Data Length (00 = 100_H) (N) or Reserved							
04	Optional Patch Data							
	- - - - - - - - - - - - - - - -							
N+3	Optional Patch Data							

Byte 02 of the data block specifies the actions to take place upon encountering an ECC check if Option 65_H is selected. The default state is established by a controller reset. These options, once set, stay in effect until the next reset. They apply only to the LUN addressed by the command.

The Set Read Error Handling Options are:

00 -- Selects default operation where a correctable error will be corrected without comment and all data transferred without check status. If the error is not correctable, the controller will transfer the uncorrected data and set check status with an error code of 91_H. The valid address will be that of the bad block.

01 -- Report all corrections and stop. A correctable error will be corrected and the data transferred, but the operation will stop with a check status and an error of 98_H. An uncorrectable error will be handled as in 00.

02 -- Do not correct. All ECC errors will be treated as uncorrectable except that the error code is set to 98_H.

6.3.2 CLASS Ø1 COMMAND DESCRIPTIONS

Table 6-6: CLASS Ø1 COMMAND CODE SUMMARY

OP CODE	COMMAND	OP CODE	COMMAND
25	READ CAPACITY	2E	WRITE AND VERIFY
28	READ	2F	VERIFY
2A	WRITE	31	SEARCH DATA EQUAL

READ CAPACITY (25_H)

BYTE \ BIT	7	6	5	4	3	2	1	Ø
ØØ	Ø	Ø	1	Ø	Ø	1	Ø	1
Ø1	Logical Unit Number			RESERVED (Ø)				Rel Ad
Ø2	(MSB) Logical Block Address							
Ø3	Logical Block Address							
Ø4	Logical Block Address							
Ø5	Logical Block Address (LSB)							
Ø6	Reserved (Ø)							
Ø7	Reserved							
Ø8	Full or Partitial Media Indicator							
Ø9*	Reserved							

*Control Byte
If byte 8 of the CDB is $ØØ_H$, 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 byte 8 is $Ø1_H$, this command will return the address of the block (after the specified starting address) at which a substantial delay in data transfer will be encountered (e.g., a cylinder boundary). Any value other than $ØØ_H$ or $Ø1_H$ in byte Ø8 will cause Check Status with an Error code of 24_H for an invalid argument.

In both cases, the format block size is defined by the last four bytes of the 8-byte data field returned as a result:

4 Bytes - Block Address

4 Bytes - Block Size

WRITE AND VERIFY ($2E_H$)

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.

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

BYTE	BIT 7	6	5	4	3	2	1	0
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							
07	Number of Blocks							
08	Number of Blocks							
09	Reserved							

VERIFY ($2F_H$)

This command is similar to the previous WRITE AND VERIFY except that it verifies the ECC of an alreading existing set of data blocks. It is up to the Host to provide data for rewriting and correcting if an error is detected.

BYTE	BIT 7	6	5	4	3	2	1	0
00	0	0	1	0	1	1	1	1
01	Logical Unit Number			Reserved				
02	(MSB) Logical Block Address							
03	Logical Block Address							
04	Logical Block Address							
05	Logical Block Address (LSB)							
06	Reserved							
07	Number of Blocks							
08	Number of Blocks							
09	Reserved (0)							

SEARCH DATA EQUAL (31_H)

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 and the data to compare.

The Invert bit (Byte Ø1, Bit Ø4) inverts the sense of the search comparison operation. With invert on, a SEARCH DATA EQUAL command would succeed on data not equal; SEARCH DATA LOW would succeed on data greater or equal. The invert bit on the ACB-4ØØØ allows SEARCH EQUAL inverted which succeeds on the first block not equal to the pattern.

By using this command , small computer systems are given the power of large mainframes by rapidly searching for record key fields when implementing indexed access methods.

When a search is satified, it will terminate with a Condition Met Status. A Request Sense Command can then be issued to determine the block address of the matching record. A Request Sense following a successful Search Data command will:

1. Report a Sense Key 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 is reported.

2. Set the Valid bit to one.

3. Report the address of the block containing the first matching record in the Information Bytes.

The Request Sense command following an unsuccessful Search Data command will:

1. Report a Sense Key of No Sense, provided no errors occured.

2. Set the Valid bit to zero.

SEARCH DATA EQUAL COMMAND

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

The argument following a SEARCH command is as follows:

BYTE	BIT 07	06	05	04	03	02	01	00
00	(MSB)			Record Size				
01				Record Size				
02				Record Size				
03				Record Size				(LSB)
04	(MSB)			First Record Offset				
05				First Record Offset				
06				First Record Offset				
07				First Record Offset				(LSB)
08	(MSB)			Number of Records				
09				Number of Records				
10				Number of Records				
11				Number of Records				(LSB)
12	(MSB)			Search Argument Length				
13				Search Argument Length				(LSB)
14	(MSB)			Search Field Displacement				
15				Search Field Displacement				
16				Search Field Displacement				
17				Search Field Displacement				(LSB)
18	(MSB)			Pattern Length				
19				Pattern Length				(LSB)
20				Data Pattern				
.								
.								
M+19				Data Pattern				

A definition of the required data in the SEARCH argument follows:

BYTES	PARAMETER
ØØ TO Ø3	Record Size (Bytes) For all ADAPTEC controllers this must equal the blocksize or zero. Zero will be taken to mean the format blocksize.
Ø4 to Ø7	First Record Offset (Bytes) For all ADAPTEC controllers this must be zero.
Ø8 to 11	Number of Records For ADAPTEC controllers 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-4ØØØ 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 field in each record. Pattern length must equal blocksize on the 4ØØØ series controllers.
2Ø to M+19	Data Pattern A variable length field of M bytes up to blocksize - displacement bytes. The ACB-4ØØØ pattern must be one block long.

6.4 COMPLETION STATUS BYTE

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

Table 6-7: COMPLETION STATUS BYTE

BYTE	BIT 7	6	5	4	3	2	1	Ø
ØØ	Reserved				Busy	Equal	Check	Reserv

Bits Ø, 5, 6 & 7 : MUST be 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 an Host that does not allow reconnection requests an operation from a reserved or busy device.

REQUEST SENSE (03_H)

This command returns unit sense.

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. Other hosts will receive BUSY status to commands for a LUN with non-zero sense to report. 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 3 bytes will default to 4 bytes. 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				
02	Reserved							
03	Reserved							
04	Number of Bytes							
05*	Reserved (0)							

* Control Byte

6.5 SENSE BYTES: Table 6-8

BYTE \ BIT	7	6	5	4	3	2	1	0
00	AdrVal	Error Class			Error Code (See Tbls 10-12)			
01	Reserved			(MSB) Logical Block Address				
02	Logical Block Address							
03	Logical Block Address							(LSB)

NOTE: The address valid bit (byte 00, bit 07) indicates that the Logical Block address bytes contain valid information.

Table 6-10: CLASS 00 ERROR CODES IN SENSE BYTE (DRIVE ERRORS)

CODE	ERROR
00	NO SENSE
01	NO INDEX SIGNAL
02	NO SEEK COMPLETE
03	WRITE FAULT
04	DRIVE NOT READY
06	NO TRACK 00

Table 6-11: CLASS Ø1 ERROR CODES IN SENSE BYTE (TARGET ERRORS)

CODE	ERROR
1Ø	I.D. CRC ERROR
11	UNCORRECTABLE DATA ERROR
12	I.D. ADDRESS MARK NOT FOUND
13	DATA ADDRESS MARK NOT FOUND
14	RECORD NOT FOUND
15	SEEK ERROR
16-17	NOT ASSIGNED
18	DATA CHECK IN NO RETRY MODE
19	ECC ERROR DURING VERIFY
1A	INTERLEAVE ERROR
1B	NOT ASSIGNED
1C	UNFORMATTED OR BAD FORMAT ON DRIVE
1D	SELF TEST FAILED
1E	DEFECTIVE TRACK (MEDIA ERRORS)
1F	NOT ASSIGNED

Table 6-12: CLASS Ø2 ERROR CODES (SYSTEM-RELATED ERRORS)

CODE	ERROR
2Ø	INVALID COMMAND
21	ILLEGAL BLOCK ADDRESS
22	NOT ASSIGNED
23	VOLUME OVERFLOW
24	BAD ARGUMENT
25	INVALID LOGICAL UNIT NUMBER
26 - 2F	NOT ASSIGNED

APPENDIX A

INSTALLATION SECTION

&

INTRODUCTION TO SASI/SCSI

1.Ø INTRODUCTION

The SASI/ ANSI SCSI bus provides a simple means for interfacing one or more contollers to a host system. However, the SCSI spec as defined by ANSI is plagued by the problem of being so precise that it is difficult to read and understand quickly. This appplication note is designed for 2 groups of people. The first are the people who just wish to see the controller work without getting bogged down with the fine points of the spec. The second group is the user who is attempting to replace a 'SASI-like' controller (hereafter called brand-X) with a true SASI/SCSI controller (hereafter called ADAPTEC). Once the initial hurdle of getting the controller/drive combination to operate properly is overcome, it is recommended that the user go back to the SASI/SCSI document and learn the subtle points.

2.Ø THE BUS AND INTERFACE

The SASI/SCSI bus is a 5Ø pin flat cable with 18 active lines and 25 interleaved ground lines (all odd-numbered pins are ground). This is the so-called unbalanced SASI/SCSI bus. 8 of the lines are the bi-directional data lines (byte-wide), 1 is the data parity (not used on the ACB-4ØØØ), and 9 are control/status lines. The bus is open-collector driven and terminated on both ends by a 22Ø-33Ø ohm pull-up/pull-down network (22Ø ohm pull-up to +5, 33Ø ohm pull-down to ground). The bus is active-low, thus a grounded line is considered active or asserted.

The host adapter must be able to drive or receive the data on the 8 data lines, drive the ACK, SEL and RST lines and receive the C/D, I/O, MSG, BSY and REQ lines.

The ATTN line is not used by brand-X controllers and must not be used with an ADAPTEC controller unless you have an intelligent host adapter. We recommend that the ATTN line be terminated by the host adapter, but it should only be pulled low if the host adapter supports the SASI/SCSI message protocol. If the ATTN line is asserted by a host adapter unable to use the message protocol, a different command send routine is required which will cause most driver software to 'hang'.

The REQ and ACK lines form the handshake to sync the data rates of the controller and the host. These lines can be used in the program I/O mode (PIO) where the host system polls the REQ, gets or puts the data and drives the ACK line or can be used with a DMA controller (LSI or discrete) for a higher transfer rate.

The SEL and BSY lines are used mainly for initial selection (waking up) of the controller. The SEL line says 'get onto the bus' and the BSY is the 'I am on the bus' reply.

The C/D, I/O and MSG lines are status lines from the controller that indicate data direction (I/O), command or data phase (C/D) and command complete (MSG).

The final control line is the RST or reset line. This line causes the controller to abort it's current operation (if any) and get off of the bus. In addition, the ADAPTEC controller will re-read the disk parameters from the drive(s).

The SASI/SCSI spec calls for a minimum RST pulse width of 25 uS. However, some brand-X controllers will accept a very short reset pulse and users have relied upon this ability in designing host adapters with very short reset pulses. The ADAPTEC controllers can accept a 50 nS. wide pulse. For new host adapter design, it is suggested that the SASI/SCSI 25 uS. spec be followed to assure upward compatability with newer controllers.

The data parity line is not used on the ADAPTEC ACB-4000 controller and is not terminated at the controller end. Future high-performance ADAPTEC controllers will make use of the parity line so new adapter designs should consider it as an expansion option.

3.0 SETTING UP THE HARDWARE

If you are replacing a brand-X unit, the ADAPTEC controller is form and footprint compatible with the brand-X units with one exception. The SASI/SCSI connector on the ADAPTEC unit is designed for consistant cable layout (pin 1 always to the left as viewed from the board edge) and so is reversed compared to the brand-X units. Pin 1 can be identified by the square pad on the solder side of the board (closest to the LED). All other connectors are identical.

If this is a new system, you will need one 34 pin (PC finger to PC finger) control cable and a 20 pin (PC finger to header socket) data cable for each drive. The 20 pin connector marked J0 goes to drive 0 and J1 goes to drive 1.

The drives should be set up as drive 0 and 1 according to the manufacturer's instructions. If there is only 1 drive, leave the terminating resistor pack in place. If there are 2 drives, remove the terminator from the drive closest to the controller. Watch the pin-1 orientation on all cables! Do **NOT** use the 'radial' or always selected option if the drive has one.

Also, if you are using a Seagate ST506 type drive, **DO NOT ENABLE** the half step option since the controller supports only the 3 ms. step rate for un-buffered drives.

The controller requires +5 and +12 volts using the same connector and pinout as the drive. The DC should be as 'clean' as possible as noise spikes can cause soft data errors on the drive system. If your system develops a large number of soft errors, you may want to try a different or isolated supply. The ADAPTEC controller requires 1.3 amps on the +5 supply and 70 mA. on the +12 supply.

The 2 trim pots on the board are used to set the phase-locked loop (PLL) of the data separator and are factory adjusted. **Do not adjust them in the field!**

There are 2 sets of jumper plugs that option the controller to match it to the drive(s) with which it is operating.

The jumpers in position J5 set the controller address and diagnostic mode. Jumpers A-B, C-D and E-F set the controller address on the SASI/SCSI bus. An installed jumper is a 1 and a removed jumper is a 0. Jumper E-F is most significant and jumper A-B is least significant.

For a controller address of 0, all three jumpers should be removed.

The K-L and M-N jumpers are not used by the controller and should be left open.

Two DMA transfer speeds are supported on the ACB-4000. Some host adapters or DMA channels cannot support the maximum transfer rate of the controller. By setting the jumper between position G-H, the transfer rate is cut in half and runs at a rate of SYSCLOCK/4 on single sector transfers. Multisector transfers are always made at a rate of DATA CLOCK/2.

Jumper O-P sets the diagnostic mode. When installed, a reset will cause a continous self-test of the controller and drive. This option is used only for factory burn-in and should not be used in the field.

The final jumper position (near J1) sets the the pre-comp to be used for the drive. If jumpers R-S are shorted, the pre-comp starts at the reduce write current point. If jumpers R-T are shorted, pre-comp is applied to **all** tracks. If no jumper is installed, **no** tracks are pre-comped. See the drive's manual to determine the pre-comp requirements of your drive. Please note that the pre-comp applies to both drives.

4.Ø TALKING TO THE CONTROLLER

Once the controller, disk and host adapter are cabled up and ready to run, the power is applied. The controller will enter a power-up mode and wait for 4Ø seconds for the drive(s) to become ready. The 4Ø second delay is required for some of the slow power-up drives on the market. Once both drives come active or the 4Ø second timeout is over, the controller 'knows' if the drive(s) are attached or not. Once the drive(s) comes ready, the controller will read the drive parameter information stored in ID flags on track Ø. This information includes block size and drive type information.

If a SASI/SCSI reset pulse is received, the sequence is the same except the 4Ø second timeout is reduced to 2ØØ mS. as the drives are assumed to be up to speed.

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

Once back on track Ø, the parameter information stored by an ADAPTEC format is read out of track Ø, head Ø and saved in the controller. If the drive is unformatted or had been formatted by a brand-X controller, the parameter information is not present so the controller then sets a bit in it's memory called 'blown format' to warn the user that the drive is unusable. If the drive format is blown, the reset sequence is stopped at this point and the controller is ready for a command.

If the drive is correctly formatted, the controller will seek the drive to the last cylinder and read the largest block address present. This data also goes into the controller's parameter table.

Once the last block address has been read, the controller will seek the drive back to track Ø, 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 a block on the disk. Since the ADAPTEC controller does not waste an entire track (8K) for a defect like the brand-X controllers, the position of a given block on the drive can't be predicted without knowing how many 'hidden' defective blocks there are before the desired block. The ADAPTEC controller divides the disk up into 'zones' and reads the number of defects in each zone during reset.

In addition to the drive seeks and reads, the ADAPTEC controller does a series of self-diagnostics after power-up The immediate selection of the disk and movement of the heads during this period, although different from the brand-X controllers, is a sign of the proper functioning of ADAPTEC controllers.

Since the ADAPTEC controller must read several blocks from the disk at reset time, the controller will be unavailable for use for about 1 second after a hardware RESET is sent. This brings up 2 very important points about the hardware RESET.

First, the RESET line should **NOT** be used except for power-up or if the disk system is 'lost'. Your software should not send out RESET pulses prior to each command or sequence. The SASI/SCSI reset is equivalent to the host computer's RESET, to be used only for power-up and emergencies.

Second, if the format of the drive is un-readable (i.e. unformatted or formatted with a brand-X controller), the FIRST command issued after reset (except REQUEST SENSE, TEST UNIT READY or MODE SELECT) will show an error. If a SENSE command is issued, the returned error code will be an error code without the 'address valid' bit (bit 7) set. The controller will NOT allow any operation that READs, WRITEs or moves the heads on a drive with a 'blown' format. Instead, any of the above operations will return an error status and an error code of 1C. The 1C code indicates that the format on this drive is bad. The drive must be formatted by an ADAPTEC controller to prevent this 'blown format' lockout. If the drive has been formatted by an ADAPTEC controller and an error comes up at reset, the probable cause is the controller being unable to read the disk. Check the 20 pin data cable and the drive select jumpers on the drive or try to issue the command again. The ADAPTEC controller will make another attempt to recover the drive format information every time a read/write command is issued. The $1C_H$ error is returned only when the recovery attempt is failed.

Let's assume that the drive is new (unformatted). The first step is to format with the ADAPTEC controller. The software should send the TEST UNIT READY command and check the returned status byte. If the status byte is not ØØ, the command should be sent again until both the CHECK and BUSY bits are clear (i.e. the status byte=ØØ). Once a ØØ status is received, the MODE SELECT command should be issued to send the drive parameters to the controller (these parameters usually come from the user in a FORMAT utility program). Then a FORMAT command will complete the sequence. At this time, the drive is formatted in an ADAPTEC format and the drive parameters sent in the MODE SELECT command have been saved on the disk. Since the drive parameters (including step pulse information) are stored on the disk and read into the controller at RESET time, there is no need to send this information to the controller with every reset or power-up as required by the brand-X controllers. The ADAPTEC system allows you to power down a system, install or change a formatted drive, power up and have the controller self-configure for the new drive. The host can determine the drive size (read capacity command) self-configure without any driver software modification. This device-independence provides a major advantage for host systems using a true SASI/SCSI controller over the SASI-like units that send parameters at reset and with commands.

5.Ø SASI/SCSI HANDSHAKING

The SASI/SCSI bus is a simple bus to interface. However, a quick reading of the SASI/SCSI spec may leave you lost due to it's extreme attention to detail. Also, some SASI-like controllers exist on the market which allow some deviation from the ANSI protocol. We have noted that some driver routines designed for brand-X controllers simply will not work due to some short-cuts and oversights that the SASI-like devices will tolerate. The following driver routines will work with a full-SASI/SCSI controller like ADAPTEC and are downward-compatable in protocol (perhaps not actual command format) with the SASI-like units. The driver code example is written in 8Ø8Ø code but can be translated to many other CPU's.

The hardware of this sample driver is assumed to be a simple PIO 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.

An important point to remember in reading this code or designing your own driver 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 the transfer (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 SASI/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 some 6 byte commands, thus some users have set up counters in their software to only send a 6 byte command. Since the ADAPTEC 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 phases (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 ACB4000, may be ignored).

6. Check Status byte. If busy bit set, re-send command, if check bit set, send sense command to get error code.

Note that some brand-X controllers do not support the SASI/SCSI defined busy bit. However, these controllers do set this bit to zero so this proceedure will work with the ADAPTEC controllers and is down-level compatible with the others.

A sample software driver is included in the back of this Appendix for reference only. A quick look will show that the code is not at all 'tight', but was expanded to avoid clever programming 'tricks' that may be difficult to follow. The code will work but is not optimized.

6.Ø THE COMMANDS

The final components needed to operate the controller are the commands. The ANSI spec defines most of the commands that are available in the ADAPTEC controller. However, some brand-X controllers use variations of the ANSI commands that will cause command rejects by the ADAPTEC SASI/SCSIcontroller.

The major conflict is the 'reserved' bits in the SASI/SCSI spec. ANSI says that these reserved bits **must be set to zero and the controller should test them.** However, some other manufacturers have placed their own unique codes in these bits. If a system sends something other than Ø in these bits, the ADAPTEC controller will reject them.

A prime example is the last or 'control' byte of the command. Some brand-X controllers place drive parameter information in reserved bits Ø-5. If an ADAPTEC SASI/SCSI controller is sent anything but Ø, the command will be rejected.

A second cause for trouble is the lack of checking unused reserved bits by the brand-X controllers. An example is the REZERO command. Bytes 2, 3 and 4 are reserved and must be zero for a SASI/SCSI controller. However, some brand-X units ignore these bytes and thus will allow driver software to default bytes and still execute the command.

If you find that your new or existing drivers get command rejects, check for 1's in these reserved bit positions.

7.Ø A SAMPLE FORMAT SEQUENCE

For your reference here is a suggested command sequence to format a blank or differently formatted drive. Note that all commands are in HEX.

Command 1. REZERO UNIT

A 'dummy' command to allow for the 'BLOWN FORMAT' error code that will come back just after power-up. This command should be sent until the returned status byte is zero. If the 'check' bit (bit 1) is set, issue a SENSE command to determine the error code.

```
Ø1
ØØ for drive Ø, 2Ø for drive 1
ØØ
ØØ
ØØ
ØØ
```

Command 2. MODE SELECT

The mode select command sends drive parameters to the controller to be written onto the drive at format time. This command has a data block to send in addition to the command.

```
15
ØØ for drive Ø, 2Ø for drive 1
ØØ
ØØ
16
ØØ
```

The data block to send after the command will be:

```
ØØ
ØØ
ØØ
Ø8
ØØ
ØØ
ØØ
ØØ
ØØ
ØØ
Ø1-high byte of block size (256 in this example)
ØØ-low byte of block size (256 in this example)
Ø1-interface code (must be Ø1)
Ø1-high byte of cylinder count (3Ø6 in this example)
32-low byte of cylinder count (3Ø6 in this example)
Ø4-total number of heads (4 in this example)
Ø1-high byte of reduced current cyl (256 in this example)
ØØ-low byte of reduced current cyl (256 in this example)
Ø1-high byte of precomp cyl (not used by controller)
ØØ-low byte of precomp cyl (not used by controller)
ØØ-landing zone position (ØØ= none)
Ø1-stepping code (Ø1= 28us/step, buffered)
```

The available step codes are:

ØØ	3mS per step, unbuffered (ST-5Ø6)
Ø1	28 uS per step, buffered (ST-412)
Ø2	12 uS per step, buffered (most others)

Command 3. FORMAT

The format command will re-write all ID and data fields on the drive. The format command must follow the mode select command or the parameters saved by the mode select command will be lost and default values (ST512) will be assumed. The format command may also have a data block consisting of a bad block list to be used by the controller to skip the known defects. The listed format command will assume no defects. The user should then verify the disk with a VERIFY command to find the bad areas and re-format, this time sending the bad block list to the controller. If you have a bad block list from the disk manufacturer, it can be used in the first format pass instead of a 2 pass format.

A valuable feature in the ADAPTEC controller is the ability to format the drive with a default data byte (6C hex), or a user selected byte. This example will assume that the format is for a CP/M system that requires a E5 (hex) data field.

```
Ø4
Ø2-for drive Ø, 22 for drive 1
E5-the desired data field
ØØ-high byte of interleave (must be Ø)
Ø2-low byte of interleave factor
ØØ
```

Some final notes on FORMAT involve the interleave factor and disk capacity. With the ADAPTEC controller you select the desired interleave factor with the Format command. The interleave can range from zero to the number of blocks per track-1. The number represents the number of blocks between consecutive block numbers, thus an interleave of 1 means that the sectors are consecutive. If you send an interleave of Ø, the controller will pick a value of 2 for you.

An interesting feature of the ADAPTEC controller is it's ability to pack 33 sectors of 256 bytes on a track. This is automatically done if the interleave is **NOT** equal to 1. Below is a table of blocks/track with different interleaves.

SECTOR SIZE	INTERLEAVE	SECTORS/TRACK
256	1	32
256	anything but 1	33
512	1	17
512	anything but 1	18
1Ø24	1	9
1Ø24	anything but 1	9

The use of an interleave factor of 1 allows a maximum transfer rate (no gaps) but will only be effective with a host adapter and system capable of very high transfer rates. On the other hand, the use of some interleave maximizes the storage capacity of your drive.

```
        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 (1=asserted)
IO      EQU     40H             ;SASI I/O line (1=input)
MSG     EQU     20H             ;SASI MSG line (1=asserted)
CD      EQU     10H             ;SASI C/D line (1=command)
BUSY    EQU     08H             ;SASI BUSY line (1=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 Ø2 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 ØØ)
```

```
        MVI     A,Ø2H           ;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,Ø1            ;controller ID=Ø1
        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,ØØH           ;release the SEL line....
        OUT     HACTRL          ;once BUSY is active
        POP     PSW             ;then recover A/PSW
        RET                     ;and return


;***********************************************************
;*                                                         *
;*      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
                00,00,00

CMD:    DB      00,00,00,00,00  ;10 byte command area
                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 B

OPTIONAL COMMAND RECOGNITION

IN THE

ACB-4000

1.Ø USING THE EXPANDED COMMAND SET

1.1 Definition.

All commands noted in Section 6 of this manual are fully supported by the ACB-4ØØØ controller. These are a direct implementation of the ANSC SCSI command set and will cause a command reject if any of the reserved areas are violated.

In order to use the ACB-4ØØØ in existing systems which are non-compliant with the SCSI specification, a series of optional commands are supported. In most cases, these commands are simply ignored since their functions are already available in the standard command set.

This optional command set does not imply that a different disk formatting scheme is possible. In all cases, a standard SCSI Mode Select and Format program must be written. Defect skipping is always done on the block (sector) level.

2. Use of Optional Command Set

In order to implement the expanded alternate command set on the ACB-4ØØØ, a jumper plug must be installed at position I-J on the jumper pad of the ACB-4ØØØ board. This jumper is continually sampled during operation, so temporary insertion is possible for periodic use of optional commands. Under normal Adaptec SCSI use, this jumper should not be installed.

3. Command Sets Available

	COMMAND	ACB-4ØØØ	EXPANDED OPTION	NOTES
ØØ	TEST UNIT READY	YES	YES	
Ø1	REZERO UNIT	YES	YES	
Ø3	REQUEST SENSE	YES	YES	
Ø4	FORMAT UNIT	YES	YES	(1)
Ø5	CHECK TRACK FORMAT	NO	YES	(2)
Ø8	READ	YES	YES	(3)
ØA	WRITE	YES	YES	(3)
ØB	SEEK	YES	YES	
ØC	INITIALIZE DRIVE	NO	YES	(4)
ØD	READ ECC BURST LEN.	NO	YES	(5)
ØF	TRANSLATE	YES	NO	
ØF	WRITE BUFFER	NO	YES	(6)

Continued.....

	COMMAND	ACB-4000	EXPANDED OPTION	NOTES
10	READ BUFFER	NO	YES	(6)
13	WRITE BUFFER	YES	YES	
14	READ BUFFER	YES	YES	
15	MODE SELECT	YES	YES	
1A	MODE SENSE	YES	YES	
1B	START/STOP UNIT	YES	YES	
1C	RECEIVE DIAGNOSTIC	YES	YES	
1D	SEND DIAGNOSTIC	YES	YES	
25	READ CAPACITY	YES	YES	
28	READ	YES	YES	
2A	WRITE	YES	YES	
2E	WRITE & VERIFY	YES	YES	
2F	VERIFY	YES	YES	
31	SEARCH DATA EQUAL	YES	YES	
E0	RAM DIAGNOSTIC	NO	YES	(7)
E3	DRIVE DIAGNOSTIC	NO	YES	(7)
E4	CONTROLLER DIAG.	NO	YES	(7)

Notes:

1) FORMAT UNIT works according to Adaptec specifications only.

2) CHECK TRACK FORMAT always returns good status.

3) The reserved and vendor unique bits in the Control Byte are ignored when the option jumper is in place. Normal ACB-4000 use causes command rejection if these bits are set. Variable step rates and ECC control are always set by Mode Select and Send Diagnostic commands, respectively.

4) INITIALIZE is accepted, but ignored. Mode Select overrules. Good status is returned.

5) READ ECC BURST LENGTH always returns a value of 8.

6) These vendor unique Buffer commands transfer one "blocksize of data.

7) Adaptec diagnostics are executed at power-on. Requests for these diagnostics are ignored and good status is always returned.