



**SUBSET 80**

**FOR**

**FIXED AND FLEXIBLE**

**DISC DRIVES**

**(HP-IB IMPLEMENTATION)**

Hewlett-Packard Company  
700 71st Avenue  
Greeley, Colorado 80634

Manual Part Number: 5958-4129

First Printing: November, 1985  
Printed in USA

## **PRINTING HISTORY**

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The date on the title page of the manual changes only when a new edition is published. When an edition is reprinted, all the prior updates to the edition are incorporated. No information is incorporated into a reprinting unless it appears as a prior update. The edition does not change.

Many product updates and fixes do not require manual changes, and conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual updates.

First Edition. .... November, 1985

The information contained in this document is subject to change without notice.

**HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.**

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied or reproduced without the prior written consent of Hewlett-Packard Company.

## PREFACE

This manual explains and discusses the Subset 80 protocol. It is a revision of the review draft issued in July, 1983. The command set, of course, is unchanged, but the text and descriptions have been rewritten in the light of questions and comments received over the past two years.

As the title indicates, the subject of this manual is the HP-IB implementation of Subset 80 for disc drives. The information here has also been used as the basis for tape drive applications of SS/80.\* In these cases, however, there are some differences and some additions. For further information, contact the Technical Publications Department, Hewlett-Packard, Greeley, Colorado.

Should there be other reference documents on Subset 80, they can be included in the binder provided with this manual.

Please give us your comments and suggestions on the content or format of this manual.

Roger Faaborg  
Fred Cannan  
Greeley, Colorado  
November 15, 1985

---

\*Both "SS/80" and "Subset 80" are terms referring to the protocol that is the subject of this manual. The slash mark in "SS/80," of course, derives from the same mark in "CS/80." That in turn was an editor's or typesetter's change from an apostrophe used in the first typewritten notes for the "Command Set for the '80's."



# CONTENTS

	Page
Preface . . . . .	iii
Contents . . . . .	v
Figures . . . . .	vii
CHAPTER 1 INTRODUCTION . . . . .	1-1
CHAPTER 2 ESSENTIALS OF HB-IB . . . . .	2-1
CHAPTER 3 GENERAL INFORMATION . . . . .	3-1
3.1 Device Addressing Hierarchy . . . . .	3-1
3.2 Transaction Structure . . . . .	3-1
3.3 Parallel Poll . . . . .	3-4
3.4 Outline of the Command Descriptions . . . . .	3-5
3.5 A Typical SS/80 Command: An Example of HP-IB Sequence for Normal SS/80 Transaction Structure . . . . .	3-6
3.6 Power-on State and the Unseen QSTAT=2 Holdoff . . . . .	3-10
3.7 Flexible Disc Loading and Removal . . . . .	3-13
3.8 Complementary Commands . . . . .	3-18
3.9 Unit 15 . . . . .	3-19
3.10 Reporting Phase . . . . .	3-19
3.11 Timeouts . . . . .	3-19
3.12 Formatting Procedure for SS/80 Devices . . . . .	3-23
3.13 Sparing Strategy for SS/80 Devices . . . . .	3-25
3.14 Media Wear Status Bit . . . . .	3-26
3.15 Clears . . . . .	3-26
CHAPTER 4 COMMAND DESCRIPTIONS . . . . .	4-1
4.1 Commands Supported . . . . .	4-1
4.2 Commands Not Supported . . . . .	4-2
4.3 Command Descriptions . . . . .	4-3
AMIGO CLEAR . . . . .	4-5
CANCEL . . . . .	4-9
CHANNEL INDEPENDENT CLEAR . . . . .	4-11
DESCRIBE . . . . .	4-15
DOOR LOCK . . . . .	4-21
DOOR UNLOCK . . . . .	4-23
DOWNLOAD . . . . .	4-25
HP-IB PARITY CHECKING . . . . .	4-29
IDENTIFY . . . . .	4-31
INITIALIZE MEDIA . . . . .	4-33
INITIATE DIAGNOSTIC . . . . .	4-35
LOCATE AND READ . . . . .	4-39
LOCATE AND VERIFY . . . . .	4-43

## **CONTENTS (CONTINUED)**

LOCATE AND WRITE . . . . .	4-45
NO OP . . . . .	4-47
READ LOOPBACK and WRITE LOOPBACK . . . . .	4-49
RELEASE (NO OP) . . . . .	4-53
RELEASE DENIED (NO OP) . . . . .	4-55
REQUEST STATUS . . . . .	4-57
SET ADDRESS . . . . .	4-67
SET FORMAT OPTIONS . . . . .	4-71
SET LENGTH . . . . .	4-73
SET RELEASE (NO OP) . . . . .	4-75
SET RETURN ADDRESSING MODE . . . . .	4-77
SET RPS (NO OP) . . . . .	4-79
SET STATUS MASK . . . . .	4-81
SET UNIT . . . . .	4-83
SET VOLUME . . . . .	4-85
SPARE BLOCK . . . . .	4-87
UNIVERSAL DEVICE CLEAR . . . . .	4-89
VALIDATE KEY . . . . .	4-91
WRITE LOOPBACK . . . . .	4-93

## **FIGURES**

	Page
<b>CHAPTER 3</b>	
3-1 Phases of a Transaction. . . . .	3-2
3-2 Status Shown by QSTAT Byte. . . . .	3-2
3-3 HP-IB Data Lines Pulled Low by Parallel Poll. . . . .	3-4
3-4 Table of Power-on (Default) Values. . . . .	3-10
3-5 List of Only Commands that Check for a New Medium. . . . .	3-13
3-6 List of Commands to Send to Check for a New Medium. . . . .	3-14
3-7 Execution Requirements. . . . .	3-16
3-8 Secondaries and Opcodes. . . . .	3-17



# INTRODUCTION

SS/80 or SUBSET/80 is a subset of the CS/80 protocol which has been used by Hewlett-Packard on high capacity fixed discs. SS/80 will be used on both low capacity fixed discs and 3 1/2 and 5 1/4-inch flexible disc drives.

SS/80 and CS/80 are essentially the same, so it is possible for the lowest cost HP desktop computer to use the entire spectrum of HP mass storage products from the 3 1/2" flexible disc drives up to the 400 Megabyte fixed discs.

A command set defines the communication protocol between the host computer and the peripheral device. If the host wants to read some data from a disc, the command set specifies what bytes must be sent to the disc, telling it to read the data and transmit it to the host.

Older HP command sets required the host to have prior knowledge of device-specific parameters of the flexible or fixed disc drive it was using. The host software called the "disc driver" had to know that a 9895A had 2 heads and 77 cylinders. The advantage of SS/80 is that the host driver can be "parametric" and does not have to know everything about the peripheral in order to use it. The host gives the flexible or fixed disc drive the Describe command and the peripheral responds by telling the host all the necessary information it needs to use the peripheral. This includes the number of units on the device, the maximum transfer rate supported by the device, whether it has removable media or not, the block size, the number of blocks it has, etc. The host sees each device as a linear address space and does not have to know how many cylinders, heads, and sectors the device has. Peripheral manufacturers can then upgrade their peripherals, increase their storage capacity and performance, and they will function correctly without the host computer making any changes to its software. Owners of older computers can then buy the latest peripherals and use them without any software changes.

Another improvement of SS/80 over previous bus protocols is that the Initialize Media command does the entire process of formatting, verifying, and sparing to prepare the medium for use with no host involvement. The preparation of the medium for use is a very important step in providing reliable data storage.

The command set itself is channel independent. At HP, SS/80 is being implemented on HP-IB, HP-IL, and on a register interface. This document, however, deals with the HP-IB implementation.

Although the command set has a wealth of commands, only a small number are needed, so a very simple driver is possible.

SS/80 is not defining a new command set. It is only placing some constraints on CS/80 in an effort to simplify the firmware development in new peripherals and software development of new host drivers. By simplifying the command set, we have also decreased the time it takes to decode a command.

Along with this document, there are several other manuals to read:

- *Condensed Description of the Hewlett-Packard Interface Bus* (PN 59401-90030). This short manual is a good introduction to HP-IB. It is probably the best starting point for learning about SS/80.
- *CS/80 Instruction Set Programming Manual* (PN 5955-3442). This explains the whole CS/80 command set. CS/80 is the big brother to SS/80 and contains much valuable information not duplicated in this manual.

- *IEEE Standard Digital Interface for Programmable Instrumentation, IEEE Std 488-1975.* This is the complete description of HP-IB for those who want to understand HP-IB thoroughly.

When SS/80 was first defined, the following basic objectives were agreed upon:

- SS/80 should preserve the basic functionality of CS/80. We want to leverage the enormous effort undertaken in developing CS/80, and definitely do not want to define a new command set. The SS/80 commands will be implemented in the same manner as CS/80 commands whenever possible.
- SS/80, like CS/80, should have as its goal device-independent drivers, easily upgradeable for new peripherals. This is very important. As new SS/80 peripherals are introduced we do not expect host driver firmware to have to be changed. We expect SS/80 host drivers to function without modification on new higher performance and greater capacity peripherals.
- SS/80 should spell out exactly what commands an SS/80 peripheral must support. To make parametric drivers possible, all SS/80 peripherals must respond to the same commands.
- SS/80 will be tailored to the lower cost devices for which higher sales volume and greater sensitivity to cost are more important than additional features.
- The SS/80 command set will include some CS/80 commands which will be treated as NO OPs or handled in a special manner for compatibility to existing CS/80 drivers. These commands will be well documented and should not be used by new drivers.

The basic differences between the SS/80 command set described here and the CS/80 command set as implemented on the 7908 disc drive are these:

- Initialization of media is done totally by the device in SS/80. CS/80 as implemented on the 7908 has a procedure the host must follow which involves diagnostic commands.
- SS/80 devices do not support the extensive diagnostic commands which were supported on the 7908 CS/80 product. The responsibility of diagnostics is left to service routines using the Download command. This is because SS/80 devices want to keep firmware to a minimum.
- SS/80 devices do not support the release sequence of CS/80. SS/80 devices will never request release.

There are other smaller differences between SS/80 and CS/80. These are mentioned with each command description.

# ESSENTIALS OF HP-IB

CHAPTER

2

This chapter contains a brief introduction to the HP-IB. It contains just enough information to understand the rest of the document. If you are already familiar with the HP-IB, then skip to Chapter 3.

Generally on HP-IB, one device is designated the CONTROLLER. It is the device which is in charge of the bus. Normally it is a computer of some sort. The rest of the devices are LISTENERS or TALKERS or both, but not controllers. SS/80 devices can be talkers or listeners but never controllers.

The HP-IB has 16 lines. Understanding SS/80 requires a knowledge of only a few of them:

DIO1 to DIO8: These are the bi-directional data lines. The bytes of information are coded in either binary or ASCII. In ASCII the eighth bit is available for parity. We will assume the parity bit is zero in this document.

DAV (Data Valid); NRFD (Not Ready for Data); NDAC (Not Data Accepted): These three lines are used to handshake the bytes on HP-IB. The interlocked handshake used permits asynchronous communication over a wide range of data rates.

ATN (Attention): Only the controller can change the ATN line. In the handshaking process, when ATN is active, the byte on the bus is either a command or an address; when ATN is not active the byte is data.

EOI (End or Identify): Either the controller or the peripheral can change EOI. The "END" meaning comes from the fact that EOI active on a byte transfer means that the byte is the last one. If the controller or peripheral sends out bytes, it always tags the last byte with EOI. The "Identify" meaning comes from parallel poll which is discussed later.

IFC (Interface Clear): These cause the peripheral's HP-IB chip to be cleared but does nothing else to the peripheral. It is unlike the Universal Clear, Amigo Clear, and Channel Independent Clear which affect the peripheral's target unit, volume, address, etc.

Each SS/80 peripheral has two addresses. This document refers to them as the major and minor address. The major address can be any number from 0 to 7. It is set by a switch on the device. Each SS/80 device also has a minor address of 31. The minor address is used only during the Identify command.

Along with the major address, the SS/80 peripherals respond to secondary addresses. In IEEE 488, secondary addresses are extended addresses, just in case more addresses are needed. In SS/80 the secondaries are used to denote phases of a transaction. The phases are command, execution, and report. The two addresses, 0-7 and 31 are the device's primary address.

If ATN is active, we know the byte is a command or address. Suppose the 8 data bits are numbered P8 through P1. In this case, bit 8 is a parity bit. We will assume in this document that the parity bit is always a zero; it is denoted by "P." Bits 7 and 6 have the following meaning:

- P7,P6 = 11      The remaining bits are a SECONDARY address.
- P7,P6 = 10      The remaining bits are the TALKERS primary address. P1011111 is the Untalk command.
- P7,P6 = 01      The remaining bits are the LISTENERS primary address. P0111111 is the Unlisten command.
- P7,P6 = 00      The remaining 5 bits define an HP-IB command. For example, P0010100 is a Universal Clear.

Let us suppose we were to set up a logic analyzer on the HP-IB and watch the bus activity for some SS/80 commands. We would want to display the eight data lines, ATN, and EOI. DAV is a good signal to use as the trigger. A simple Set Unit command would appear as follows:

DATA	ATN	EOI	COMMENTS
P0111111	ATN	-	The controller gives the Unlisten command. Nobody on the bus should listen.
P1001111	ATN	-	The controller which happens to be at address 15 will be the talker.
P0100111	ATN	-	The controller tells the device at address 7 to listen.
P1100101	ATN	-	This is a secondary of 65H. It marks the start of a command message in SS/80.
0010XXXX	-	EOI	This is a data byte since ATN is not active. This byte is an opcode for a Set Unit command, 2XH, where X is the unit. Since the host is sending only one byte, it is the last, and is tagged with EOI.
P0111111	ATN	-	Host Unlisten command. Nobody is addressed to listen.
P1011111	ATN	-	Host Untalk command. Nobody is addressed to talk.

The preceding sequence is a command message. The included command is Set Unit, but could have included additional complementary commands. Complementary commands are the commands Set Unit, Set Volume, Set Address, Set Length, and Set Status Mask which can be sent alone, or in groups with other complementsaries. After the last complementary in the command message, there could also be a Real Time, General Purpose, or Diagnostic command.

The data passed in the command message for a read usually includes the following after the secondary of 65H:

Set Unit	20H	opcode for Set Unit to Unit 0.
Set Volume	40H	opcode for Set Volume to Volume 0.
Set Address	10H	opcode for Set Address 00H Set Address is followed by 6 00H parameter bytes which define the 00H target address in blocks. 00H Example shows address of 0. 00H 00H
Set Length	18H	opcode for Set Length 00H Set length is followed by 4 00H parameter bytes which define 01H the target length in bytes. 00H 256 bytes used in this example
Locate and Read	00H	opcode for Locate and Read

In the above example, the opcode for Locate and Read would have EOI set since it is the last byte.

We have shown the command message. There are also execution messages and report messages. An execution message has a secondary of 6EH and the data passed are not opcodes or parameters. In Locate and Read or Locate and Write, the data in the execution message are the data being written to or read from the disc. In a reporting message, the secondary is 70H, and only one byte is sent from the peripheral to the host. This single byte is the QSTAT, or QUICK STATUS, which gives the results of the transaction.



# GENERAL INFORMATION

## 3.1 DEVICE ADDRESSING HIERARCHY

In SS/80, as in CS/80, the addressing hierarchy in descending order is (1) device, (2) unit, and (3) volume. A device on HP-IB is characterized by an HP-IB address, usually switch selectable. The HP-IB address can be set to any address from 0 through 7. Each device on the bus must have a unique HP-IB address.

For example, a flexible disc drive and fixed disc might be packaged in one box with one HP-IB connector and its associated HP-IB address switch. In this combination, each drive is a unit. The fixed disc might be Unit 0 and the flexible disc drive Unit 1. There can be up to 15 units in a device. See the Set Unit command for more information on units.

A unit can be further split up into volumes. There can be up to eight possible volumes for each unit, numbered 0 through 7. Some of the SS/80 fixed discs support multiple volumes. A volume is a portion of a unit. A volume must be initialized independently, with its own interleave factor. A fixed disc split into several volumes allows for different file systems on each volume and keeps directories to a more manageable size. See the Set Volume command for more information on volumes.

To fully address a volume requires giving the HP-IB address, unit number, and volume number.

## 3.2 TRANSACTION STRUCTURE

A transaction is a logically complete operation between a host controller and a device (peripheral). The transaction begins when the peripheral accepts a command and ends when a reporting message indicating the pass/fail status of the transaction is accepted by the host. The three phases of a transaction are command, execution, and reporting. The following discussion of a transaction is summarized in Figure 3-1.

For example, a transaction might be a read of 512 bytes. The transaction would start with a command message from the host. The host addresses the device to listen, gives it a secondary of 65H followed by a string of bytes. These bytes may have complementary commands and parameters. The last byte would be the opcode for the Locate and Read command. The peripheral takes in the secondary, disables parallel poll, takes in all the parameter bytes, checks them, and does a seek if necessary. When the peripheral is ready to enter the execution phase of the transaction, it enables parallel poll. This ends the command phase of the transaction. We are now in the execution phase.

The host sees that the device has enabled parallel poll, addresses the peripheral device to talk and sends a secondary of 6EH. The device accepts this secondary, disables parallel poll and sends the 512 bytes of data to the host. When the last byte has been sent, the peripheral enables parallel poll to tell the host that it is ready for the reporting message. This ends the execution phase of the transaction and we are now in reporting phase.

Upon seeing that the peripheral device has again enabled parallel poll, the host sends the reporting phase secondary of 70H. The device accepts this secondary, disables parallel poll and sends back a single byte of QSTAT. Parallel poll is not enabled. Sending the byte of QSTAT to the host completes the reporting

message and completes the transaction. The device then goes to a command ready state or command phase where it waits for the next command.

COMMAND MESSAGE	-	Secondary of 65H (P1100101) Secondary address of 05H
	.	optional complementary commands.
	.	optional command with EOI on last byte
	.	Host waits for peripheral to enable parallel poll.
EXECUTION MESSAGE	-	Secondary of 6EH (P1101110) Secondary address of 0EH
	.	real data bytes (read or write data)
	.	EOI on last byte
	.	Host waits for peripheral to enable parallel poll.
REPORTING MESSAGE	-	Secondary of 70H (P1110000) Secondary address of 10H
	.	one byte of QSTAT with EOI
	.	Peripheral leaves parallel poll disabled

Figure 3-1. Phases of a Transaction

The QSTAT byte reflects the current status of the currently selected unit. The various meanings of the QSTAT byte are listed in Figure 3-2.

QSTAT = 0	Everything is O.K.
QSTAT = 1	There has been an error. Do a Request Status to find out which error occurred.
QSTAT = 2	The device has just powered up or a new flexible disc has been loaded. The current transaction may not have been done. The host should reconfigure for the new medium.

Figure 3-2. Status Shown by QSTAT Byte.

The sequence of command, execution, and report must be followed. The peripheral controls when the execution phase and report phase can begin by enabling parallel poll response. After sending the QSTAT byte the peripheral enters a command ready state where it waits for the next command from the host. This command ready state is called command phase so the device is always in one of the three phases. The command message secondary actually starts the transaction, although the device is already in command phase. Thus the read transaction is split up into the command message, execution message, and reporting message. The peripheral goes from command phase, to execution phase, to reporting phase and back to command phase.

Some transactions don't require that any read or write data be passed and therefore don't have an execution phase. In these commands the peripheral goes from command phase to reporting phase.

In SS/80 any background tasks performed by the peripheral are done in the command ready phase and will be stopped if there is a command on the bus. Thus the background diagnostics, if they exist, are totally transparent to the host. Unlike CS/80 devices which may request release from the host to go off line, SS/80 devices never request release.

An SS/80 peripheral device can only do one transaction at a time. The peripheral device can consist of several units and volumes. A host must only do one transaction at a time with one of the units or volumes of the device. It must complete the transaction, all 3 phases, with the SS/80 unit or volume of the device before starting another transaction with a unit or volume on the same device. The SS/80 device does not support overlapping of transactions between units and volumes of the device. A host can overlap transactions with separate devices on the same HP-IB but not to separate units or volumes of the same device. A host could do several transactions at once. A device is limited to a single transaction. Needless to say, when a host is overlapping transactions between several devices on the HP-IB it should be careful to make sure each transaction is completed properly with each device.

Commands that may not follow this transaction procedure are the Transparent Commands. The transaction model does not always apply; when using an SS/80 command, check the command listing in Chapter 4 for the precise interactions between the host and peripheral device for that command.

### 3.3 PARALLEL POLL

A device can have its parallel poll response either enabled or disabled. When the host controller puts both ATN and EOI active at the same time, it is conducting a parallel poll. Any device which is enabled for parallel poll will pull one data line low. If the device is not enabled for parallel poll, it will do nothing. Which line the device pulls low is dependent on its HP-IB address, as shown in Figure 3-3.

HP-IB major address	Data Line pulled low
0	DI08
1	DI07
2	DI06
3	DI05
4	DI04
5	DI03
6	DI02
7	DI01

Figure 3-3. HP-IB Data Lines Pulled Low by Parallel Poll

There is no byte handshake in a parallel poll. The host sets ATN and EOI high, waits at least 2 microseconds, and then reads the data lines to see what devices have parallel poll response enabled and what ones do not. If the host is interested in the device at address 7, then it will watch DIO1. If DIO1 is low during a parallel poll, the device at address 7 has its parallel poll response enabled. During a command message, the host will give the peripheral the command phase secondary of 65H, then the opcodes and parameters for the command, followed by an Unlisten and Untalk. The peripheral first disables its parallel poll response (DPPR) and then goes off to do the Set Unit, Set Volume, etc. When the peripheral is done, it enables its parallel poll response (EPPR). This tells the host that the peripheral is ready to enter the next phase of the transaction. By enabling its parallel poll response, the peripheral is telling the host it is done with this part of the transaction and ready for the next.

Another way of looking at parallel poll is that when a device has its parallel poll response enabled, it is requesting service from the host. The device is then ready to accept the next phase of the transaction. Parallel poll response is disabled when the secondary is received by the peripheral from the host, and is not re-enabled until after all data bytes have been sent or received, or any steps the device has to do are completed. The exception is the reporting phase message which ends when the peripheral returns the QSTAT byte to the host. Parallel poll is not enabled at this time.

For the peripheral, it may be best actually to disable its parallel poll response before the secondary is taken off the bus, thereby keeping a fast host from giving the peripheral the secondary, seeing parallel poll enabled (not yet disabled), and assuming the device is done when it actually hasn't yet started. This race condition also occurs on a Universal Clear where there is no secondary. To avoid this the host should give the Universal Clear and then wait one millisecond before it starts looking for the device to enable its parallel poll response.

A host should always wait for parallel poll to be re-enabled before giving the peripheral device the secondary for the execution or reporting phases of a transaction. If the host sends the next secondary without waiting for parallel poll, some low cost HP-IB chips used in the peripheral will not handshake the secondary off the bus until the peripheral device is done with the previous phase of the transaction.

It is poor usage of the HP-IB to have secondaries hanging around. If the host chooses to send secondaries without waiting for parallel poll, it must be sure he does not incorrectly timeout the device, since the device is still working on the first secondary. After the reporting phase, the device does not re-enable parallel poll. At this point the transaction is done. For a typical example of how parallel poll is used see "A Typical SS/80 Command," p. 3-6.

### **3.4 OUTLINE OF THE COMMAND DESCRIPTIONS**

The last chapter of this manual lists all the SS/80 commands in alphabetical order. The listing for each command is made up of the parts listed and defined below. For commands which follow the normal transaction structure, the opcode and parameters are listed in the "Format" section. For commands that do not follow the normal transaction structure, the individual HP-IB steps are shown.

**FORMAT:** Gives the opcode and parameters of the command. Only the opcode and parameters are specified if the command follows normal transaction structure (command-execution-report (CER), or command-report (CR)). The following section ("A Typical SS-80 Command," p. 3-6) gives the HP-IB sequence for a command that uses this normal structure. All the other commands following the same structure can easily be formed by following this typical command, changing the opcode and parameters, and deleting the execution phase if necessary. If a command does not follow normal transaction structure, the command listing includes the full HP-IB sequence, and the Format section is omitted.

**HP-IB SEQUENCE:** For certain commands that do not follow normal transaction sequence described above, the complete sequence of events over the HP-IB for a command with good syntax and normal completion is included. The following abbreviations are used in listing the HP-IB sequence:

<XXXXXXXX>	bytes transferred from the peripheral to the host.
-XXXXXXXX-	bytes transferred from the host to the peripheral.
P	parity bit on the HP-IB. In this manual the parity bit is shown as P or set to 0.
ADDRS	HP-IB address of the peripheral device in binary.
addrs	host controller's HP-IB address in binary.
ATN	attention line of the HP-IB.
EOI	termination line of the HP-IB.
X	a "don't care" value of a bit.
DPPR	peripheral device disables its parallel poll response.
EPPR	peripheral device enables its parallel poll response.

**TYPE:** States whether, in CS/80, the command is classified as Real Time, Complementary, General Purpose, Diagnostic, or Transparent.

**TRANSACTION FLOW:** Specifies whether the command has Command, Execution, or Report phases.

**DESCRIPTION:** A paragraph or two explaining how the command works.

**VARIATIONS FROM CS/80:** Spells out any differences between the SS/80 command and the same command in CS/80.

### 3.5 A TYPICAL SS/80 COMMAND: An Example of HB-IB Sequence for Normal SS/80 Transaction Structure

Most SS/80 Commands follow the command message structure listed below. For these commands, only the opcode and the parameters are given in the Format section of the command listing. For any command which does not follow this structure, the full HP-IB sequence is included in the command listing.

The command used in the example below is Locate and Read, in which the full transaction structure is used. (That is, Command Message, Execution Message, and Reporting Message.) Real Time, Diagnostic, and General Purpose commands follow this structure. Many of them, however, omit the Execution Message.

#### \*\*\*\*\* COMMAND MESSAGE \*\*\*\*\*

< P 0 1 1 1 1 1 1 >	ATN	Unlisten from the host. Nobody is addressed to listen.
< P 1 0 a d d r s >	ATN	The host addresses itself to be the talker. "addrs" = bus controller's address.
< P 0 1 A D D R S >	ATN	The host tells the device at address "ADDRS" to be a listener. "ADDRS" is a peripheral's address.
< P 1 1 0 0 1 0 1 >	ATN	Command message secondary from the host. This 65H tells the peripheral the command message is coming. (Secondary address of 05H)
	DPPR	The peripheral device disables parallel poll response when it sees the secondary.
< . . . . . . . >		Host sends 0 to N complementary commands.
< . . . . . . . >		The required order is:
< . . . . . . . >		Set Unit, Set Volume, Set
< . . . . . . . >		Address, and Set Length.
< . . . . . . . >		(Set Unit must come first.)
etc.		
< 0 0 0 0 0 0 0 0 >	EOI	Host sends the opcode for the Locate and Read command. Last byte is always tagged with EOI. (For another command, this opcode would be different and may have parameters following it. The last byte would have the EOI)

< P 0 1 1 1 1 1 >	ATN	The host unlists the peripheral. This Unlisten can be sent anytime after the last parameter byte is sent and doesn't have to occur exactly at this time.
. . .		The peripheral device executes any complementaries, checks to see if medium is ready and seeks to the correct track.
	EPPR	The peripheral device enables its parallel poll response. By enabling parallel poll, the device is asking for the execution message.
		The host should always wait for the device to enable its parallel poll response before starting the execution message.

\*\*\*\*\* EXECUTION MESSAGE \*\*\*\*\*

< P 0 1 1 1 1 1 >	ATN	Unlisten from the host. Nobody is addressed to listen.
< P 0 1 a d d r s >	ATN	The host addresses itself to be a listener. "addrs" is the host address.
< P 1 0 A D D R S >	ATN	The host tells the device at address "ADDRS" to become the talker. This is the primary talk from the host.
< P 1 1 0 1 1 1 0 >	ATN	The host gives the execution phase secondary, 6EH, to the device. (Secondary address of 0EH)
	DPPR	The peripheral device disables its parallel poll response when it sees the secondary.
. . .		Peripheral device reads first sector.
- X X X X X X X X -		Peripheral sends data to the host.
- X X X X X X X X -		
. . .		

- X X X X X X X X -		
. . .		If more data is needed, peripheral device reads another sector.
. . .		Peripheral sends data to host, tagging last byte sent with EOI. 1 to n bytes can be sent to host.
- X X X X X X X X -		
- X X X X X X X X -		
. . .	EOI	
- X X X X X X X X -		
	EPPR	Peripheral enables parallel poll response. Peripheral is requesting reporting phase message.
< P 1 0 1 1 1 1 1 >	ATN	Host untalks the peripheral. This Untalk can occur anytime after the last data byte is sent to the host.
		The host should wait for the peripheral to enable its parallel poll response before starting the reporting message.

\*\*\*\*\* REPORTING MESSAGE \*\*\*\*\*

< P 0 1 1 1 1 1 1 >	ATN	Unlisten from the host. Nobody is addressed to listen.
< P 0 1 a d d r s >	ATN	The host addresses itself to be a listener.
< P 1 0 A D D R S >	ATN	The host tells the device at address "ADDRS" to talk. Primary talk from the host.
< P 1 1 1 0 0 0 0 >	ATN	Reporting message secondary from the host. The reporting message secondary is 70H. (Secondary address of 10H)
	DPPR	The peripheral device detects the secondary and disables parallel poll.
- Q S T A T -	EOI	The peripheral sends the QSTAT byte tagged with EOI. This completes the reporting message and completes the transaction. Note that parallel poll is not enabled at the end of the command. The peripheral goes to

		command ready phase to wait for a new command.
< P 0 1 1 1 1 1 1 >	ATN	The host gives the Unlisten command. Nobody should listen. The host doesn't have to do this at this time.
< P 1 0 1 1 1 1 1 >	ATN	The host gives the Untalk command. Nobody should talk. The host can do this whenever it wants to.

Although this shows a Locate and Read command, all Real Time, General Purpose, and Diagnostic commands follow this same structure. The opcode and its parameters will be different, and you may not want to include as many complementaries, but the rest is identical.

Some hosts put Unlisten <P0111111> and Untalk <P10111111> at the start and end of each of the phases to make sure they know the state of the bus. The order of the Untalk and Unlisten is not important. The Unlisten is a good idea to make sure there are no other listeners. The Untalk is not necessary because someone is always addressed to talk at the beginning of each message, which effectively untalks all others. There can only be one talker, but there can be multiple listeners. What is important is what devices are addressed to talk and what ones are addressed to listen.

With the ABI and PHI HP-IB chips, the command addressing the peripheral to talk or listen must be immediately before the secondary. The device primary address and the secondary must be paired for these HP-IB chips.

### 3.6 POWER-ON STATE AND THE UNSEEN QSTAT = 2 HOLDOFF

When power is first applied, SS/80 devices initialize their HP-IB chips and do not initially respond to polls. Any bus activity directed to the device's HP-IB chip may be lost when the chip is reset and initialized during the power-on selftest. A selftest is then done by all units and all parameters will be set to their power-on state. (See Figure 3-4 for these power-on values.) Each unit will set its Power Fail status bit, causing its QSTAT to be equal to 2. The QSTAT byte is just a reflection of what status bits are set. It is a summation of the status or QUICK STATUS.

Set Unit	0
Set Volume	0
Set Address	0
Set length	-1 (full volume)
Set Burst	disabled (Not supported in SS/80)
Set RPS	disabled
Set Status Mask	disabled (No status bits are masked)
Set Release	T=0 Z=0
Set Format Options	device specific default value
Set Return Addressing Mode	single vector (only mode allowed in SS/80)

Figure 3-4. Table of Power-on (Default) Values.

The HP-IB chips on SS/80 devices may have a single register for incoming secondaries and data, whereas the CS/80 devices usually use the PHI chip which has an 8-byte FIFO. This makes the operations of the two somewhat different. A secondary given to an SS/80 device while it is doing selftest may not be removed from the bus until the processor finishes the power-on selftest, enables parallel poll, and then notices an incoming secondary. For example, if the secondary for an Amigo Clear is given to an SS/80 device while it is doing the power-on selftest, the peripheral device will finish selftest, enable parallel poll, notice the secondary, disable parallel poll and do the clear. The secondary hangs the bus while the SS/80 device is doing selftest. The host can remove the secondary from the bus by using an IFC (Interface Clear).

When everything is tested, parallel poll response will be enabled and the device is then ready for operation. Each unit will store the results of its selftest in its status bits and will set the Power Fail status bit which causes a QSTAT = 2. Each unit will then enter the reporting phase. Until the host has seen the power-on QSTAT= 2, the peripheral will keep the host from doing certain commands. (Commands affected are listed in Figure 3-7.)

Whenever the Power Fail bit is set and QSTAT = 2, the peripheral must keep the host from doing anything to the medium until the host has realized that it is a new medium. This is the reason there is a holdoff. As soon as the host has seen the QSTAT=2, then the peripheral will execute any of the commands. Whenever power fails or a new medium is detected, the Power Fail bit is set and QSTAT will equal 2.

If the host gives a command like a Locate and Write while there is an unseen QSTAT = 2 holdoff, then the bytes for the command will be accepted from the host but no write will be done. In the reporting message of the Locate and Write command, the host will see the QSTAT of 2 and realize that the current transaction was not done. The QSTAT of 2 that the unit in the peripheral has set will be seen by the host on the next reporting message that the host does. Returning the QSTAT=2 byte to the host eliminates the holdoff but does not clear the QSTAT byte.

To clear the QSTAT byte the host must give one of the clear commands or do a Request Status command. An Amigo Clear, Universal Clear or Channel Independent Clear to Unit 15 will clear all the status bits in all units except the Diagnostic Result bit. In SS/80, none of the Clears will ever clear the Diagnostic Result bit. Since the Power Fail bit is cleared, the QSTAT of 2 will no longer exist. If the Diagnostic Result bit is

set, the QSTAT will change from 2 to 1. At the end of the next transaction the host will then see QSTAT=1 and do a Request Status to see that the Diagnostic Result bit was set. The Diagnostic Result bit is set if there has been a selftest failure. We want to make sure the host sees that the selftest failed. A selftest failure of any unit is reported only in that unit's status at power-on. The other units that passed selftest are fully functional and can be used. The unit which failed selftest shouldn't be used but there is no holdoff in SS/80 for any QSTAT other than an unseen QSTAT=2. After any failure, the host will do a Request Status command which clears the QSTAT to 0 and then the host can execute any command.

Remember that each unit, including Unit 15, has a QSTAT byte and if you do a Channel Independent Clear to Unit 0, the QSTAT of 2 held by Unit 0 will be cleared but QSTAT will still equal 2 in Unit 15. This is not a problem if you never want to talk to Unit 15, but you should understand that each unit has its own QSTAT. It is advisable at power-on to clear the QSTAT=2 from all units to which you intend to give commands.

It is allowable for the host to do what is called a "Stand alone QSTAT," in which the host does a reporting phase message not associated with a transaction. This will also give the host the QSTAT = 2 if the unit of the peripheral has already set its QSTAT value to 2. The host must send a Request Status or do a Clear actually to change QSTAT from its power-on value of 2. Once the host has seen the QSTAT=2 for a unit, that unit will accept all commands even though the QSTAT hasn't been cleared. The peripheral will keep track of whether the host has seen the QSTAT or not. The host should do a Request Status command or one of the Clears, but it will be allowed to give commands and they will be executed as long as the QSTAT for that unit was seen by the host. This is the way CS/80 does it. Thus there is no QSTAT=2 holdoff. The holdoff is on an "unseen" QSTAT = 2. As soon as the QSTAT has been returned, although it may still be 2, the peripheral device enters command phase and all commands may be executed. It is imperative that the host, upon seeing the QSTAT=2, do a Describe command and reconfigure the system for the new medium.

With a fixed disc drive, the host will see QSTAT=2 only at power-on. With a flexible disc drive, the host could see QSTAT=2 during the reporting message of certain commands because the user can at any time pull out a flexible disc and put it or another one back in. See "Flexible Disc Loading and Removal," p. 3-14. The peripheral controller checks for a new flexible disc only on certain commands that access the flexible disc drive. Thus the situation could occur in which the host will see QSTAT=2 twice in a row. First the flexible disc drive powers up with no medium loaded. QSTAT equals "2" because the Power Fail bit is set to indicate power-on. Then before the host talks to the device, the user loads a flexible disc. When the host first talks to the device, it will see the QSTAT=2 from power-on. When the host first gives an access command to the flexible disc drive, the controller will detect the new medium and set the power fail bit causing a second QSTAT=2. If the medium is loaded when power is applied, then the host will see only one QSTAT=2. This situation occurs because the flexible disc drive looks to see if there is a new medium only when it does commands which have to access the medium. The host driver should be able to handle QSTAT=2 occurring at the reporting phase of any command and then it should go through a routine to re-configure for the new medium.

For a host to see what CS/80, Amigo, or SS/80 devices are connected to the HP-IB, the following two methods are recommended:

#### METHOD A

1. The host scans the HP-IB by doing an Identify command to each of the 8 addresses and waits 25 ms at each address for bytes to be returned. If no bytes are returned, the host issues an IFC (Interface Clear) and then goes to the next address and continues the process. It takes only 200 ms for each loop of checking for devices on the bus. the next address and continues the process. It takes only 200 ms for each loop of checking for devices on the bus.
2. Once a device has returned the Identify bytes, the host then gives an Amigo Clear to the device and waits for parallel poll. When parallel poll is returned, the host knows the device is

ready and is in a cleared state. The Amigo Clear will clear the QSTAT=2 in all units of the device.

3. Then the host can do a Describe command if the device is a CS/80 or SS/80 device and follow the Describe with whatever commands it wants to give.

#### METHOD B

1. The host gives a Universal Clear to clear all devices on the bus, or gives Amigo Clears to each of the eight bus addresses.
2. The host then waits for parallel poll from the devices which may be doing selftest.
3. Once a device has enabled its parallel poll response, the host then gives another Amigo Clear in case the first one was lost when the peripheral device's HP-IB chip was reset.
4. Then the host can give the device whatever commands it wants, which should include a Describe for SS/80 or CS/80 devices.

Both methods require that the host wait for a device to finish power-on selftest without the host knowing how long it takes to do the selftest. There is no way around this, so the timeouts the host uses at power-on must be long enough to accommodate any current or future device. See "Timeouts," p. 3-20.

Method A holds to the principle that you shouldn't do clears other than IFC's to devices unless you know what they are. Some printers may lose information or the top of the form if they are given a Universal Clear or Amigo Clear. There may be devices on the bus which are neither Amigo, CS/80 or SS/80 devices. These devices may not recognize an Amigo Clear.

The host can't just wait for a device to enable its parallel poll response at power-on because that device may have been on and have its parallel poll disabled. The host itself may have been powered off and then back on while the peripheral device remained on. The normal idle state of the peripheral after the reporting message is to have its parallel poll response disabled.

When the host first does a Request Status after power-on, it is important that it look through the status bits for the Diagnostic Result bit. In SS/80 the power-on selftest is very thorough; it is important that the user be notified if the disc is failing at power-on.

All SS/80 devices do a thorough selftest at power-on. The selftest will test as many of the components in the product as possible. This test includes checking the microprocessor, doing a checksum on the ROM, testing the microprocessor RAM and buffer RAM, checking the HP-IB chip, testing any controller chips, and doing a read/write test on the medium. On a flexible disc drive, the read/write test will be performed only if the medium is loaded and not write protected. For a fixed disc, the medium is assumed to be formatted, so the power-on selftest for a fixed disc will fail if the medium can't be read or written. Because the read/write selftests are performed on special system tracks, no user data is involved.

### 3.7 FLEXIBLE DISC LOADING AND REMOVAL

When a flexible disc is loaded into an SS/80 unit, that unit will set the Power Fail status bit which causes a QSTAT of 2. This will happen when the flexible disc drive controller first notices that there is a new medium. (The term "new" here means that the medium is newly loaded. The peripheral controller does not distinguish between a new piece of medium and one that was unloaded and then put back in again. In either case the medium is considered new.) The controller will look to see if there is a new disc or not only when it is given a command which involves some activity on the medium. For example, if a new medium is loaded and the host gives a Set Unit command, the peripheral will not check to see if there is a new medium present. The peripheral simply does the Set Unit command and doesn't check for a new medium. However, if the host then gives a Locate and Read to the unit that has the new medium, the flexible disc drive controller will detect the new medium, not do the read command and go to reporting phase where a QSTAT of 2 is returned. The QSTAT of 2 tells the host that the current transaction was not done and it must reconfigure because a new medium was loaded.

When a flexible disc is removed from an SS/80 unit, no error bits are set, and QSTAT is not changed.

Figure 3-5 lists the commands that check for the presence of a medium. They will return a QSTAT = 2 if a new medium is detected:

```
Locate and Read
Locate and Write
Locate and Verify
Spare Block
Validate Key
Initiate Diagnostic
Describe (checks medium only once)
Initialize Media
```

Figure 3-5. Commands that Check for a New Medium.

The command which first returned the QSTAT = 2 in the reporting message may not have been done. This keeps the host from writing on a new medium. In general, the host should assume that any transaction which returns a QSTAT = 2 was not done and the transaction must be redone after the host reconfigures for the new medium.

The following commands do not access the medium and would not detect a new medium: Set Unit, Set Volume, Set Address, Set Length, No Op, Set RPS, Set Release, Set Status Mask, Set Return Addressing Mode, Request Status, Release, Release Denied, Door Lock, Door Unlock, HP-IB Parity Checking, Loopback, Cancel, any clear, or Identify. Any command can return a QSTAT of 2, but only certain commands actually check to see if a new medium is present. These commands will set QSTAT=2 and it will remain set until a Clear or Request Status clears it.

If the host wants to see whether a medium is present before a Read, it should do a Locate and Read of Length 0 and not include the Set Address Complementary. This is a seek to the track the device is already on. It will result in a QSTAT of 2 being set if a new medium is loaded and a Not Ready error (QSTAT=1) if there is no medium loaded. The host will then know if the medium is there or not. The complementaries would be Set Unit, Set Volume, and Set Length to 0, with the Locate and Read opcode. Not including the Set Address complementary will cause the peripheral to use the target address which is where the head actuator is already positioned. No seek will take place.

If the host wants to see whether the medium is present before a write, it should give a Locate and Write of Length 0 and not include the Set Address Complementary. This is again a seek to the track the head actuator is already on. It will result in the Power fail bit being set (QSTAT=2) if a new medium has been loaded, a Not Ready bit set (QSTAT=1) if there is no medium, and a Write Protect bit being set (QSTAT=1)

if the medium is write protected. The host then knows whether the medium is there and if he can write on it. The complementaries would be Set Unit, Set Volume, and Set Length to 0, with the Locate and Write opcode. Again, the Set Address is not included because we don't want to move the head, only to have the controller determine whether the medium is new or not.

Host should do a seek without changing the target address.

2XH	Set Unit opcode
4XH	Set Volume opcode
10H	Set Length opcode
00H	
00H	4 bytes of length should be zeros.
00H	
00H	
00H/02H	Locate and Read/ Locate and Write opcode. These are seeks so there is no execution phase.

Figure 3-6. Commands to Send to Check for a New Medium.

The host should not use the Describe command to see whether a medium is there or not. It is much quicker to do the seeks to the target address mentioned above. The Describe command is very important in telling you the properties of the device and medium, but shouldn't be used to check if a disc is in or not. The bytes V7-V12 in Describe contain the maximum value of the single-vector address for the medium loaded. If there is no medium loaded, this single vector address is 0. If a Describe command is given to a unit and a new medium is detected, that unit will update its Describe information and return the description of the new medium. The Power Fail status bit is set causing a QSTAT of 2 to be returned. The host will see the QSTAT of 2 and realize that the a new medium was loaded. The host should clear the QSTAT and do a Describe again, then set up any operating parameters for the new medium.

After the peripheral has detected a new medium and updated its Describe information, the peripheral will not read this information from the disc again. Describe commands from the host after the first one will execute more quickly since the peripheral is simply returning the Describe information from its RAM and does not have to read the disc each time. Of course, if the medium is changed, the peripheral will read the new Describe information from the disc. Another way of saying this is that if the host loads a new flexible disc, and does six Describe commands in a row, only the first one will cause the Flexible Disc Access LED to light up.

There are problems with using Describe when you give it to Unit 15. Unit 0 may detect a new medium and update its Describe information. The reporting message returned from Unit 15 at the end of the command would not tell you that unit 0 has a new medium. In general is is safest to give no commands to Unit 15.

Use the read or write seeks specified above if you want to check that the medium is present before giving a command that accesses the flexible disc. The Describe command wasn't intended to become a "medium-in" command.

When the host gets back a QSTAT=2, it knows that a new disc has been loaded or power failed. The current command may not have been executed. To clear the QSTAT=2 the host must issue a Request Status command to that unit or do a Channel Independent Clear to that unit. Remember that each unit has status bytes and also a QSTAT, which is a kind of summation of those status bytes. The Amigo Clear, Universal Clear, and Channel Independent Clear to Unit 15 clear all units of a device. After seeing the QSTAT = 2 in the unit with the new medium, the host should clear that unit's QSTAT = 2 and then do a Describe command and reconfigure for the new medium.

Another case worth mentioning is that of a write protected flexible disc. If a new flexible disc is loaded which is write protected, the host will first see a QSTAT=2 after the write command. The write was not done. The host could then do a Request Status to clear the QSTAT=2. If the host then does the Locate and Write, it will now fail with a QSTAT=1 because the medium is write protected. As soon as the flexible disc detects the new medium it stops before doing the write, and sets the Power Fail status bit causing a QSTAT=2.

Better performance can be obtained if the driver doesn't have to do a check for the new medium. The driver should be designed so it can recover from seeing the QSTAT of 2 and reconfigure.

### Low Cost, Unintelligent Drives

Some of the very lowest cost flexible disc drives have no way of detecting a new medium. These devices will have bit 7 in U1 of the Describe command set to 1. Such an unintelligent, inept, Boeotian flexible disc drive will have QSTAT=2 only at power-on. It provides no protection against over-writing good data if the user inadvertently changes the medium. Intelligent flexible disc drives have bit 7 in U1 of the Describe command set to 0.

### Table of Execution Requirements

Figure 3-7 is a table of execution requirements. It is organized as follows:

The first column lists all the commands done by an SS/80 flexible disc or fixed disc.

The second column tells whether this command is held off and not executed if the host has not seen the QSTAT of 2. Note that all the commands are held off except Set Unit and the transparent commands.

The third column gives the state in which the peripheral must be before it will receive and execute the command. Normally the peripheral must be in the command-ready state (command phase) to execute a command. SS/80 peripherals will accept a stand alone QSTAT or reporting message in any phase. Should the host and peripheral somehow get out of synchronization, they will regain it in the reporting phase. In the command ready state, SS/80 peripherals will accept and sink (throw away) any number of bytes from the host. If the SS/80 device is a talker and some failure occurs, the SS/80 device will send a byte of value 1 tagged with EOI. The host must be able to stop receiving bytes on this EOI. The capability to sink bytes and source one with EOI keeps the host and peripheral from hanging during certain error conditions.

The host must always follow the transaction process for the standard SS/80 commands.

The last column shows which target units are allowed when a command is executed. In most cases it is obvious since it makes no sense to give a Locate and Read to the controller (Unit 15).

Command	Holdoff on Unseen QSTAT = 2?	Required state which peripheral must be in to start transaction.	Allowed target units for this command.
Set Unit	No	Command ready	Any
Set Volume	Yes	Command ready	Not Unit 15.
Set Address	Yes	Command ready	Not Unit 15.
Set Length	Yes	Command ready	Any
No Op	Yes	Command ready	Any
Set RPS	Yes	Command ready	Any
Set Release	Yes	Command ready	Any
Set Status Mask	Yes	Command ready	Any
Set Return	Yes	Command ready	Any
Address Mode			
Locate and Read	Yes	Command ready	Not Unit 15.
Locate and Write	Yes	Command ready	Not Unit 15.
Locate and Verify	Yes	Command ready	Not Unit 15.
Spare Block	Yes	Command ready	Not Unit 15.
Request Status	Yes	Command ready	Any
Release	Yes	Command ready	Any
Release Denied	Yes	Command ready	Any
Door Lock	Yes	Command ready	Not Unit 15.
Door Unlock	Yes	Command ready	Not Unit 15.
Validate Key	Yes	Command ready	Not Unit 15.
Download	Yes	Command ready	Any
Initiate Diagnostic	Yes	Command ready	Any
Describe	Yes	Command ready	Any
Set Format Options	Yes	Command ready	Not Unit 15.
Initialize Media	Yes	Command ready	Not Unit 15.
HP-IB Parity Check	No	Command ready	Not applicable
Read Loopback	No	Command ready	Not applicable
Write Loopback	No	Command ready	Not applicable
Channel Independent			
Clear	No	Command ready*	Any
Cancel	No	Command ready*	Any
Amigo Clear	No	Command ready*	Not applicable
Stand alone QSTAT.	No	Command ready	Not applicable
Identify	No	Any	Not applicable
Universal Clear	No	Command ready*	Not applicable

\* These commands can be used to abort an Initialize Media command.  
 They are allowed any time during the Initialize Media command and will cause  
 the initialization to be aborted.

Figure 3-7. Execution Requirements.

Address State	Secondary (without parity)	Opcode	# of other parameter bytes	Command
L major	65H	2XH	None	Set Unit
L major	65H	40-47H	None	Set Volume
L major	65H	10H	6 bytes	Set Address
L major	65H	18H	4 bytes	Set Length
L major	65H	34H	None	No Op
L major	65H	39H	2 bytes	Set RPS (No Op)
L major	65H	3BH	1 byte	Set Release (No Op)
L major	65H	3EH	8 bytes	Set Status Mask
L major	65H	48H	1 byte	Set Return Addressing Mode (Only single vector allowed)
L major	65H	00H	None	Locate and Read
L major	65H	02H	None	Locate and Write
L major	65H	04H	None	Locate and Verify
L major	65H	06H	1 byte	Spare Block
L major	65H	0DH	None	Request Status
L major	65H	0EH	None	Release (No Op)
L major	65H	0FH	None	Release Denied (No Op)
L major	65H	31H	2 bytes	Validate Key
L major	65H	31H	2 bytes	Set Format Options
L major	65H	31H	6 bytes	Download
L major	65H	33H	3 bytes	Initiate Diagnostic
L major	65H	35H	None	Describe
L major	65H	37H	2 bytes	Initialize Media
L major	65H	4CH	None	Door Unlock
L major	65H	4DH	None	Door Lock
L major	72H	01H	1 byte	HP-IB Parity Checking
L major	72H	02H	4 bytes	Read Loopback
L major	72H	03H	4 bytes	Write Loopback
L major	72H	08H	None	Channel Independent Clear
L major	72H	09H	None	Cancel
L major	72H	--	--	Transparent execution (listen) Write Loopback execution
L major	6EH	--	--	Normal execution (listen) Write execution phase Download execution phase
L major	70H	--	1 byte	Amigo Clear (1 parameter byte + SDC)
T major	70H	--	1 byte	Reporting phase (Stand alone QSTAT)
T major	6EH	--	--	Normal execution (talk) Read execution phase Describe execution phase Request Status execution
T major	72H	--	--	Transparent execution (talk) Read Loopback execution
T minor DEC	ADDRS	--	--	Identify
	--	--	--	Universal Device Clear

Figure 3-8. Secondaries and Opcodes.

## 3.8 COMPLEMENTARY COMMANDS

Complementary commands are those like Set Unit, Set Volume, Set Address, Set Length, and Set Status Mask which can be included before a Real Time, General Purpose, and Diagnostic command. They can also be given alone, either singly or as a group of complementaries. For example the Set Unit command could be sent by itself during a command message. The Set Unit and Set Volume opcodes and any other complementaries could be sent together in one command message. Another approach would be to send some complementaries and have the last opcode be a Real Time, Diagnostic, or General Purpose command.

CS/80 defined special meanings for the complementary commands when they were given alone as opposed to being given with a Real Time, Diagnostic, or General Purpose command. SS/80 does not do this. In SS/80, any time the host gives a complementary command, it changes the target value and that change stays in effect until the host gives the same complementary again or does one of the clear commands.

For example, if the host does a Set Length to 256 bytes as part of a Locate and Read command. The target length will stay at 256 until another Set Length command or one of the clears. If the host gave the Set Length command all by itself and not part of a Real Time command, it would still have the same affect. In CS/80 the Set Length included with the Locate and Read would only be in effect during that command and then would return back to what it had been before. SS/80 doesn't do this.

The exception to this is the Set Address command where the peripheral increments the target address as it reads or writes through the data.

For the fastest decoding, and for the commands to work successfully, the complementaries must be in the following order:

```
Set Unit  
Set Volume  
Set Address  
Set Length  
(Next comes the Locate and Read, Locate and Write,  
etc.)
```

The Set Unit command has to be the first complementary if it is included. The complementaries are executed in the order they are given. The Set Volume complementary must follow the Set Unit and precede the Set Address and Set Length, otherwise, the Set Length and Set Address would be for the previously chosen volume. The order shown is the logical order and it is the order that must be used. See the Set Unit and Set Volume command descriptions for an explanation of how target addresses and lengths are related to units and volumes.

Any complementaries given should be included only once per transaction.

No Op can be placed between any of these complementaries. Any of the complementaries can be left out, but those which remain must be in this order. This is the order for fastest decoding. Complementaries other than those listed above can be placed in any order.

When the peripheral's command decoder is working its way through the complementary commands, it will stop at the first one which is erroneous, or on the first illegal opcode it detects. The remaining commands following the error or illegal opcode will not be executed.

Whenever the host does a command and gets a QSTAT=1 in the reporting phase, the host should do a Request Status command to find out what happened. There is no holdoff if QSTAT=1, but the errors will be less meaningful on the next command. The QSTAT=1 will remain and be reported in each reporting phase until it is cleared by the Request Status command.

The last byte of the command string must always be tagged with EOI. If the last byte does not have EOI, a Message Length error will be generated.

## 3.9 UNIT 15

The controller on the peripheral device is referred to as Unit 15. For example, if the device consists of both a flexible and a fixed disc drive, the fixed disc drive might be Unit 0, the flexible disc drive would then be Unit 1, but the processor or controller is always Unit 15.

The concept of Unit 15 is used on certain commands like Initiate Diagnostic and Channel Independent Clear. An Initiate Diagnostic to Unit 15 will result in each unit of the device doing the Initiate Diagnostic command. An Initiate Diagnostic to Unit 0 will result in only Unit 0 doing the command. A Channel Independent Clear to Unit 15 will cause every unit in the device to do a clear. A Channel Independent Clear to unit 0 or 1 would cause only unit 0 or 1 to do the clear.

Unit 15 has a full set of status bytes, which can be masked just like any other unit's status bytes. Unit 15 has a QSTAT value kept in RAM. At power-on the QSTAT of Unit 15 is 2 and it must be seen by the host before most commands directed to Unit 15 will be executed.

Figure 3-7, "Execution Requirements," shows which commands can be addressed to Unit 15 and which cannot. If you set the target unit to Unit 15 and give it a command which it cannot execute, an illegal opcode error will result.

If you are writing a driver that supports only flexible and fixed disc drives, it is advisable **not** to give any commands to Unit 15. There is no need to use it for small flexible and fixed discs.

## 3.10 REPORTING PHASE

It is very important that the host always follow the correct sequence of secondaries in a transaction. **If the command includes a Reporting Phase, the host must always complete a reporting message.** This means the host must send the secondary for the reporting message and get back the QSTAT byte from the peripheral. If the correct sequencing in a transaction is not followed, a Message Sequence error will be set. (The Message Sequence error is set only if there are no reject or fault errors set. A fault error causes the peripheral to go to the Reporting Phase. This may then create a Message Sequence error which is just a result of the fault error and the fault error is what is important.)

## 3.11 TIMEOUTS

Timeouts are supposed to notify the user that the hardware has failed rather than just letting the system hang. This is admirable. However, timeouts have very often been set incorrectly with the result that perfectly good peripherals can not be used on certain hosts because they can't meet all the absurd timeouts. An example of this is a host which set a timeout of 20 minutes for a 20 Megabyte drive to do an Initialize Media command. When a 40 Megabyte drive came out, it was not usable on this host because it took more than 20 minutes to complete the Initialize Media command. **The host should not use timeouts unless it is absolutely necessary. Any timeouts used should be set to conservatively high values.** No performance is sacrificed by having long timeouts. There is no rule that says it is the duty of the host to tell the user immediately that his peripheral is dead. Bad news can wait.

To have truly parametric present and future drivers, the timeouts used on devices must be parametric, or be fixed for all SS/80 devices. When a 20 Megabyte fixed disc is upgraded to a 40 Megabyte fixed disc, no timeout problems should be encountered and no changes to the driver should be necessary.

## Table of SS/80 Timeouts

This manual specifies all the possible timeout values for SS/80 commands. These timeout values are given either in the following table or as part of the information a peripheral returns when given the Describe command. The table is organized in order of normal transaction structure, i.e., command, execution, report. After the commands that follow this structure are discussed, information is given for commands that do not follow normal transaction structure and for Power-on.

SS/80 devices will guarantee a response time in certain situations listed below. The long commands like Initialize Media will require using infinite timeouts. Infinite timeouts are being used on several existing products and have not been a problem. To make sure the device is present, it is best to do a Loopback command or some other simple command before doing the long command. The peripheral's microprocessor will take care of timing out all hardware. On most devices a long Initialize Media command can be aborted with a Clear or Cancel command.

The worst case timeouts specified below are indeed worst case. For example, normally an SS/80 device will respond to a command phase secondary and take the parameter bytes in a few milliseconds. However, to allow some protection for future devices with characteristics that are presently unknown, the timeouts are set high. When the timeout the host should use is specified as 5 seconds, the SS/80 peripheral will actually complete in 4 seconds, so there is some margin. Similarly, the host timeout of 25 milliseconds will be done by the peripheral in 23 milliseconds. Only four timeouts, then, are recommended for use with SS/80: 25 milliseconds, 5 seconds, 60 seconds, and infinite.

### COMMAND PHASE

Command message      An SS/80 peripheral will accept the command phase secondary, disable parallel poll, and accept all the parameters including the complementaries and any command opcode in 5 seconds worst case. Typical response time will be in milliseconds.

Certain commands require longer Command Phase timeouts. They are listed below.

COMMAND	TIMEOUT TO USE	TYPICAL RESPONSE TIME
Any complementary given alone or a group of complementaries.	5 seconds total	Milliseconds
Locate and Read	U15-U16, Access time parameter in Describe	Depends on device
Locate and Write	U15-U16, Access time parameter in Describe	Depends on device
Validate Key	U15-U16, Access time parameter in Describe	Depends on device
Describe	Infinite	Depends on device
Request Status	5 seconds total	Milliseconds

Release	5 seconds total	Milliseconds
Release Denied	5 seconds total	Milliseconds
Download	5 seconds total	Milliseconds
Spare Block	Infinite	Depends on device.
Initialize Media	Infinite	Depends on device.
Initiate Diagnostic (0,1,0)	60 seconds	Depends on device.
Locate and Verify	Infinite	Depends on length and device.

EXECUTION An SS/80 peripheral will accept the execution phase secondary and disable parallel poll in any execution phase in 25 milliseconds worst case.

Execution phase data The first block of data will be sent to the host or received from the host within the optimal retry time specified in U13-U14 in the Describe command.

Successive blocks of data will be sent to the host or accepted from the host within the optimal retry time specified in U13-U14 in the Describe command.

After the last block has been accepted from the host, parallel poll will be enabled within the optimal retry time specified in U13-U14 in the Describe command.

After the last block has been sent to the host on a read, parallel poll will be enabled within the optimal retry time specified in U13-U14 in the Describe command. U13-U14 is used to permit all execution phase timeouts to be the same and to allow time for any cleanup tasks after for any cleanup tasks after a Locate and Read.

The above execution phase timing covers all commands which have an execution phase.

REPORT PHASE An SS/80 peripheral will accept the reporting phase secondary, disable parallel poll and send the QSTAT byte within 25 milliseconds worst case.

OTHER COMMANDS The remaining commands do not follow the normal transaction structure. Their timeouts are specified below. All these timeouts assume the peripheral is in a command ready state (command phase) and any previous transaction has been completed.

HP-IB Parity Checking	The SS/80 peripheral will accept the secondary, disable parallel poll and accept the two bytes in 5 seconds worst case. Typical response time is in milliseconds.
Read and Write Loopback	The SS/80 peripheral will accept the secondary, disable parallel poll, and then either accept any bytes sent or send the number specified within 5 seconds worst case. Typical response time depends on the number of bytes sent.
Universal Clear, Channel Independent Clear, and Amigo Clear	The SS/80 peripheral will accept the secondary, (if there is one), disable parallel poll response and accept any parameter bytes within 5 seconds worst case. The actual time to do the clear is long and depends on the device, so a host driver should use a timeout of 60 seconds while waiting for the peripheral to enable parallel poll.
Identify	The SS/80 peripheral will realize there is an Identify command and send the first byte of identification within 25 milliseconds worst case. The second byte will then follow in another 25 milliseconds worst case.
Cancel	The SS/80 peripheral will accept the secondary, disable parallel poll, accept the 1 or 2 parameter bytes, do the cancel and enable parallel poll within 5 seconds worst case. Typical response time will be in milliseconds.
<b>NOTE</b>	
<p>Some SS/80 devices allow the host to clear or cancel out of an Initialize Media command. These peripheral devices will check the HP-IB periodically for a clear or Cancel command. The frequency at which they check the bus is dependent on the peripheral device. This will allow the user to abort a format operation if desired. Support for this feature is device dependent; if the device cannot allow a format to be aborted, it will not allow you to Clear or Cancel out of it.</p>	
POWER-ON	An SS/80 device will be ready to respond to commands from the host after power is turned on in 60 seconds worst case. Typical times depend on the device.

#### NOTE

The Series 200 boot ROM requires that the peripheral take the command phase secondary off the bus, disable parallel poll, and accept the first parameter byte within 25 ms. Each following parameter byte or opcode must be accepted within 25 milliseconds. For Describe, Request Status, and Identify commands, the data bytes sent from the device to the host must meet a 25 ms timeout on each byte.

### 3.12 FORMATTING PROCEDURE FOR SS/80 DEVICES

To format an SS/80 medium, the following steps should be performed by the host:

1. The host should select the appropriate unit and volume using the Set Unit and Set Volume commands.
2. The host should give the Set Format Options command with an option byte of 0FFH. If this command fails with a Parameter bounds error, the host knows that the device has no format options and the host should go to step 3. If the Set Format Options passes, the host should request an option byte from the user and do another Set Format Options command using the user supplied options byte. See the Set Format Options command for more information.
3. Next the host should ask the user for the interleave factor to use.

An interleave of 0 is changed to 1 by the peripheral. An interleave greater than the largest meaningful interleave on a peripheral is changed to the largest meaningful interleave, as determined by the peripheral device. In most cases a disc with X available sectors per track would have a largest meaningful interleave of X-1. For example, a disc with 32 sectors per track uses interleaves between 1 and 31. Interleaves greater than 31 for this disc are converted to 31. Some peripherals work differently and may set the largest meaningful interleave to a value other than X-1. In any case the peripheral will adjust the interleave specified by the host to a largest meaningful interleave if the host tries to set an interleave which is too large.

4. The host next gives the Initialize Media command with the user-supplied interleave factor and with the Initialize Media option byte equal to 00H. A default value should be provided for users who do not know what interleave to use. The host would then do a Describe command and use whatever interleave is in the V13 byte of Describe for the default value. See the Initialize Media command for more information. The default value of interleave returned in V13 of Describe is set up for a specific host and may not be the optimum interleave for all hosts.

The Initialize Media command in SS/80 does the entire process of formatting, verifying, and sparing of bad sectors. The host can issue this command and wait for the device to finish. The device will enable its parallel poll response when it is done. If the reporting phase of the Initialize Media command returns QSTAT=0, then the medium is ready for the host to use. The disc is now available for use by the host; no further verification of the medium is needed.

If the initialization procedure fails for some reason, then QSTAT will equal 1 and some status error bit will be set. Some possible status errors returned from the Initialize Media command are as follows:

No Spares Available	The medium has too many defective areas to pass initialization. The medium is bad and not usable.
Unit fault	There is a hardware failure with the unit. This unit is not usable.
Media Wear	Medium initialized successfully but almost all the spares are used up. It is still O.K. to continue.
Not Ready	There is no medium in the drive. (Flexible disc)
Autosparsing Invoked	Some SS/80 devices set this bit during Initialization if any sparing was done. The initialization was successful.
Write Protected	The medium is write protected so the Initialization was not performed.

Other error conditions might be set and they should be self-explanatory.

Since some devices have two units, Unit 0 and Unit 1. Make sure you are initializing the correct unit and volume.

The peripheral's processor always times out all hardware. The host should use an infinite timeout because as the capacities of fixed discs increase, format times also increase.

The host should not do an Initiate Diagnostic command before the Initialize Media command. The SS/80 peripheral will do the selftest itself before doing the Initialize Media. If the host does the Initiate Diagnostic command, the fault LED will come on momentarily during this command and then go off when it passes. The appearance to the user is that the drive failed momentarily during the Initialization.

Block interleaving allows the transfer rate of a device to be matched most efficiently with that of the host computer connected to it. A host computer cannot always process blocks of data as fast as they are presented by the disc. Often by the time the host computer is ready for another block, the data head has already passed that particular block on the disc, and a time delay or latency equal to as much as one revolution of the disc is incurred. Block interleaving allows the data to be staggered or interleaved by one or more blocks. Block interleaving reduces inherent latencies which are characteristic of disc drive memories without extensive internal buffering.

Interleaving, however, is not the total solution and the device can help improve performance by using its buffer memory correctly. This makes the device less sensitive to the interleave factor. How this is done depends on the device and how sophisticated its low cost hardware is. The ability to transfer data to the host over the HP-IB while at the same time the device is filling its buffer is highly advantageous. The ability of the device to have a minimum overhead time between successive blocks read would allow a slow host to handshake bytes and still run at a lower interleave factor. This allows the host increased throughput without having expensive DMA hardware on the host side.

#### Calculating the Interleave Factor

To get maximum performance from low cost SS/80 devices it is important to choose the correct interleave factor. The only foolproof method to get the best performance is to initialize the media with different interleave factors and measure the performance. Do not use the Latency Induced bit to set interleave in SS/80, because it does not work. If there is some doubt as to what value to use, it is much better to pick a value too large. If the host misses interleave, the performance will drop dramatically because only as many blocks as are buffered will be transferred per revolution.

There is no good way to calculate the interleave factor using the information from the Describe command. The Describe command does not tell you whether the peripheral can do concurrent DMA, that is, whether the peripheral can read data into the buffer from the drive at the same time as it is sending data out over the HP-IB. Transfer rates on the HP-IB are usually directional, but C3,C4 in Describe gives only one value and that value depends on the interaction between the host and peripheral. A host could transfer at 800 Kbytes per second and the peripheral at 750 Kbytes per second, but when they work together the result may be 400 Kbytes per second. Calculations can give an approximate answer, but usually not the optimum one. Only trying an application with different interleave factors will result in the best performance.

On all current SS/80 devices it is very important to set the interleave factor correctly to get maximum performance. In the future it is hoped that lower RAM and hardware costs will allow for future low cost controllers that will make performance less sensitive to the interleave factor.

### 3.13 SPARING STRATEGY FOR SS/80 DEVICES

One of the guiding principles in SS/80 is that certain items should be taken care of by the peripheral and not by the host. Sparing is one of these. Sparing strategy is a question of data integrity. It is intimately related to the peripheral and should be handled by the peripheral whenever possible.

All SS/80 devices spare bad blocks or bad tracks during the Initialize Media command. SS/80 devices may also do run time sparing or "autosparing." Unless the Autosparing Invoked bit is set, autosparing is completely transparent to the host. Since Autosparing Invoked is an information status bit, it is most likely masked by the host and would never be seen anyway. Most hosts mask the information status bits so they never get set and never cause QSTAT=1. Ideally it would be better if the host masked the information status bits like Recoverable Data and Latency Induced but returned to the user information status bits like Media Wear and Autosparing Invoked. Media Wear is something the user should know about.

During the Initialize Media command all bad portions of the medium are spared. What if a bad block occurs because of damage or interference after Initialization? The SS/80 peripheral will do all that is possible to recover the data, but if the error is hard and too large to be corrected, the Unrecoverable Data bit will be set. It will not happen often but it has to be dealt with.

If the Unrecoverable Data bit is set during a read, the host could do retries; however, since the peripheral will already have done extensive retries, the host needn't bother. **The host does not have to do retries on data errors.** When the host sees the Unrecoverable Data bit set during reads, it has two options:

1. One is to re-write the data. If the problem was noise during the writing process, simply re-writing will correct the problem. This assumes the data has been backed up on another medium so it is available.
2. If the medium has been physically damaged the host needs to re-initialize that volume. During the initialization process the damaged portion of the medium will be spared. After initialization, the host data can be written back onto the medium. If the medium is a flexible disc, it is better to replace the medium rather than to continue re-initialization, because flexible discs subject to wear.

Although SS/80 peripherals are very reliable, they can have problems and any data that must not be lost should be backed up.

### **3.14 MEDIA WEAR STATUS BIT**

An SS/80 device with media that can wear out will set the Media Wear informational status bit if the medium is wearing out. The driver should inform the user that "The medium is wearing out and should be replaced." This bit is an informational status bit but is important and should not be masked. As the medium wears out the ferrite particles generated could damage the head.

The Media Wear bit may also be set after an Initialize Media command. This means that the medium was initialized but so many spares had to be used that there are dangerously few left. In this case it is a warning signal and no action has to be done other than possibly warning the user of the state of the medium.

### **3.15 CLEARS**

There are four clears used by SS/80 devices on the HP-IB: The Interface Clear (IFC), Universal Clear, Amigo Clear, and Channel Independent Clear. Each has a particular use:

The **Interface Clear** is an HP-IB clear which clears the HP-IB chip but does nothing to the drives or their target parameters like address, length, volume, etc. If the host puts a secondary on the bus and the peripheral is not powered up yet, the handshake for the secondary will not be completed. The host in this case could do an IFC to remove the secondary from the bus. (Since Interface Clear is purely an HP-IB command, it is not in the command listings in Chapter 4. It is included in the HP-IB manual listed in Chapter 1, p. 1-1).

The **Universal Clear** clears all devices attached to the HP-IB bus.

The **Amigo Clear** will clear all units of the addressed device that receives the command.

A **Channel Independent Clear** to Unit 15 of the addressed device will clear all units in that device.

For more information on Universal Clear, Amigo Clear, and Channel Independent Clear, see the individual command listings.

# COMMAND DESCRIPTIONS

CHAPTER

4

## 4.1 COMMANDS SUPPORTED

REAL TIME COMMANDS:	LOCATE AND READ LOCATE AND WRITE
GENERAL PURPOSE:	DESCRIBE INITIALIZE MEDIA LOCATE AND VERIFY RELEASE (NO OP) RELEASE DENIED (NO OP) SPARE BLOCK DOOR LOCK DOOR UNLOCK VALIDATE KEY SET FORMAT OPTIONS DOWNLOAD
COMPLEMENTARY:	SET UNIT SET VOLUME SET ADDRESS SET LENGTH SET STATUS MASK SET RPS (NO OP) SET RELEASE (NO OP) SET RETURN ADDRESSING MODE NO OP
TRANSPARENT:	UNIVERSAL DEVICE CLEAR AMIGO CLEAR CANCEL CHANNEL INDEPENDENT CLEAR IDENTIFY READ LOOPBACK WRITE LOOPBACK HP-IB PARITY CHECKING
DIAGNOSTIC:	INITIATE DIAGNOSTIC REQUEST STATUS

Each SS/80 peripheral must respond to all the commands listed above.

Certain commands like Set RPS are accepted by the SS/80 device but no action is taken. These are "dummy" commands which are treated as "No ops" and no error bits are set. They exist for compatibility reasons. The simplest, smallest drivers which only talk to SS/80 devices should not use these "dummy" commands.

Drivers for low cost hosts should pick only the few commands they need and forget about the rest. They should pick a subset of SS/80.

Just as constraints are placed on which commands can be used, SS/80 also specifies which status bits must be used and which must not. See the Request Status command for which status bits can be set on SS/80 devices.

## 4.2 COMMANDS NOT SUPPORTED

The following is a list of CS/80 commands which are not supported in SS/80. New drivers should not use these commands with SS/80 devices. If these commands are given, an Illegal Opcode error status bit will be set.

Command:	Comments:
COLD LOAD READ	Identical to LOCATE AND READ so it is not needed in Subset 80.
WRITE FILE MARK	Tape command, not supported by fixed and flexible discs.
SET BLOCK DISPLACEMENT	Not needed on low cost peripheral.
SET BURST	Not needed on low cost peripheral.
SET RETRY TIME	The SS/80 peripheral determines the appropriate number of retries and the error recovery algorithm.
SET OPTIONS	Tape command, not supported by fixed and flexible discs.
COPY DATA	Not supported by flexible and small fixed discs.
SELECTED DEVICE CLEAR	This is not possible with the HP 8291A the way most SS/80 devices use it. Additional hardware would be needed and the extra cost isn't justified. (Since phases of a transaction are secondaries, the HP 8291A must respond to secondary addressing. The chip does not become addressed until a primary and secondary address are received. An SDC all by itself has a primary address followed by the SDC. The HP 8291A doesn't see it because without a secondary address the HP 8291A is not yet correctly addressed and ignores the SDC.)
UNLOAD	Tape command.

## **4.3 COMMAND DESCRIPTIONS**

The remainder of this document describes each SS/80 command. For information about these listings and about the use of the command set, see Chapter 3.

The commands are listed in alphabetical order.



**HP-IB SEQUENCE:**

< P 0 1 1 1 1 1 >	ATN	Unlisten from host
< P 1 0 a d d r s >	ATN	Host addresses itself to talk.
< P 0 1 A D D R S >	ATN	Host addresses device at address ADDRS to listen.
< P 1 1 1 0 0 0 0 >	ATN	Amigo Clear secondary from the host
	DPPR	Device disables parallel poll response
< X X X X X X X A >	EOI	Control byte tagged with EOI from the host A = HP-IB parity check bit A = 1 Enable parity check A = 0 Disable parity check
< P 0 0 0 0 1 0 0 >	ATN	Selected Device clear primary
.		
.		Clear being done by device
.		
	EPPR	Device enables parallel poll response when done
< P 0 1 1 1 1 1 >	ATN	Unlisten from host.(The host could send this unlisten any time after the SDC primary. It doesn't have to occur here.)
		The host should make sure the device has enabled its parallel poll response before doing the optional reporting phase message or starting the next command.

TYPE: Transparent

TRANSACTION FLOW: Not Applicable

**DESCRIPTION:**

The Amigo Clear command has the same effect on an SS/80 device as the Universal Clear command or the Channel Independent Clear to Unit 15. The Amigo Clear causes all units of the device to do a clear, including Unit 15.

## **AMIGO CLEAR**

**(Continued)**

The Amigo Clear command does the following:

1. It aborts the current operation at the earliest opportunity such that no data corruption can take place.
2. The channel options set by the HP-IB Parity Checking command are not cleared during a clear. The Amigo Clear will enable or disable parity checking based on the control byte. SS/80 devices will do parity checking on commands only if their hardware allows them to. If it does not, the A bit in the control byte will be ignored.
3. All complementary parameters and targets will be reset to their power-on values:
  - A. The selected unit becomes Unit 0.
  - B. The selected volume becomes Volume 0.
  - C. The target address becomes block 0.
  - D. The target length becomes -1.
  - E. The Status Mask is cleared so no bits are masked.
  - F. Each unit will set any Set Format options back to their power-on default values.
4. The status report bits are cleared. All status bits are cleared unless the Diagnostic Result bit is set. If the Diagnostic Result bit is set, then no status bits will be cleared. The only way to clear the status if the Diagnostic Result bit is set is to do a Request Status command. This ensures that the host will see that the peripheral failed a diagnostic which means something is wrong. The Power Fail bit is always cleared, so a clear will make QSTAT go from 2 to either 0 or 1.
5. QSTAT is set to indicate whether or not status should be requested. QSTAT will be 1 after a Clear only if the Diagnostic Result bit is set.
6. The Amigo Clear is a "hard" clear so the device should perform a recalibration or restore to cylinder 0. This step is time consuming and may take many seconds.
7. The peripheral device may want to reset its HP-IB chip. This is risky and not recommended. The Untalk and Unlisten after the Amigo Clear may hang if the chip is reset before it can complete the handshake. It is nice to reset the chip in case it is causing the problem; however, resetting the chip can cause problems itself. SS/80 peripherals are better off not resetting their HP-IB chips. SS/80 peripherals do not read the HP-IB address switch during a clear. The HP-IB address switch is only read by SS/80 devices at power-on.
8. The peripheral device lastly enters an optional reporting state. It will accept any command from the host. It is recommended that the host always do a reporting message at the end of every command.

SS/80 devices do not recognize the Selected Device Clear (SDC) command unless it is given as part of the Amigo Clear. Do not use isolated SDC's on SS/80 devices.

On an Initialize Media command, an SS/80 device will periodically check the bus for a clear or a cancel Command which can be used to abort the initialization.

The reporting message after the Amigo Clear is optional but is highly recommended. It is possible for an Amigo Clear to return a QSTAT of 1 if the Diagnostic Result status bit is set or if the peripheral hardware fails during the restore or recalibration.

**VARIATIONS FROM CS/80:**

- The HP-IB parity check bit in the control byte is treated as a No Op if the SS/80 device's HP-IB chip can't do parity checking.
- Some CS/80 devices like the HP 7908 will allow a clear to clear the Diagnostic Result bit if it has been returned to the host by some other unit of the device.



## HP-IB SEQUENCE:

< P 0 1 1 1 1 1 >	ATN	Unlisten from host
< P 1 0 a d d r s >	ATN	Host addresses itself to talk.
< P 0 1 A D D R S >	ATN	Host addresses device at address ADDRS to listen.
< P 1 1 1 0 0 1 0 >	ATN	Secondary for a transparent command.
	DPPR	Peripheral disables parallel poll response.
[ 0 0 1 0 Y Y Y Y ]		Optional Set Unit command from the host. host. YYYY = unit number. "[" means optional.
< 0 0 0 0 1 0 0 1 >	EOI	Opcode for a Cancel command from the host.
< P 0 1 1 1 1 1 >	ATN	Unlisten from the host. This can occur anytime after the Cancel opcode is sent.
		Peripheral does the cancel at this time.
	EPPR	Peripheral enables parallel poll when the cancel is completed and the peripheral wants a reporting message.
		The host should make sure the peripheral has enabled its parallel poll response before sending the reporting phase secondary.

## \*\*\*\*\* REPORTING MESSAGE \*\*\*\*\*

< P 0 1 1 1 1 1 >	ATN	Unlisten from the host.
< P 0 1 a d d r s >	ATN	Host addresses itself to listen.
< P 1 0 A D D R S >	ATN	Host addresses device at address ADDRS to talk.
< P 1 1 1 0 0 0 0 >	ATN	Reporting phase message secondary from the host.
	DPPR	Peripheral disables parallel poll.
- Q S T A T -	EOI	Peripheral sends the QSTAT byte to the host.

**CANCEL**  
**(Continued)**

< P 1 0 1 1 1 1 >      ATN      Host untalks the peripheral.  
Can occur anytime after QSTAT  
is accepted by host.

TYPE:      Transparent

TRANSACTION FLOW:      Command, Report.

**DESCRIPTION:**

This command causes graceful termination of the transaction, leaving it in the reporting phase.

An example might be useful. Suppose that the user decides to format a fixed disc. This process may take 30 minutes. Just after the format procedure has started, the user changes his or her mind and somehow issues the Cancel command. The peripheral device is doing a format, but periodically it looks for a Clear or Cancel from the host. When it sees the Cancel, the peripheral will stop what it is doing, and get ready for the reporting message of the Cancel command. The Message Length error will not be set. Neither will the Message Sequence error bit. The device just stops what it is doing and gets ready to receive a report phase secondary. There is no clearing of status as is done during a Clear. The Cancel command is telling the peripheral device to "stop what you are doing at the first opportunity and don't complain".

The reporting message is required in the Cancel command. If the host does a Cancel without doing the reporting phase, a Message Sequence error will be generated.

Cancel commands are polled for during Initialize Media commands so a peripheral device may not respond immediately. If the peripheral has enabled parallel poll and is waiting for the secondary for the next phase of a transaction, then Cancel will be done immediately. This is the most advantageous situation. If the device is not polling and is in the command ready phase, there is really nothing to cancel, but the device will go into reporting phase anyway just to be consistent. Devices which allow you to cancel out of a long format will check for a Clear or Cancel command periodically. The frequency at which the peripheral checks for the Clear or Cancel is device dependent.

If a command which involves data transfer is cancelled, there may have already been a status error like Unrecoverable Data set. These errors are not changed and will be seen in the reporting phase. The only errors which are suppressed (cleared) are Message Sequence and Message Length errors.

The first opcode in the above HP-IB SEQUENCE is optional. It is a Set Unit complementary which may be included with the Cancel command. No other complementaries can be included with a Cancel command.

Some future device may not allow you to cancel out of a format. Because of this, the ability to Cancel a format is a feature which most SS/80 devices will have, but is not a definite requirement.

Do not start one unit doing an Initialize Media and then try to give a Cancel to another unit. Transactions to a device can not be overlapped. The first unit will cancel out of the Initialize Media command when you do the Cancel even though the Cancel is to the unit not being formatted. You must complete each transaction with one unit or volume of a device before starting the next transaction with a different unit or volume of the same device.

**VARIATIONS FROM CS/80:** None

# CHANNEL INDEPENDENT CLEAR

## HP-IB SEQUENCE:

< P 0 1 1 1 1 1 >	ATN	Unlisten from the host.
< P 1 0 a d d r s >	ATN	Host addresses itself to be the talker.
< P 0 1 A D D R S >	ATN	Host addresses device to listen.
< P 1 1 1 0 0 1 0 >	ATN	Secondary for a transparent command.
	DPPR	Peripheral disables parallel poll response.
[ 0 0 1 0 Y Y Y Y ]		Optional Set Unit command from the host. YYYY = unit number. "[ ]" means optional. YYYY = 1111 is device controller.
< 0 0 0 0 1 0 0 0 >	EOI	Opcode for a Channel Independent Clear command from the host.
< P 0 1 1 1 1 1 >	ATN	Unlisten from the host. This can occur any time after the Channel Independent Clear command is received.
		Peripheral does the clear at this time.
	EPPR	Peripheral enables parallel poll response when the clear is done and wants a reporting message.
		The host should make sure the peripheral has enabled its parallel poll response before sending the reporting phase secondary.

\*\*\*\*\* REPORTING MESSAGE \*\*\*\*\*

< P 0 1 1 1 1 1 >	ATN	Unlisten from the host
< P 0 1 a d d r s >	ATN	Host addresses itself to listen.

## CHANNEL INDEPENDENT CLEAR

(Continued)

< P 1 0 A D D R S >	ATN	Host addresses device to talk.
< P 1 1 1 0 0 0 0 >	ATN	Reporting phase message secondary from the host.
	DPPR	Peripheral disables parallel poll response.
- Q S T A T -	EOI	Peripheral sends the QSTAT byte to the host.
< P 1 0 1 1 1 1 >	ATN	Host untalks the peripheral. This can occur anytime after QSTAT is accepted by the host.

TYPE: Transparent

TRANSACTION FLOW: Command, Report (report is optional but should be done)

### DESCRIPTION:

This is the recommended clear command in SS/80 for clearing individual units or the device controller (Unit 15). If the device controller is specified, all units it controls will be cleared. If another unit is specified, only that unit will be cleared. The device does the following steps during a Channel Independent Clear:

1. It aborts the current operation at the earliest opportunity such that no data corruption can take place.
2. The channel options set by the HP-IB Parity Checking command are not cleared during a Channel Independent Clear.
3. All complementary parameters and targets will be reset to their power-on values:
  - A. If the clear is directed to Unit 15, then after this Channel Independent Clear the selected unit is Unit 0. If the Channel Independent Clear is directed to a unit other than Unit 15, then that unit remains the target unit.
  - B. The selected volume becomes Volume 0.
  - C. The target address becomes block 0.
  - D. The target length becomes -1.
  - E. The Status Mask is cleared so no bits are masked.
  - F. If the clear is directed to Unit 15, then all units will set their values for any Set Format options back to the default values. If the clear is directed to a unit other than Unit 15, then only that unit will set its Set Format options back to its default value.
4. The status report bits are cleared. All status bits are cleared unless the Diagnostic Result bit is set. If the Diagnostic Result bit is set, then no status bits will be cleared. The only way to clear the status if the Diagnostic Result bit is set is to do a Request Status command. This ensures that the host will see that the peripheral failed a diagnostic which means something is wrong. The Power Fail bit is always cleared, so a clear will make QSTAT go from 2 to either 0 or 1.

5. QSTAT is set to indicate whether or not status should be requested. QSTAT will be 1 after a Clear only if the Diagnostic Result bit is set.
6. The Channel Independent Clear is a "hard" clear if it is directed to Unit 15. In this case the device should perform a recalibration or restore to cylinder 0. If the Channel Independent Clear is directed to a unit other than 15, then it is a "soft" clear and the device should not recalibrate to cylinder 0. The host expects "soft" clears to be much faster than "hard" ones because the recalibration or restore is the most time consuming part of a clear.
7. The peripheral device may want to reset its HP-IB chip. This is risky and not recommended. The Untalk and Unlisten after this clear may hang if the chip is reset before it can Complete the handshake. It is nice to reset the chip in case it is causing the problem; however, resetting the chip can cause problems itself. SS/80 peripherals are better off not resetting their HP-IB chips. SS/80 devices do not read the HP-IB address switch during a clear. The HP-IB address switch is only read during power-on by SS/80 devices.
8. The peripheral device lastly enters an optional reporting state. It will accept any command from the host. It is recommended that the host always do a reporting message at the end of every command.

On an Initialize Media command, an SS/80 device will periodically check the bus for a clear or a Cancel command, which can be used to abort the initialization.

Note that the first data byte, [0010YYYY] is in "[ ]" brackets rather than "<>" brackets. This is because this byte is optional and does not have to be included.

**VARIATIONS FROM CS/80:** None



**FORMAT:**                   **Describe opcode = <00110101>**

**TYPE:**                   **General Purpose**

**TRANSACTION FLOW:**       **Command, Execution, Report.**

**DESCRIPTION:**

This command provides enough information about the device to allow it to be configured into a system without the host having prior knowledge about this device type. The device will return a maximum of 256 bytes of information in the execution message. The last byte of this message will be tagged with EOI, so that fewer than 256 bytes can be transferred. There are three types of description fields returned: the controller field (5 bytes), the unit field (19 bytes), and the volume field (13 bytes).

If the device has two units, each with a single volume, and the Describe command is given to Unit 15, the order of the bytes returned is as follows:

```
controller field
Unit 0 field
Volume 0 field of Unit 0
Unit 1 field
Volume 0 field of Unit 1
```

If the device has two units, Unit 0 having 3 volumes and Unit 1 having 1 volume, and the Describe command is given to Unit 15, the order of the bytes returned is as follows:

```
controller field
Unit 0 field
Volume 0 field of Unit 0
Volume 1 field of Unit 0
Volume 2 field of Unit 0
Unit 1 field
Volume 0 field of Unit 1
```

Because the number of bytes returned is not known when the Describe is given to Unit 15, it is best never to do a Describe to Unit 15. In fact, host drivers written for flexible and small fixed discs should ignore Unit 15 totally.

If the device has two units, each with a single volume, and the Describe command is given to Unit 0, the order of the bytes returned is as follows:

<b>controller field</b>	<b>5 bytes</b>
<b>Unit 0 field</b>	<b>19 bytes</b>
<b>Volume 0 field of Unit 0</b>	<b>13 bytes</b>

If the device has two units, with Unit 1 selected, and Unit 1 has 6 volumes with Volume 5 selected, the Describe to Unit 1 will return the following:

<b>controller field</b>	<b>5 bytes</b>
<b>Unit 1 field</b>	<b>19 bytes</b>
<b>Volume 5 of Unit 1</b>	<b>13 bytes</b>

**A DESCRIBE COMMAND TO A UNIT OTHER THAN UNIT 15 WILL ALWAYS RETURN 37 BYTES.**  
The host driver can be assured if it gives the Describe command to unit 0,1,2, etc. but not Unit 15, that it

## **DESCRIBE** **(Continued)**

will always return 37 bytes. The unit field will be for the selected or target unit and the volume field will be for the selected or target volume.

### **NOTE**

- Remember that if a Describe command is given when QSTAT=2, only one byte of value 1 tagged with EOI will be returned to the host. After the host sees the QSTAT of 2, then the Describe and all other commands will be executed. After detecting a QSTAT=2, the host should do a Request Status or clear to make QSTAT=0 and then do a Describe command to find out the properties of the new medium.
- When the Describe command is given to Unit 15, the QSTAT returned in the reporting message will be 0 even if a new medium was detected on one of the units.

If the peripheral device is unaddressed without having sent the last byte tagged with EOI, a Message Length error is generated. This is generally true of all execution messages. The correct termination in SS/80 is EOI. The host must take all the Describe bytes. The last one is always tagged with EOI.

To get all the necessary information for the execution message, the peripheral may have to do an access to the disc and read information from it. Thus the Describe command would have timeouts similar to a Locate and Read command; however, when the host is timing out the Describe command, it doesn't know what the timeouts in U15-U16 are. Thus an infinite timeout should be used. See "Timeouts," p. 3-19.

For a flexible disc the Describe fields will reflect the format of the disc in the drive. If no disc is in the drive, V1-V6 will reflect the format of the last disc, but the address field, V7-V12 will be zero. After power-on and before any flexible disc has been loaded, V7-V12 will be zero and V1-V6 will represent a default disc.

## **DESCRIBE COMMAND SUMMARY**

The following list summarizes the three fields of the DESCRIBE command. The fields are Controller Description, Unit Description, and Volume Description.

\*\*\*\*\* CONTROLLER DESCRIPTION FIELD \*\*\*\*\*  
( C1 - C5, 5-byte field )

C1,C2 = Installed unit byte; 1 bit for each unit. (Unit 0 = LSB.) Remember that Unit 15 is always present so the MSB of C1 is always set.

Examples:

2 units C1=10000000, C2=00000011  
3 units C1=10000000, C2=00000111  
4 units C1=10000000, C2=00001111

C3,C4 = Instantaneous transfer rate in thousands of bytes per second. This is found by measuring the time between bytes when the peripheral is transferring over HP-IB to or

from the peripheral's data buffer. If the rate depends on direction, the lower rate (slower) should be used.

C5 = Controller Type

0 = CS/80 integrated single unit controller.  
1 = CS/80 integrated multi-unit controller.  
2 = CS/80 integrated multi-port controller.  
  
4 = SS/80 integrated single unit controller.  
5 = SS/80 integrated multi-unit controller.  
6 = SS/80 integrated multi-port controller.  
( Thus if bit 2 is set, the device is an SS/80 device.)

**\*\*\*\*\* UNIT DESCRIPTION FIELD \*\*\*\*\***

U1 = Generic Unit Type. This tells the host whether the selected unit is a fixed disc drive, flexible disc drive, tape, or "dumb" (i.e., unintelligent) flexible disc drive of the sort . described in Section 3.7. See p. 3-15.

0 = Fixed disc  
1 = Flexible or removable disc.  
2 = Tape (Random access format like the HP 9144A)

129 = "Dumb" (i.e., unintelligent) flexible disc drive. If bit 7 of U1 is set, then the device has no sensor to tell whether a new medium has been loaded. It will set QSTAT=2 only at power-on. See p. 3-15.

U2-U4 = Device number. Represents actual HP product number:  
XX XX XY (BCD Coded, 2 digits per byte). XXXXX =  
product number. Y = option.  
Example: HP 9133 would be 09 13 30

U5-U6 = Number of bytes per block.

U7 = Number of blocks which can be buffered.

U8 = 0 Recommended burst size. SS/80 devices will set  
U8 = 0. SS/80 devices do not support burst mode.

U9-U10 = Block time in microseconds (Time is from beginning of one block to beginning of next.)

U11-U12 = Continuous average transfer rate for long (full volume) transfer in thousands of bytes per second.  
This should be a measured value using the device's fastest host.

**DESCRIBE**  
**(Continued)**

U13-U14 = Optimal retry time in 10's of milliseconds.  
This is the worst case time it takes the device to read or write 1 block. It is the time it takes the device to do retries, error correction, recalibration, re-seeking, etc.

U15-U16 = Access time parameter in 10's of milliseconds.(Maximum time starting in the command message when the device accepts the last parameter byte or opcode and going until the device enables parallel poll requesting the next phase of the transaction be started.) See the "Timeouts," p. 3-19. This is the worst case time to seek to the target cylinder. It includes seek retries, recalibration, etc.

U17 = Maximum Interleave factor. If the controller or media do not support interleave factors, then U17 should be returned as 0. Example: A disc with 32 sectors would set this to 31. A tape would set U17 to 0.

U18 = Fixed volume byte; one bit per volume (set if fixed); Volume 0 = LSB. This gives the number of fixed volumes for this unit.  
  
Examples: 3 fixed volumes, U18 = 00000111  
1 fixed volume, U18 = 00000001  
8 fixed volumes, U18 = 11111111  
No fixed volumes, U18= 00000000

U19 = Removable volume byte; one bit per volume (set if removable); Volume 0 = LSB. This gives the number of removable volumes for this unit.  
  
Examples:  
4 removable volumes, U19 = 00001111  
1 removable volume, U19 = 00000001  
No removable volumes,U19 = 00000000  
7 removable volumes, U19 = 01111111

\*\*\*\*\* VOLUME DESCRIPTION FIELD \*\*\*\*\*

V1-V3 = Maximum value of cylinder address vector.

V4 = Maximum value of the head address vector.

V5-V6 = Maximum value of sector address vector.

NOTE: SS/80 devices use only single vector addressing. The maximum values for three vector addressing will be returned in V1 through V6 if they are meaningful. If the device's mode of sparing corrupts cylinder boundaries, then that

product should set V1-V6 to 0. Any product which does supply V1-V6 should make sure that the following expression is satisfied:

$$(V1, V2, V3 +1)(V4 + 1)(V5, V6+1) = V7, V8, V9, V10, V11, V12 +1$$

The host should never do a Set Address or Set Return Addressing Mode in any mode except single vector. The 3 vector address in V1-V6 is not changed to zero if no medium is loaded.

V7-V12 = Maximum value of single vector address in blocks.

The maximum value of single vector address in blocks will be set to 0 if there is no medium loaded. If a medium is loaded, it will give the maximum value of the single vector address in blocks for that particular medium which is loaded.

V13 = Current Interleave Factor.

If the drive knows its current interleave factor, it puts it in V13. This is always the case, even if it cannot maintain that interleave factor.

If the medium is unformatted or there is no medium loaded, the peripheral will return a default factor for a specific host. That is, if the peripheral were sold mainly for Product A, then the host would return, as a default, the optimum interleave for Product A if it were unformatted. V13 is then the default interleave for the host to use if it doesn't already know the optimum value. See "Formatting Procedure for SS/80 Devices," p. 3-23.

#### VARIATIONS FROM CS/80:

- In the controller description field, SS/80 devices are denoted by having Controller Type 4,5, or 6. This means bit 2 is set for SS/80 devices. This is the way you tell SS/80 devices from CS/80 devices.
- CS/80 specifies that the access time parameter in U15- U16 applies for read and write commands only. In SS/80 this access time parameter is used in other commands which access the disc. See "Timeouts," p. 3-19.
- SS/80 sets bit 7 of U1 to designate a "dumb" (i.e., unintelligent) flexible disc drive. (See p. 3-15.) CS/80 always has bit 7 equal to 0.



**FORMAT:** Door Lock opcode = <01001101>

**TYPE:** General Purpose

**TRANSACTION FLOW:** Command, Report

**DESCRIPTION:**

In SS/80 this command instructs the target unit of the device to lock its door. If the device does not have a door lock, the device sets an Illegal Opcode status error bit. The host can determine if the SS/80 device has door locks by giving this command and seeing if an Illegal Opcode results.

This command should be given to any unit except Unit 15. Unit 15 is the controller and it doesn't have any doors. If this command is given to Unit 15, an Illegal Opcode error will result.

**VARIATIONS FROM CS/80:**

- This command does not exist in CS/80. If you give this command to a CS/80 device, it will set an Illegal Opcode error. Only give this command to SS/80 devices. You can tell a device is an SS/80 device by doing a Describe command and checking that bit 2 of byte C5 is set. See the Describe command.



**FORMAT:** Door Unlock opcode = <01001100>

**TYPE:** General Purpose

**TRANSACTION FLOW:** Command, Report

**DESCRIPTION:**

In SS/80 this command instructs the target unit of the device to unlock its door. If the device does not have a door lock, the device sets an Illegal Opcode status error bit. The host can determine if the SS/80 device has door locks by giving this command and seeing if an Illegal Opcode results.

This command should be given to any unit except Unit 15. Unit 15 is the controller and it doesn't have any doors. If this command is given to Unit 15, an Illegal Opcode error will result.

**VARIATIONS FROM CS/80:**

- This command does not exist in CS/80. If you give this command to a CS/80 device it will set an Illegal Opcode error. Only give this command to SS/80 devices. You can tell a device is an SS/80 device by doing a Describe command and checking that bit 2 of byte C5 is set. See the Describe command.



## HP-IB SEQUENCE:

< P 0 1 1 1 1 1 >	ATN	Unlisten from the host
< P 1 0 a d d r s >	ATN	Host addresses itself to be the talker.
< P 0 1 A D D R S >	ATN	Host tells device at address "ADDRS" to listen.
< P 1 1 0 0 1 0 1 >	ATN	Command phase secondary
	DPPR	Device disables parallel poll response.
[ 0 to n complementary commands ]		Device takes in data from host.
< 0 0 1 1 0 0 0 1 >		Opcode for Initiate Utility where device receives data.
< 1 1 1 1 0 0 1 0 > = F2H < 1 0 1 0 0 1 0 1 > = A5H		These two bytes specify a Download command.
< A A A A B B B B > < C C C C D D D D > < E E E E Y Y Y Y >		Device number. Represents actual HP HP product number. AB CD EY is BCD coded, 2 digits per byte. ABCDE = product number. Y = option. Example: The HP 9122 is coded 09 12 20.
< F F F F F F F F >	EOI	The Download revision number in unsigned binary. Six parameter bytes follow the opcode.
		Device decodes command.
	EPPR	Device enables parallel poll response when ready for execution message.
		The host should make sure the peripheral has its parallel poll response enabled before sending the execution phase secondary.
< P 0 1 1 1 1 1 >		Host tells device to unlisten. (Can be sent by host anytime after last parameter byte.) Doesn't have to be here.
< P 1 0 a d d r s >	ATN	Host addresses itself to be the talker.
< P 0 1 A D D R S >	ATN	Primary listen from host

**DOWNLOAD**  
(Continued)

< P 1 1 0 1 1 1 0 >	ATN	Execution phase secondary
	DPPR	Device disables parallel poll response
< from 1 to a download buffers worth of data>		Device received up to a download buffers worth of data from the host.
< last byte >	EOI	Last byte has EOI.
...		Device executes the code. From this point on, the protocol depends on the code executed. The test controller and peripheral device will agree on what to do from here. There can be multiple execution messages.
	EPPR	Device enables parallel poll when entire command is done.
		The host should make sure the device device has enabled its parallel poll response before sending the reporting phase secondary.

\*\*\*\*\* REPORTING MESSAGE \*\*\*\*\*

< P 0 1 1 1 1 1 >	ATN	Unlisten from the host
< P 0 1 a d d r s >	ATN	Host addresses itself to listen.
< P 1 0 A D D R S >	ATN	Host addresses device to talk.
< P 1 1 1 0 0 0 0 >	ATN	Reporting phase message secondary from the host.
	DPPR	Peripheral disables parallel poll response.
- Q S T A T -	EOI	Peripheral sends the QSTAT byte to the host.
< P 1 0 1 1 1 1 >	ATN	Host untalks the peripheral. This can This can occur anytime after QSTAT is accepted by the host.

**TYPE:** Diagnostic  
**TRANSACTION FLOW:** Command, Execution, whatever, Report  
**DESCRIPTION:**

Host drivers should not use this command!

The structure of the command is that of an Initiate Utility command with the parameters specifying a download. The parameters also contain the HP product number, option number, and a download revision number. The device receives the downloaded code in the execution message. After receiving the last byte, which is tagged with EOI, the protocol between the test controller and peripheral device is defined by prior agreement.

During the execution phase, the test controller will give the device a number of bytes, not exceeding the number of bytes it can hold in its download buffer. The download buffer may vary in size from peripheral to peripheral. The last byte of the transfer will be tagged with EOI. The device will then do a subroutine branch to the start of this code. Execution from then on is dependent on this downloaded code.

Thus the downloaded code could have the peripheral do an error rate test and send the results back to the host in a second execution message. The protocol is prearranged between the host and peripheral so that anything can be done. One example would be a download which returns the drive's spare table. After the host sends the code to the peripheral, the peripheral executes the code. This code tells the peripheral to get the spare table and send it to the host. The host in the mean time is addressing itself to listen and telling the peripheral to talk. After receiving the spare table, the host Unlistens and Untalks the peripheral. The peripheral returns to report phase and the host then gives a standard reporting message. The host should always do a reporting message and get back QSTAT at the end of every command. This is the only way to know that the command completed successfully.

SS/80 devices can use the parameter field bytes P7 through P10 to put diagnostic information. The bytes P7 through P10 are a good place for SS/80 peripherals to put any special information which might be useful to service or manufacturing. The normal status error bits can be set with P7-P10 providing an expanded description of the error. Host drivers should ignore the parameter field bytes P7 through P10.

The purpose of the Download command is to allow service and test engineers complete freedom to write any routine they need without it having to go into the ROM of a low cost host.

If the parameters are not the correct HP product code or Download revision number, a Parameter Bounds status error will set, and the report phase will be entered. The Download command can be given to any unit including Unit 15.

In writing download routines for a peripheral, it is best to create a jump table of the modules that may be called in downloads. Each time the code is changed, a module's absolute address will change. However, the position of the module in the table will remain the same. By using an indirect branch to the subroutine, the downloads don't have to change each time the firmware is changed and relinked.

#### VARIATIONS FROM CS/80:

- This is the CS/80 Initiate Utility command.



# HP-IB PARITY CHECKING

## HP-IB SEQUENCE:

< P 0 1 1 1 1 1 >	ATN	Unlisten from the host
< P 1 0 a d d r s >	ATN	Host addresses itself to be the talker.
< P 0 1 A D D R S >	ATN	Host tells device at address "ADDRS" to listen
< P 1 1 1 0 0 1 0 >	ATN	Secondary for a transparent command from the host.
	DPPR	Device disables parallel poll response.
< 0 0 0 0 0 0 0 1 >		Opcode for HP-IB Parity Checking command.
< 0 0 0 0 0 0 S V >	EOI	Second data byte from host tagged with EOI.
< P 0 1 1 1 1 1 >	ATN	Unlisten from the host

where:

S = 0 Disable SRQ during poll (power-on state)  
S = 1 Enable SRQ during poll  
V = 0 Parity Checking disabled (power-on state)  
V = 1 Parity Checking enabled

TYPE: Transparent

TRANSACTION FLOW: Not applicable

DESCRIPTION:

Parity checking on commands is done by SS/80 devices only if their hardware allows it. If it does not, the V bit is ignored.

If the S bit is set, the Service Request (SRQ) line will be asserted whenever the device enables parallel poll. The SRQ line is disabled when the device disables parallel poll. Unfortunately, SS/80 devices which use the HP 8291A chip do not handle the serial poll sequence well; therefore host drivers should use parallel poll instead of serial poll. The problem stems from the use of secondaries as phases of a transaction and not just extended addresses as IEEE 488 intended. The HP 8291A is set up to respond to secondary addressing. To be fully addressed, the peripheral must get a primary address and a secondary one. The Universal Command SPE is given by the host. The host then sets ATN and sends the primary address to the device to address it to talk. The HP 8291A needs a secondary address as well as a primary one but what secondary should be used? The result is the host waiting for a byte from the peripheral while the peripheral waits for a secondary from the host. There are ways around this. One is to always do an Identify before the SPE. This Identify must end in Untalk, rather than MTA (the host talk address). All the Identify command does is get

## **HP-IB PARITY CHECKING**

**(Continued)**

the peripheral in a fully addressed state. Then it will take the SPE and give the correct byte back to the host. It is best to avoid serial poll and use parallel poll on SS/80 devices. Serial polling could be used if only one peripheral is enabled to pull on SRQ at a time. In this case no SPE command is required since it is known which device pulled on SRQ.

Note that parallel poll is not enabled by the device at the end of this command and the transaction sequence does not apply.

### **VARIATIONS FROM CS/80:**

- Most of the commands which do not follow the transaction sequence are remnants of an earlier disc protocol called the Amigo protocol.
- Most CS/80 devices use the PHI or ABI HP-IB chips which were designed to handle secondaries in the manner of the Amigo and CS/80 protocols. With these chips serial poll works fine. It is possible that SS/80 devices will change HP-IB chips in the future; for now, however, it is best to not use serial poll on SS/80 devices.

**HP-IB SEQUENCE:**

< P 0 1 1 1 1 1 >	ATN	Unlisten from the host
< P 0 1 a d d r s >	ATN	Host addresses itself to listen.
< P 1 0 1 1 1 1 >	ATN	Untalk (Here it is Talk 31) The device at address 31 is addressed to talk.
< P 1 1 A D D R S >	ATN	Secondary with device address
- 0 0 0 0 0 0 1 0 - - T T T T T T T T -	EOI	2 identification bytes supplied by the device 00000010 = ID byte 1 TTTTTTTT = ID byte 2 and is device specific.
< P 1 0 a d d r s >	ATN	Talk controller. The host sends MTA (My talk address) which automatically untalks everyone but itself. "addrs" is the host controller's address.

**TYPE:** Unique

**TRANSACTION FLOW:** Not applicable.

**DESCRIPTION:**

Identify is a special-case Amigo protocol command used by the host at power-on to identify the devices connected to the bus. Each device returns a two-byte identity code which the host can use to configure itself. The two bytes are returned for as long as the host will accept them. All CS/80 and SS/80 devices return the value of 2 (00000010) in ID byte 1, and the product type code in ID byte 2.

The Identify command was developed for the Amigo protocol which preceded CS/80. It is a way for the host to identify all the devices on the bus. Normally an Untalk command would tell all the devices on the bus that none is addressed to talk. However, the Amigo protocol defined Untalk to be Talk 31 and each Amigo, CS/80 and SS/80 device must have address 31 as its minor address. SS/80 devices have two addresses, a major address as defined by the HP-IB switch and a minor address which is always 31. The minor address is only for the Identify command. Thus when the host sends Untalk, all the IEEE 488 devices stop talking and all the HP discs get ready to talk. They first must receive the secondary that has their major address in it.

If the device gets the secondary but the address is not for it, it will complete the handshake and do nothing. If the device gets the secondary and the secondary contains its primary address, the device will send two bytes back to the host. It sends a byte of value 2 followed by another product-dependent byte. The peripheral will continue to send these 2 bytes as long as the host asks for them. The second byte is always tagged with EOI. Amigo devices do not use 2 as the first byte, so this is one way to separate Amigo devices from CS/80 and SS/80 devices. For CS/80 and SS/80 devices, the first byte is always a 2.

## **IDENTIFY** **(Continued)**

Normally in a command which has the peripheral device talk, the host ends the command with an Untalk. However, the Untalk was defined by HP to be a Primary Talk for this command. Thus an Untalk at the end of the command will not unaddress SS/80 devices. The CS/80 manual shows the termination as Talk 30. This came about because the PHI or ABI chip has to be at address 30 if it is a controller. A generalization of this would be for the host to terminate the Identify by addressing itself to talk. Since there can be only one talker on HP-IB, all other devices are untalked.

**VARIATIONS FROM CS/80:** None

**CAUTION**

Execution of the Initialize Media command destroys all user data on the selected unit. Before executing the Initialize Media command, make certain that the proper unit and volume have been selected. Failure to do so may result in the loss of needed data.

**FORMAT:**                   Initialize Media opcode = <00110111>

Parameter byte 1 = <00000YYY>

Parameter byte 2 = <P2>

YYY = Initialize options

For discs:

000 = initialize selected volume retaining all  
factory and field spares

(The above Initialize option is the one  
which drivers must use.)

001 = initialize selected volume retaining only factory spares

010 = initialize selected volume retaining no spares

P2 = Block interleave byte (binary number)

**TYPE:**                   General Purpose

**TRANSACTION FLOW:**     Command, Report.

**DESCRIPTION:**

**CAUTION**

HOST DRIVERS MUST ALWAYS USE INITIALIZE OPTION YYY=000. The other options should be used with care because they will erase spared blocks which were previously found to be bad. The loss of factory spares can render a device unusable. The 001 and 010 options should be used only by service engineers or others who understand what they are doing. If the device has multiple volumes and there is a need to use Initialize options 1 or 2, then the device should first be configured as a single volume device before doing the initialization. All data on the disc will be lost. Spares are a property of the drive and not how it is divided into volumes. Initialize options 1 and 2 do not retain all spares and that affects all volumes.

## **INITIALIZE MEDIA**

**(Continued)**

SS/80 devices will accept the initialize options byte and do what is appropriate for that device.

A device which does not distinguish between factory and field spares should retain spares unless option 2 is specified.

In SS/80 this command does the complete initialization process that prepares the medium for use by the host. SS/80 devices do multiple passes of writing, verifying, and sparing, returning only when all sparing has been done. The host can then issue this command and if the QSTAT returned is 0, the medium is ready for use. The host should not verify the medium for this is done by the device. When this command is done, the host can write the directory and start using the medium. This means that this command may take a long time to complete. See "Timeouts," p. 3-19.

If the Initialize command fails because there are not enough spares, the No Spares Available error is returned. If there were enough spares, but only an insufficiently small number remained, the Media Wear error bit will be set.

The process of initialization should be in the province of the peripheral since it understands the sensitivities and failure modes of the device better than the host driver writer does.

A "0" interleave factor has the same value as a factor of "1". If a block interleave factor greater than the maximum allowable (as specified in the Describe command) is specified, the interleave value defaults to an appropriate value, usually the maximum interleave. The maximum interleave for a disc is chosen by each product and can vary. For many SS/80 drives the maximum interleave is X-1 where the drive has X usable sectors on a track.

The data pattern left on the medium after executing this command is device dependent; the host should never rely on a device to fill each block with zeros or some other byte.

It is important to read also "Formatting Procedure for SS/80 Devices," p. 3-23.

It is possible on most SS/80 devices to abort a long Initialize Media command by sending a clear or a Cancel command. If this is done, the medium will be left in an unfinished state and will have to be initialized again at some other time to be useful.

### **VARIATIONS FROM CS/80:**

- SS/80 devices interpret this command to mean that the device does the entire process of multiple passes of formatting, writing worse case patterns, verifying, and sparing, and in general preparing the media for use. This is not the case for the HP 7908 which had an involved Initialization procedure. See "Formatting Procedure for SS/80 Devices," p. 3-23, for the recommended procedure for initializing an SS/80 device.

# INITIATE DIAGNOSTIC

FORMAT:                   Initiate Diagnostic opcode = <00110011>  
                          Parameter byte 1 = <00000000>  
                          Parameter byte 2 = <00000001>  
                          Parameter byte 3 = <00000000>

TYPE:                   Diagnostic

TRANSACTION FLOW:      Command, Report.

## DESCRIPTION:

If this command is given to Unit 15 (the controller), then all units of the device will be tested. This command can also be given to a unit other than Unit 15. In this case only the specified unit is tested.

In SS/80 this command directs the device or unit to perform a complete diagnostic selftest. If the medium is present and formatted, the device will do a test which involves reads and writes on the medium. Writes will be done only if the medium is not write protected. The test will not destroy any user data. A special area of the medium will be reserved by the peripheral. The host doesn't have to worry about this. If there is no medium present, everything except reading and writing is tested. The test will not fail if there is no medium. It will also not fail if the medium is not formatted. If the medium is unformatted or not present, the peripheral will test what it can but do no reads or writes.

If the command is addressed to Unit 15, then failure of any unit will result in the Diagnostic Result bit being set in the unit's status bytes and in Unit 15's status bytes. The parameter bytes P1 through P6 which contain diagnostic information will be copied into Unit 15's parameter bytes also. For example, if the command is addressed to Unit 15 and Unit 1 fails, then the error status in Unit 1 will be copied into Unit 15. Unit 0 in this case will not have the Diagnostic Result bit set. This is because the command was addressed to Unit 15, and we want to make sure the host sees that one of the units failed. If the first unit tested in this case fails, then the next unit will not be tested.

If this command is addressed to a specific unit other than Unit 15, then, if the unit fails selftest, the Diagnostic Result bit is only set in that unit's status bytes. Unit 15's status will be unaffected. Only the specific unit which is told to do the Initiate Diagnostic will set its Diagnostic Result bit and set P1-P6 accordingly if the selftests fails.

The Diagnostic result information stored in P1 through P6 when the Diagnostic Result status bit is set is as follows:

- P1 - P5   These bytes contain information for test and service telling which component failed. The codes are device specific and are not the same as those used on HP 7908.
- P6       This byte has the first unit which failed. This allows the host driver to display that "Unit 0 failed selftest", or "Unit 1 failed selftest", if the command was given to Unit 15. Another way to get this information is to check the status of each unit on the device or only selftest one unit at a time. Byte P6 is not used in CS/80

## INITIATE DIAGNOSTIC

(Continued)

so this is a departure. We feel it is important for the host to know at least which device failed, but we still don't want the host driver to get involved in specific diagnostic information.

The bytes P7 through P10 contain additional diagnostic information which should be ignored by the host driver. The bytes in P1-P6 or P7-P10 contain detailed information as to which component failed or what specific test failed. See "Timeouts," p. 3-19, for how long this command may take.

The idea of this command is to test everything possible about the device to make sure it is functioning properly. The tests involve doing a checksum on the ROM, checking the processor RAM, buffer RAM, and any controller chips present. It is difficult to check the area of RAM which contains operating parameters and not destroy them. There are also problems involved with taking the HP-IB chip off line just to test a few registers.

If you give the command to Unit 15, then the following steps should be taken:

1. Give the Initiate Diagnostic (0,1,0) command to Unit 15.
2. If QSTAT on Unit 15 is bad, then give the Request Status command to Unit 15.
3. If the Diagnostic Result bit is set, then read P6 to find out which unit is defective. If P6 is 0, then display "Unit 0 failed selftest." The unit which failed could be unit 0,1, etc. (Unit 15 will not be placed here).
4. Next you have to do a Request Status command to the unit which failed, since it also has its Diagnostic Result bit set. The Clear commands, Universal Clear, Amigo Clear, and Channel Independent Clear will never clear the Diagnostic result bit.

It is easiest to give the Initiate Diagnostic (0,1,0) to the unit you are interested in. You will see whether it failed when you get the QSTAT and don't have to worry about having the Diagnostic Result bit set in Unit 15.

Failure of selftest is very important and shouldn't be ignored. It means part of the system isn't working correctly and further use of the device may create more problems and could destroy user data. We do urge drivers to use this command to determine whether the device is in proper working condition. Only one diagnostic command was defined in SS/80 and we hope it is clear enough so that host drivers will use it.

The idea of this command in SS/80 is to do a thorough "go" or "no go" check of the device. This command is the most effective way to test an SS/80 peripheral. Using the Initiate Diagnostic command and the Read and Write Loopback commands should show up most problems with the device.

The Initiate Diagnostic command does no clearing of complementaries or status. The target address of the device may be changed during this command.

### VARIATIONS FROM CS/80:

- With the HP 7908, 7911, and 7912, this command can be given only to Unit 15. The HP 9144A allows this command to a specific unit. SS/80 allows it to any unit. In fact it is simpler to use if given to a specific unit other than Unit 15.

**INITIATE DIAGNOSTIC**  
**(Continued)**

- In SS/80 the Initiate Diagnostic (0,1,0) is the basic selftest. Most devices will support no other selftest.
- In SS/80 the parameter byte P6 is used to tell which unit failed selftest. The first unit which fails is placed in P6. P6 will not be set to 15 if the controller fails. In CS/80 P6 was not used. Thus a SS/80 driver using a CS/80 device like the HP 7908 might display that unit [P6] failed when P6 did not contain this information. CS/80 devices on the HP 7908, 7911, 7912 have P5 and P6 equal to 0. The driver could display this only with an SS/80 device if it expects to run with SS/80 and CS/80 devices. You tell whether the device is SS/80 by the byte C5 in the Describe command.
- In CS/80 for the HP 7908, 7911, and 7912, the Clear commands clear the Diagnostic result bit if it has been seen once by the host. IN SS/80 THE CLEAR COMMANDS NEVER CLEAR THE DIAGNOSTIC RESULT BIT. It must be cleared by doing a Request Status command.



**FORMAT:** Locate and Read opcode = <00000000>

**TYPE:** Real Time command

**TRANSACTION FLOW:** Normally Command, Execution, Report.  
For seek it is Command, Report.

**DESCRIPTION:**

This command locates the data indicated by the target address and transmits it to the host.

The normal sequence of events for this command is as follows: The device gets a command phase secondary. The device disables parallel poll and accepts the secondary. Following the secondary are one or more data bytes. The last data byte is the 00H opcode given above. The optional data bytes preceding this opcode are the complementary commands. The device takes in all this data, decodes it, checks that it is all proper and makes sense, and does a seek if it is needed. At this point the device re-enables parallel poll telling the host that it is ready to enter the execution phase.

See the complementary commands Set Unit, Set Volume, Set Address, and Set Length for explanations of how these work.

If a problem is found by the device, like an illegal opcode, it will go to the report phase, and an Illegal Opcode error bit will be set causing QSTAT to be set to 1. If the host asks for data at this point, a byte of value 1 tagged with EOI will be sent.

Assuming there have been no errors, after the device enables parallel poll, the host will give it the execution phase secondary. Again the device disables parallel poll and accepts the secondary. The device then reads the data into his buffer and transfers it to the host. Both sides should keep a count of how many bytes are transferred. The last byte transferred by the device to the host will be tagged with EOI.

After the last byte is transferred, the device enables parallel poll, which is effectively asking for a report phase secondary. The host then does a reporting message. The device disables parallel poll, accepts the secondary, and sends the QSTAT byte to the host. Parallel poll is not re-enabled at this point since the transaction is complete.

The length of the total data transfer is the number of bytes specified in a Set Length (Complementary) command, which may be included in the message with the Locate and Read command. If Set Length is not part of the Locate and Read command, the power-on or last Set Length value is used. See the Set Length command. If the length used is 0, then this command is actually a seek and no data is transmitted. After performing the seek the peripheral goes directly to the reporting phase and there is no execution phase. If the host mistakenly tries to do an execution phase with the length equal to 0, a Message Sequence error will be generated and the peripheral will send the host a byte of value 1 tagged with EOI.

If a data error is encountered in the course of the transfer, the peripheral will do whatever is necessary to try to acquire or recreate the data. If the data is unrecoverable, the device sends its most accurate reconstruction of the data and returns this to the host. The address of the first block of any bad data will be included with the status report returned by a Request Status command. If the length of the transfer spans several blocks of data and any block was bad, the entire transfer will take place. QSTAT=1 will notify the host that the transfer had a problem and the status bytes P1 through P6 indicate the address of the first bad block. Thus the transfer always contains the amount of data requested by the host unless the host intervenes or a hardware failure occurs. SS/80 devices will set the Unrecoverable Data status error bit if there is a data error. If another block is found to be bad, the Unrecoverable Data Overflow bit is set. Regardless of the number of data errors, the device will try to read all blocks and send back its best

## **LOCATE AND READ**

(Continued)

reconstruction of the data until the total length of the transfer has been met. If there is just one data error, the host knows where it is from the address in P1 through P6. If there is more than one error, P1 through P6 give the address of the first bad block.

If the device has a header error that can not be recovered by retries, the Unrecoverable Data error bit will be set. The buffer will still be sent to the host. Thus in SS/80 a header error is treated like a data error. The device should put information into P7-P10 to describe the exact nature of the error.

If the device suffers a hardware error where it can't read anything, or the flexible disc has been removed during the transfer, the device will terminate the transfer by sending one byte of value 1 tagged with EOI to the host. The host has to be able to detect EOI as the termination of the transfer. For a hardware error the host may only receive one byte. In such a case, the host should re-setup the target address. This is a good idea whenever there is an error during a read or write.

The host can always issue the command again if it wants to do retries. **THE HOST DOES NOT HAVE TO DO RETRIES ON UNRECOVERABLE DATA ERRORS AS THEY HAVE ALREADY BEEN TAKEN CARE OF BY THE PERIPHERAL.**

A Locate and Read operation updates the target address as explained in the Set Address command.

The length of the read can be any number of bytes. Of course, the device can read only a block at a time into its internal buffer, but the correct number of bytes will be transferred to the host. The host can do a read of 1 byte if it wants to. For the special case of a length equal to -1, see the Set Length command.

If the host includes the following complementaries in the Locate and Read or Locate and Write commands, they must be in the order shown below:

- Set Unit
- Set Volume
- Set Address
- Set Length

No Op can be placed anywhere between these complementaries and all four of them don't have to be included. There must not be a No Op following the Locate and Read opcode or an error will be generated. The above complementaries should be in this order for Locate and Read or Locate and Write commands. This doesn't affect other complementaries or other commands.

If the host wants to see whether the medium is there before a read, the host should do a Locate and Read of length 0 and not include the Set Address complementary command. This is a seek to the track the device is already on. It will result in a QSTAT of 2 being set if a new medium is loaded and a Not Ready error (QSTAT=1) if there is no medium loaded. The host will then know if the medium is there or not. The complementaries would be Set Unit, Set Volume, and Set Length to 0, with the Locate and Read opcode. Not including the Set Address complementary will cause the peripheral to use the target address which is where the head actuator is already positioned. No seek will take place.

### **NOTE**

The term "block" refers to a specific quantity of data, the amount of which is a function of the storage format of a particular mass storage device. SS/80 devices currently have block sizes of 256,512, or 1024 bytes. The device's block size can be found by doing a Describe command.

**VARIATIONS FROM CS/80:**

- SS/80 devices do not use RPS (Rotational Position Sensing). They accept this command as a No Op.
- SS/80 devices do not use burst mode.
- SS/80 devices do not allow the host to set the retry time. It is a fixed time set by each unit of the peripheral.



## LOCATE AND VERIFY

**FORMAT:** Locate and Verify opcode = <00000100>

**TYPE:** General Purpose

**TRANSACTION FLOW:** Command, Report.

**DESCRIPTION:**

This command instructs the device to perform an internal verification of a section of data to ensure that it can be read. None of the data is transferred to the host so no execution message is required. The Set Length and Set Address complementary commands are used as described elsewhere.

The verification starts at the target address and continues for the amount of data (in bytes) specified in a Set Length complementary command or in the power-on value if no Set Length command has been given. If this byte count length is not an integral multiple of the number of bytes per block the count will be rounded up to verify the entire block, since a block is the smallest quantity of data which can be verified.

Verification will terminate immediately on an unrecoverable data error. The peripheral device should NOT do read retries during a verify. This is not always possible with a device's hardware. For example, some controller chips do not let the peripheral do a read without retries. In any case, the idea is to let the host know if the previous write did not take so the host can rewrite the data. The error recovery process on the reads will then be shortened. In this manner, records which are only slightly bad (soft errors) will cause the Unrecoverable Data error bit to be set.

This command will not set Marginal Data or Recoverable Data status bits, since with the reduced number of retries and with no error correction, any marginal data will be unrecoverable.

When the Unrecoverable Data status bit is set during a Locate and Verify command, the address of the bad block will be in bytes P1-P6 of the status bits and the target address will point to the next block to read, that is the one beyond the bad block. See the Set Address command for more information about target address changes.

If this command completes with no errors, the host knows the data is well written. If an Unrecoverable Data error is set, that portion of the data should be rewritten because it didn't take. See "Formatting Procedure," p. 3-23, and "Sparing Strategy," p. 3-25.

Since in SS/80 the peripheral does the entire Initialization process, the Locate and Verify command should not be used by the host immediately after an Initialize command. It should be used after Locate and Write commands if the host wants to make sure the data was written correctly onto the medium.

Another way to check whether the data is written correctly onto the medium is to read the data back using a Locate and Read and compare it byte for byte. This will check the data written; however, the Locate and Read will use retries and error correction so it may be a better check to use Locate and Verify to see that the data was well written. Locate and Verify provides a means to see whether the data is recoverable without retries or error correction, and, since no data is transferred over the HP-IB, it is fast.

The Locate and Verify command is most often used by production to speed up error rate tests. Locate and Verify is certainly not a command that a host driver must support.

**VARIATIONS FROM CS/80:**

- SS/80 devices do not keep error logs.



**FORMAT:** Locate and Write opcode = <00000010>

**TYPE:** Real Time command

**TRANSACTION FLOW:** Normally Command, Execution, Report.  
For Seek it is Command, Report.

**DESCRIPTION:**

This command transfers data from the host to the mass storage medium. The data is placed on the medium starting at the target address.

The normal sequence of events for this command is as follows: The host sends the device a command phase secondary. The device disables parallel poll and accepts the secondary. Following the secondary are one or more data bytes. The last data byte must be the 02H opcode given above. The optional data bytes preceding the opcode are complementary commands. The device takes in all this data, decodes it, and checks it. The medium is checked at this time to see whether it is ready and not write protected. If all is correct, the device does a seek to the target address and then enables parallel poll. By enabling parallel poll, the peripheral is telling the host to send the execution message secondary to the device.

If there is a failure during decoding, like an Address Bounds error, the Address Bounds status error bit will be set causing a QSTAT of 1, parallel poll will be enabled, and the peripheral will go to reporting phase. If the host sends data bytes, the peripheral will sink them until the host is also in the reporting phase and accepts the QSTAT of 1.

See the complementary commands Set Unit, Set Volume, Set Address, and Set Length for explanations about how these work.

Assuming there is no problem, the host next sends the execution phase secondary. The device disables parallel poll and accepts the secondary. Next the host transfers data to the device. The device fills his buffer and does a write to the medium each time the buffer is full, or when the transfer is over. Thus the transfer of bytes over the bus may be interrupted periodically as the device writes onto the medium. Both the host and the peripheral should keep track of the number of bytes transferred. If a partial block is written, the remainder of the block will be filled by the peripheral by duplicating the last data byte, or by filling with zeros. For security reasons the peripheral should not write whatever was left over in the buffer onto the disc.

The last data byte the host transfers to the peripheral is tagged with EOI. After this block has been written, the device enables parallel poll to tell the host it is ready for the reporting message. The host sends the reporting phase secondary. The device disables parallel poll and accepts the secondary. The device then returns QSTAT, which will be 0 if everything worked correctly.

As with a read, once the execution phase is entered and data is being transferred, the total length will always be transferred. However, unlike a read, if the device encounters a header error and cannot write a block, it will not try to write anymore blocks. The device will set the Unrecoverable Data status error bit, set QSTAT = 1, put the address of the bad block into P1 through P6 of the status bytes and sink any data the host gives the device. The Unrecoverable Data Overflow bit will also be set, since nothing beyond the bad block will ever be written. Eventually the host will enter the report phase and see the bad QSTAT. The bytes P1 through P6 will tell the host where the transfer failed. The reporting phase is used to re-synchronize the transaction. Since the device was unable to find a header, there is probably something seriously wrong with the medium or device itself.

A Locate and Write operation updates the target address as explained in the Set Address command. In the case of a bad block, (header error), the peripheral device's target address points to the block just beyond the

## **LOCATE AND WRITE**

**(Continued)**

bad block. The address of the bad block will be in parameter bytes P1-P6. The target address always points to the next block to be done, which in this case is the one after the bad block. For example, suppose the host tells the peripheral to write 2560 bytes starting at address 100. Each block on this peripheral has 256 bytes so 10 blocks will be written. After writing two blocks, a header error is encountered. Repeated retries by the peripheral are unable to read the header correctly. This bad block is at address 102, so P1-P6 will have 102, and the target address of the peripheral will be address 103. Successful completion of this command would have resulted in the target address being 110. Thus for any errors during a Locate and Write, the host should do a Set Address command to reset the target address.

If the length is 0, then this command is a seek and there is no execution phase and no write data transferred.

If the host wants to see if the medium is there and not write protected before a write, the host should give a Locate and Write of length 0 and not include the Set Address complementary. No seek will take place. The seek is to the track the head actuator is already on. It will result in the Power Fail bit being set (QSTAT=2) if a new medium has been loaded, a Not Ready bit set (QSTAT=1) if there is no medium present, and a Write Protect bit being set (QSTAT=1) if the medium is write protected. The host then knows whether the medium is present and whether it may be written on. The complementaries would be Set Unit, Set Volume, and Set Length to 0, with the Locate and Write opcode. The Set Address complementary is not included because we don't want to move the head, only to have the controller determine whether the medium is ready to use.

### **VARIATIONS FROM CS/80:**

- SS/80 devices do not use burst mode.
- RPS is not used in SS/80.

**FORMAT:** No Op opcode = <00110100>

**TYPE:** Complementary

**TRANSACTION FLOW:** Command, Report.

**DESCRIPTION:**

This byte is disregarded if it appears as an opcode in a command message. It may be useful to align messages to word boundaries.

No Op can be placed at the beginning of a string of complementary commands, in-between complementary commands, and between the last complementary command and a Real Time, General Purpose, or Diagnostic Command. However, No Op should not be placed after the last Real Time, General Purpose, or Diagnostic Command. Doing so will cause an Illegal Parameter error to be set. A No Op could be the last byte if the command contains only complementaries.

The following examples show where No Op can and cannot be placed:

Example 1.    No Op  
              Set Unit  
              No Op  
              Set Volume  
              No Op                      Valid use of No Op  
              Set Address  
              No Op  
              Set Length  
              No Op  
              Locate and Read

Example 2.    No Op                      Valid use of No Op  
              Request Status

Example 3.    No Op                      Valid use of No Op  
              Set Length  
              No Op

Example 4    Set Unit                    INVALID use of No Op  
              Locate and Read  
              No Op                      No Op can not come  
    after a non-complementary  
    command.

VARIATIONS FROM CS/80: None



# READ LOOPBACK (Also WRITE LOOPBACK)

## NOTE

This command listing covers both READ LOOPBACK and WRITE LOOPBACK.

### HP-IB SEQUENCE:

< P 0 1 1 1 1 1 >	ATN	Unlisten from the host.
< P 1 0 a d d r s >	ATN	Host addresses itself to talk.
< P 0 1 A D D R S >	ATN	Primary listen from host. Peripheral at address ADDRS will be a listener
< P 1 1 1 0 0 1 0 >	ATN	Secondary for a transparent command.
	DPPR	Peripheral disables parallel poll response.
< 0 0 0 0 0 0 1 T >		Opcode for Read Loopback or Write Loopback. T=0 Read Loopback Test T=1 Write Loopback Test
< P 1 MSB >		4 byte length parameter (unsigned binary)
< P 2 >		Length in bytes to use in Loopback test.
< P 3 >		
< P 4 LSB >	EOI	
< P 0 1 1 1 1 1 >	ATN	Unlisten from the host.
< P J K a d d r s >	ATN	JK=10, Host will talk for Write Loopback. JK=01, Host will listen for Read Loopback.
< P J K A D D R S >	ATN	JK = 10, Primary Talk for Read Loopback. JK = 01, Primary listen for Write Loopback.
< P 1 1 1 0 0 1 0 >	ATN	Secondary for a transparent command.
	DPPR	Peripheral disables parallel poll response.

## **READ LOOPBACK WRITE LOOPBACK (Continued)**

-< X X X X X X X X X >-		Read Loopback Data goes from the peripheral to the host.
-< X X X X X X X X X >-		Write Loopback Data goes from the host to the peripheral.
.	.	The number of bytes sent is specified in the length parameter shown above.
.	.	Last byte is tagged with EOI.
-< X X X X X X X X X >-	EOI	
-< X X X X X X X X X >-	ATN	Host sends Unlisten
< P 0 1 1 1 1 1 1 >		

\*\*\*\*\* REPORTING MESSAGE \*\*\*\*\*

< P 0 1 1 1 1 1 >	ATN	Host sends Unlisten
< P 0 1 a d d r s >	ATN	Host addresses itself to listen.
< P 1 0 A D D R S >	ATN	Primary Talk from the host. Peripheral will talk
< P 1 1 1 0 0 0 0 >	ATN	Reporting phase message secondary from the host.
	DPPR	Peripheral disables parallel poll response.
- Q S T A T -	EOI	Peripheral sends the QSTAT byte to the host.
< P 1 0 1 1 1 1 >	ATN	Host untalks the peripheral.

**TYPE:** **Transparent**

**TRANSACTION FLOW:** Transparent Command, Transparent Command, Report.

**DESCRIPTION:**

Loopback is an interface test consisting of two transparent commands followed by a reporting message. It is a sequence of transparent messages which is valid only during the command phase to test data transmission reliability over the channel. That means the loopback commands should not be separated by other commands, and they should not be part of another transaction. The first transparent message specifies that a read or write loopback operation of n bytes will follow, and the second transparent message contains the test data specified by the first. The host should not wait for poll before proceeding with the loopback data message (the second message). The peripheral device will normally not enable parallel poll during this command. The loopback data always consists of the same data pattern which goes FFH, 00H, 01H, 02H. . . OFFH, 00H, 01H, etc. for the length specified. The device generates or checks the length and content of the message depending on the direction of transfer. After loopback completes, the host should do a reporting message without waiting for poll to verify success of the operation. The reporting message is optional but highly recommended.

**READ LOOPBACK  
WRITE LOOPBACK  
(Continued)**

If the loopback is successful, the device never asserts parallel poll. If something goes wrong, however, the device enters reporting phase to report the error. In this error mode the device does assert parallel poll response.

If the host improperly executes the loopback command or transparent message secondary or direction, a Message Sequence error is reported.

Another command is not allowed between the first and second messages in the loopback sequence. In SS/80 no overlapping of transactions to one device is allowed.

If any of the bytes in a Write Loopback are wrong, a Channel Parity error is reported. If the length of the loopback message was wrong, a Message Length error is reported. If the length specified is 0, then a Parameter bounds error is reported. A length of 0 for this command is not allowed. The length of the loopback can be any 4 byte unsigned binary number except zero. In the length the first byte is the most significant byte.

Loopback is designed to test the channel and data path through the device's HP-IB chip to its microprocessor. Whether or not it tests the device's DMA hardware is dependent on the device.

VARIATIONS FROM CS/80: None



**FORMAT:** Release opcode = <00001110>

**TYPE:** General Purpose

**TRANSACTION FLOW:** Command, Report

**DESCRIPTION:**

In SS/80 this command is a No Op. It is accepted and no action is taken and no errors are generated. It can be addressed to any unit.

**VARIATIONS FROM CS/80:**

- The meaning of this command in CS/80 is very different from that in SS/80. See the *CS/80 Instruction Set Programming Manual* for an explanation of Release in CS/80.



## RELEASE DENIED [NO OP]

FORMAT: Release Denied opcode = <00001111>

TYPE: General Purpose

TRANSACTION FLOW: Command, Report

### DESCRIPTION:

In SS/80 this command is a No Op. No action is taken and no error bits are set. The command can be addressed to any unit.

### VARIATIONS FROM CS/80:

- The meaning of this command in CS/80 is very different from that in SS/80. See the *CS/80 Instruction Set Programming Manual* for an explanation of Release Denied in CS/80.



**FORMAT:** Request Status opcode = <00001101>

**TYPE:** Diagnostic

**TRANSACTION FLOW:** Command, Execution, Report.

**DESCRIPTION:**

This command instructs the device to return (in an execution message) the status report. It is important to note that this command returns the status bits held in RAM but it does not update that status. This command does NOT cause an access to the disc and it does not update the Not Ready, Power Fail or any other status bits. It simply returns what status has already been set based on previous commands. See "Flexible Disc Loading and Removal," p. 3-13. The Request Status command simply transfers what status bits are already in the peripheral's RAM to the host.

In SS/80, the Request Status command returns a 20-byte status report for a unit, indicating the cumulative status of all transactions that have occurred since the status report was last cleared. The status report can be cleared only by executing a Request Status command or a Clear command. (See the Universal Device Clear and Channel Independent Clear commands.) The status report consists of a 2-byte identification field, an 8-byte error reporting field, and 10 bytes of additional information in the parameter field. The complete format of the status report for SS/80 devices is given in the Request Status Summary below.

If all 20 bytes are not accepted by the host, the status of the unit is not cleared, and a Message Length error is set.

The identification field has 2 bytes. The most significant nibble of the first byte is the volume field, the least significant nibble of the first byte is the unit field, and the second entire byte is always set to all 1's by SS/80 devices.

The 8-byte error reporting field contains four categories: Reject Errors, Fault Errors, Access Errors, and Information Errors. Each category has a 2-byte error field. All error conditions are assigned specific bit positions in one of these fields. The fault errors are the only errors which cannot be masked using the Set Status Mask command.

The content of the parameter field is dependent on the errors being reported. The parameter field contents are awarded to the error with the highest priority (lowest bit position in the error reporting field). An error that has been masked in a Set Status Mask(complementary) command will not be reported and will not generate parameters. All address parameters are reported in single vector in SS/80. See the status description which follows for more information on what P1 through P10 can contain.

Reject errors include status bits that indicate a logical error in the host's interaction with the device. Reject errors result from opcode or parameter errors in the command message or message type or length errors in any message. Typically, incorrect programming or channel malfunction is the cause of these errors.

Fault errors indicate device hardware failures.

Access errors indicate problems encountered in executing specific commands relating to such factors as uninitialized media, write-protected media, unrecoverable data, etc.

Information errors provide maintenance information to the host. Most hosts have chosen to mask information errors. If the host masks information errors, it must remember that after every clear, the mask has been cleared and the host must again issue a Set Status Mask command. See the Set Status Mask command.

## REQUEST STATUS

(Continued)

The QSTAT byte returned in the reporting message is a kind of summation of the status bits. If the Power Fail status bit is set, then QSTAT=2. If any other status bit is set, but not Power Fail, then QSTAT=1. If no status bits are set, then QSTAT=0.

With the Request Status command, the host is advised to include no complementaries. If the host includes a Set Unit command with the Request Status command, and the unit is illegal, a Module Addressing error bit will be set and the peripheral will return one byte of value 1 tagged with EOI. The peripheral's command decoder always stops at the first failure and does not continue decoding the command string. In this way an infinite loop can occur in which the host gives the Request Status with a bad Set Unit command, sees the QSTAT=1 and does another Request Status command with the bad Set Unit command included. Thus it is best to do the Request Status command with nothing in the command string except the 0DH opcode for Request Status.

The host should do retries on all reject errors and the Power fail and Re-transmit errors. Other errors do not require host retries.

### VARIATIONS FROM CS/80:

- In SS/80 the SSSSSSSS bits in the identification field are always equal to FFH, all 1's. The low cost device generally does one thing at a time and does not want the complexity of examining status bits from multiple units.
- The following fault error bits will NEVER be set by an SS/80 device:

STATUS ERROR BIT:	COMMENT:
26 = 0      Operator request	SS/80 devices never
27 = 0      Diagnostic request	request release.
28 = 0      Internal maintenance	
32 = 0      Illegal parallel operation	
48 = 0      Operator Request	SS/80 devices never
49 = 0      Diagnostic Request	request release.
50 = 0      Internal Maintenance	
58 = 0      Marginal Data	
61 = 0      Maintenance Track Overflow	SS/80 devices have no maintenance tracks

- When the Diagnostic Result bit is set, P6 contains the number of the unit which failed selftest. See the Initiate Diagnostic command description for more details. CS/80 doesn't use P6 in this way so this is a difference.

### REQUEST STATUS SUMMARY (20 BYTES)

The following lists describe the 20 bytes returned in the execution phase of the Request Status command. The bytes are grouped as follows:

- Identification Field
- Reject Errors Field
- Fault Errors Field
- Access Errors Field
- Parameter Field

\*\*\*\*\*

#### • Identification Field

	byte 1	byte 2
IDENTIFICATION FIELD	<VVVVUUUU>	<11111111>

VVVV = Volume number  
 UUUU = Unit number

\*\*\*\*\*

#### • Reject Errors Field

	byte 3	byte 4
REJECT ERRORS FIELD	< 0 0 2 0 0 5 6 7 >	<8 9 10 0 12 0 0 0>
2 = Channel Parity Error	A channel command was received without odd parity or the Loopback command failed to pass.	
5 = Illegal Opcode	An unrecognizable opcode was received or a command was given to Unit 15 which is not allowed for Unit 15.	
6 = Module Addressing	An illegal volume or unit number was specified for this device.	
7 = Address Bounds	The target address has exceeded the bounds for this device.	
8 = Parameter Bounds	A parameter (other than unit, volume, or target address) is not allowed for this device.	
9 = Illegal Parameter	A parameter field was the wrong length for the opcode preceding it.	
10 = Message Sequence	The message sequence has been violated. (Error suppressed if any	

**REQUEST STATUS  
(Continued)**

reject, fault, or access errors occurred prior to the Message Sequence error.) Another way to say this is that the peripheral received a secondary it did not expect.

12 = Message Length

The total length of the execution message differs from the current default value. The host and device didn't agree on the number of bytes transferred, or the host sent a command phase secondary followed by too many bytes, or the host didn't accept the QSTAT in a reporting phase. If this error is set, the host and peripheral most likely do not have the same target address. The host should do a Set Address command.

Of course, the peripheral should not set this bit if it caused the transfer to end prematurely.

\*\*\*\*\*

- Fault Errors Field

FAULT ERRORS FIELD < 0 17 0 19 0 0 22 0 > < 24 0 0 0 0 0 30 31 >

(NONE OF THE FAULT ERRORS ARE MASKABLE.)

17 = Cross-unit

This error occurs during copy data when activity on Unit 0 or 1 causes an error on the unit, but the command was sent to Unit 15. The cross-unit error is set on Unit 15's status to tell the host that there is a bad status on another unit as a result of the command sent to Unit 15. Unit 15's P1 through P6 gives the value of the unit with bad status. The host should then select the bad unit and find out what the real error is. See P1 through P6 below. This error is not set by SS/80 fixed or flexible disc drives.

19 = Controller Fault	A hardware fault occurred in the controller. When this bit is set it will be set in the status of the unit selected when it was detected.
22 = Unit Fault	A hardware fault occurred in the unit addressed. If something in the hardware fails, this is the bit which will be set.
24 = Diagnostic Result	The device failed selftest. See Initiate Diagnostic command for more details.
30 = Power Fail	The power to the unit failed, or a new Tape or Flexible disc was loaded. Device should be reconfigured. When this bit is set, QSTAT=2.
31 = Re-transmit	The preceding transaction should be retried. We have included this bit in SS/80 for future devices which may use a low cost version of the ABI chip which has a FIFO. Current SS/80 devices use the Intel HP 8291A and don't set this bit.

\*\*\*\*\*

- Access Errors Field

	byte 7	byte 8
ACCESS ERRORS FIELD < 0 33 34 35 36 37 0 0 > < 40 41 0 43 44 0 0 0 >		
33 = Uninitialized media		The host attempted to access unformatted media, or unusable media has been detected. This is only set by a command which accesses the medium.
34 = No Spares Available		The Spare Block command cannot be executed because there are not enough spares available, or the Initialize Media command failed because there are not enough spares, or the Spare Block command was given to a device which spares only during initialization.

**REQUEST STATUS  
(Continued)**

35 = Not Ready	The device is not ready, probably because the medium has not been loaded. Set only on commands which access the medium.
36 = Write Protect	The selected volume is write protected and the current command failed because of that. Set only on commands which access the medium and potentially write onto the medium.
37 = No Data Found	For tapes this means a block accessed during a read has not been written. For fixed and flexible discs, this means the key given by the host to the peripheral in the Validate Key command did not compare to the key written on the medium.
40 = Unrecoverable Data Overflow	There has been more than one unrecoverable data error.
41 = Unrecoverable Data	Unrecoverable data at indicated block(s). P1 through P6 will contain the address of first bad block. Status bit 40 is set if there is more than one error. This bit is set for either header or data errors.
43 = End of File	End of file encountered on file structured device.
44 = End of Volume	The host attempted to access across a volume boundary. When this occurs, the volume sets its target address back to 0.

\*\*\*\*\*

- **Information Errors Field**

INFORMATION FIELD ERRORS < 0 0 0 51 52 0 0 55 > < 0 57 0 59 0 0 0 0 >

MOST HOST DRIVERS MASK THE INFORMATION FIELD ERRORS.

51 = Media Wear  
The flexible disc or tape medium is wearing out and should be replaced. This is informational

but requires immediate action by the user to insure data is not lost. The medium should be replaced. After a Spare Block command or Initialize Media command, this bit means that the device has very few spares left. Again it should be replaced.

52 = Latency induced

This bit is set if the performance of the device is generally not up to par. The SS/80 device is doing retries, either on seeks or data. DO NOT USE THIS BIT TO DETERMINE INTERLEAVE ON SS/80 DEVICES. See "Formatting Procedure," p. 3-23, for information about interleave.

55 = Auto Sparing Invoked

This is set if the device has automatically spared a block. It can be set after an Initialize Media command to note that some sparing was done. It can also be set after a Spare Block command. Setting this bit does not change P1 through P6.

57 = Recoverable Data Overflow

More than one recoverable data error was encountered since the last status request. The data is good, it just took retries or error correction to recover it.

59 = Recoverable Data

A latency was introduced in order to recover a data error. We had to do either retries or error correction to recover the data. The data is good.

\*\*\*\*\*

• Parameter Field

PARAMETER FIELD <P1> <P2> ..... <P10>

The information in the bytes P1-P6 is controlled by the smallest numbered error bit. The error bit with the smallest number is the most serious and its information will be displayed in P1-P6. Thus the following table shows the errors which can use P1-P6 and their order of priority: (Smallest number has priority)

## REQUEST STATUS

(Continued)

#17 Cross Unit Not used by flexible or fixed discs.  
#24 Diagnostic Results  
#41 Unrecoverable Data  
#59 Recoverable Data  
No errors P1-P6 is the target address.

P1 - P6 In SS/80 these bytes have the following possible meanings:

1. If there are no errors, P1 through P6 contains the target address. It will always be single vector. A Request Status to Unit 15 will return zeros in P1-P6 since Unit 15 has no target address.
2. When the Cross-unit (copy error, tapes only) bit is set, P1 through P6 contain the encoded values of each unit that has experienced an error. It uses the format shown below:

P1	P2	P3	P4	P5	P6	
00H	FFH	FFH	FFH	FFH	FFH	Unit 0 has the error
01H	FFH	FFH	FFH	FFH	FFH	Unit 1 has the error
00H	01H	FFH	FFH	FFH	FFH	Both Unit 0 and Unit 1 have an error

The sequence of events would be this: The host gives a Copy Data command and gets back a QSTAT=1. The host gives a Request Status command and sees the Cross-unit bit set. Next the host should read P1 to see which unit is bad and also check P2 to see which other unit may have had a problem. The host would then do a Request Status to the bad unit or units to get the actual failure. If the faulty unit was the controller (Unit 15) then the Cross-unit bit wouldn't have been set and only the Controller Fault bit would have been set in the status of all units.

3. If the Diagnostic Result bit is set, P1 through P6 will contain the following information:

P1 - P5 Device specific diagnostic information which should be ignored by a low cost host driver.

P6 In SS/80 this byte contains the unit which failed.

For a more complete explanation of the status after a diagnostic, see the Initiate Diagnostic command.

4. When the Unrecoverable Data bit is set, P1 through P6 contains the single vector address of the bad block.
5. When the Recoverable Data bit is set, P1 through P6 contains the single vector address of the recoverable block.
6. After a Spare Block command, P1 through P6 contains the beginning single vector address of the reformatted area. This is only true immediately after a Spare Block command.

P7 - P10 In SS/80 these bytes usually contain device dependent diagnostic information which is of no interest to the host driver. However, after a Spare Block command, P7-P10 contain the length in blocks of the reformatted area. See the Spare Block command.

Peripherals should set only 1 or 2 bits at a time since it is difficult for the controller to handle multiple bits. For a new flexible disc that is also write protected, it is advisable for the peripheral first to set the Power Fail bit (QSTAT=2) when the flexible disc is first detected, and then on the next access command set the Write Protect bit (QSTAT=1).

The Cross Unit error bit was described above, but will never be set by flexible or fixed discs, so if your driver does not include tapes, it can ignore the Cross Unit bit. The Spare Block command adds confusion to the setting of errors. The Spare Block command is not recommended for new drivers so this confusion can be eliminated by not using the Spare Block command.



**FORMAT:** Set Address opcode = <00010000>  
                   6 parameter bytes,  
                   <P1> <P2> <P3> <P4> <P5> <P6>  
                   MSB ..... LSB  
                   Only single vector addressing is allowed. Single vector format: 6-byte unsigned binary number

**TYPE:** Complementary

**TRANSACTION FLOW:** Command, Report.

**DESCRIPTION:**

This command is used to set the unit's target address. The 6-byte number is the address in blocks. A block refers to a specific quantity of data, the amount of which is a function of a particular mass storage device. The size of a unit's block is specified in the response to the Describe command. Typical block sizes are 256 bytes, 512 bytes, or 1024 bytes.

Whenever this command is given it changes the unit's target address. The unit keeps a "target" address in RAM which is changed by this command. The target address can also be changed when it is incremented as the result of a Locate and Read, Locate and Write, or Locate and Verify command. It will be changed to all zeros during a clear command.

Whether the Set Address complementary is given alone, with only other complementaries, or as part of a Real Time, General Purpose, or Diagnostic command message, it changes the unit's target address.

The power-on value of the target address is always 0.

The Set Address is unlike other complementary commands in that it is updated by any command that accesses data, and does not revert to a prior value when another accessing command is sent. This allows sequential data accessing.

Upon completion of a Locate and Read which uses the target address, the target address will point to the block after the last block accessed during that transaction, whether or not there was any unrecoverable data. If the transaction was successful, the target address will be in parameter bytes P1 through P6 of the status. If there was an unrecoverable data error, P1 through P6 will have the single vector address of the bad data, but the target address will point to the next block after the last one read.

In general then, the target address always points to the next block to be processed.

This is also true in Locate and Verify. The target address will point to the next block to be read. If there is an error, it points to the block following the bad one. The single vector address of the bad block is placed into P1 through P6.

In Locate and Write, the target address will again point to the next block to write. If there was a header error, the target address points to the next block to write and the bad block will be in P1 through P6. In the case of a header error, the write is not continued so the next block to write is the one after the bad block. If there are no errors, the target address can be obtained from the parameter bytes P1 through P6 returned in a Request Status command.

## SET ADDRESS

(Continued)

Since the target address is in blocks and the length is specified in bytes, a read which ends on the first byte of block n or any byte within n will result in the target address being set to block n+1.

Only single vector addressing can be used with SS/80 devices. The 6-byte address field of the Set Address command is treated as one large unsigned binary number. Every addressable block of storage is located via this 6-byte address. This simplifies device addressing for systems which are not interested in the particular device's configuration. The peripheral organizes his cylinder, head, and sector addresses on a disc drive such that access to sequential sectors is provided with maximum performance. (Increment sector, head, then cylinder).

The address of a peripheral wraps around when it gets to the end. Thus if the host does a Locate and Read, Locate and Verify, or Locate and Write of the very last block on the volume, the target address of the peripheral is then 0. If the Length had been enough bytes for 2 blocks and the target address was the very last block, then the read, write, and verify commands would have failed causing, an End of Volume error. The target address in this case is also set back to 0.

Some examples may be helpful in understanding the Set Address and Set Length commands. In the following the block size is always 256 bytes and read, write, and verify refer to the Locate and Read, Locate and Write, and Locate and Verify commands.

- Example 1. A read, write, or verify is done with Set Address to block 40 and length 256\*8. After successful completion the target address of the volume is block 48. The 8 blocks processed were blocks 40, 41, 42, 43, 44, 45, 46, and 47.
- Example 2. A read, write, or verify is done with Set Address to block 40 and length 256\*7 + 255 bytes. After successful completion, the target address of the volume is block 48. The read and write transferred 1 byte short of 8 blocks. The verify included all 8 blocks. You can read or write any number of bytes but the verify always verifies full blocks. If any byte of a block is read, written, or verified, the target address will point to the next block.
- Example 3. A read is done with Set Address to block 40 and length 256\*8 bytes. Block 42 is bad. An Unrecoverable Data error is set causing a QSTAT=1. At the end of this command, 8 blocks were transferred to the host. The P1-P6 parameter bytes in the status have block 42, the bad block. The target address is 48. If the host does a second Request Status command, the QSTAT will then be 0 and the P1-P6 bytes will contain the target address, 48.
- Example 4. A write is done with Set Address to block 40 and length 256\*8 bytes. Block 42 has a terrible header error. An Unrecoverable Data error is set causing a QSTAT=1. At the end of this command, 8 blocks worth of bytes were accepted from the host, blocks 40 and 41 were written, P1-P6 has block 42, and the target address is block 43. The SS/80 peripheral does not continue writing data after encountering a block it can't write. The first Request Status will show the Unrecoverable Data error and have P1-P6 point to block 42. A second Request Status will have no error bits set and P1-P6 will have the target address, 43.
- Example 5. A read is done with Set Address to block 3999. This volume has block numbers 0 through 3999. The length is 256\*3 bytes. This command will fail with the End of Volume bit being set, P1-P6 will have the target address of 0, and only 256 bytes will be transferred to the host. These are the 256 bytes of block 3999 and the last byte of these 256 will be tagged with EOI to terminate the transfer.

**Example 6.** A read is done with Set Address to block 3999. This volume has block numbers 0 through 3999. The length is set to -1, all 1's. This command will pass with no errors being set. The target address will be 0. The peripheral will have transferred 256 bytes to the host. These are the 256 bytes of block 3999 with the last byte tagged with EOI.

**Example 7.** If immediately after example 6 the host does a write with no Set Address complementary and a Set Length of 256 bytes, the command will complete with 256 bytes being written to block zero and the unit's target address being block 1.

**Example 8.** Suppose the host does a read with Set Address to block 100 and length of 2560 bytes. Suppose the peripheral is not functioning and is unable to seek to block 100. In this case the peripheral will return only one byte tagged with EOI, the unit fault bit will be set causing a QSTAT of 1. The target address could be anywhere. For hardware failures, there is no guarantee as to the target address.

The host should be careful about doing writes near the end of the address space without specifying an address, because the target address will wrap around and the writes may write over good data.

In general it is best for the host to always send the Set Unit, Set Volume, Set Address, and Set Length template with each read, write or verify opcode.

#### VARIATIONS FROM CS/80:

- CS/80 allows for three vector as well as single vector addressing. We have tried to simplify things by allowing only single vector addressing in SS/80.
- In the response to Request Status, an SS/80 device will always specify the address in bytes P1 through P6 in single vector mode.

#### STATUS BITS SET:

If the single vector address is outside of the range for the intended unit, the status error bit Address bounds will be set. If an Address bounds error occurs during a Set Address command, the target address will not be changed. The old value for the target address will be left as is. The target address will be set to zero any time an End of Volume error occurs. If you read up to the end of the volume of the unit, the unit will automatically set the address back to block 0.



## SET FORMAT OPTIONS

### FORMAT:

Set Format Options opcode = <00110001>

<00110001> is the opcode for an Initiate Utility command where the host sends an execution message.

Parameter bytes = < P1 > < P2 >

( 2 parameter bytes )

P1 = 11110011 = F3H  
P2 = 01011111 = 5FH

The HP-IB sequence is the standard format shown in "A Typical SS/80 Command," p. 3-6.

The execution message is a single byte, the option byte, tagged with EOI which the host sends to the peripheral.

### TYPE:

Utility

### TRANSACTION FLOW:

Command, Execution, Report.

### DESCRIPTION:

#### CAUTION

Sending Format Options requires knowledge of the specific product. Do not use these option values in a Parametric Driver.

This command is used to tell the peripheral to format in a special way. It is different from most other SS/80 commands in that the data passed will be different for different products. The command consists of the opcode and 2 parameter bytes. In the execution message, one byte tagged with EOI specifies to the selected unit which option should be performed during the following Initialize Media commands. The option remains selected until that unit receives any of the clear commands or the power is cycled to the peripheral.

For example, a user wanting to format a HP 9122 flexible disc drive with 512 byte sectors, should send this command to the HP 9122 with the option byte set to 2. For 1024 byte sectors, the option byte should be set to 3. The proper byte to be set can be determined from the user manual or documentation included with the disc drive. This command is product specific.

Because this command is product specific, it must be used with great caution. Certain option numbers may not be supported on future generations of peripherals. Therefore Host firmware that includes fixed options numbers may lead to serious problems in the future. A driver is not parametric if it must know that Format Option 2 means a format with 512 bytes on a HP 9122. One possibility is to ask the user to supply the options byte. The driver is only passing information from user to peripheral and does not have to know what the byte means.

This brings up the problem of how to know if the device has any options. If the host sends the Set Format Options command with the option byte of FFH, the SS/80 device will accept this and set no errors if the device has format options. If the SS/80 device does not have any selection of format options, then the device will set a Parameter Bounds error. All SS/80 devices will respond to this command. CS/80 devices do not understand this command. If the peripheral has only the default options and no others, it will respond to Set Format Options(options byte = FFH) by setting the Parameter Bounds error. If the peripheral

## SET FORMAT OPTIONS

(Continued)

does have some options, it will respond to the Set Format Options command (options byte = FFH) by setting no errors. Once the driver knows the device has some format options it can then ask the user to supply a format options byte. The driver then sends this byte to the peripheral. If the user supplies a number which is not a valid options number, the Parameter Bounds error is set. After the unit is cleared or after powering on, the unit will set his format option to the default value. A format options byte of 00H also specifies the default value. Default values, of course, may be different for different products.

Format options byte	Meaning
00H	Use product default. See product documentation.
01H - FEH	These may be valid format options. Their meaning is product specific.
FFH	This option byte is used to see if the SS/80 device has any format options.

The meaning of the options byte is defined by a product and the user must see that particular product's list of options to send the correct byte.

This command is very valuable in allowing a user to format a flexible drive in different ways. It is very dangerous from a command set perspective, for the options byte is meaningful only for certain products and not others. If the host puts the option bytes into firmware, that firmware may have to be changed on the next SS/80 peripheral. The division producing the peripheral will try to keep all options the same for similar products but this can not be guaranteed. We chose a parametric command set because we wanted to avoid changes to host code each time a new peripheral comes out. Because this is a product specific command, the host has to be very careful how to use it. If the host is not careful, we will all be back in the Amigo protocol situation of having to update host code for each new peripheral.

An example: HP 9122 3 1/2-Inch flexible disc drive

Format Option Byte	Option
0 (default)	256 byte blocks, double sided
1	256 byte blocks, double sided
2	512 byte blocks, double sided
3	1024 byte blocks, double sided
4	HP 9121 format (256 byte blocks single sided )

### VARIATIONS FROM CS/80:

- Not applicable. This command is not in CS/80.

**FORMAT:**

Set Length opcode = &lt;00011000&gt;

Parameter bytes = &lt; P1 &gt; &lt; P2 &gt; &lt; P3 &gt; &lt; P4 &gt;

( 4 parameter bytes )  
( MSB . . . . LSB )

Parameter format: 4-byte unsigned binary number ( Except all 1's means the entire volume )

**TYPE:**

Complementary

**TRANSACTION FLOW:**

Command, Report.

**DESCRIPTION:**

This command specifies the length of a data transfer in number of bytes.

The four bytes following the Set Length opcode contain the byte count of the transfer length. If this field is not included in the command message, the transfer length will be determined by the unit's power-on or target value.

Each unit keeps a "target" value of length. At power-on it is the power-on value of all 1's which means a full volume. Whenever the Set Length command is given, whether it is alone, with only other complementaries, or with other complementaries and a Real Time, General Purpose, or Diagnostic command, the Set Length command changes the unit's target value of length.

The peripheral device does not keep the host from attempting to read or write through the end of the media on a given volume. Since there is no bounds checking on the Set Length complementary parameters, the device does not realize that end of volume has been violated until the execution of a read, write, or verify command actually encounters it. All of the data specified by the host will be read or written right up to the end of the media, at which time the end of volume status bit is set, and remaining data is sinked or sourced. The peripheral device sources data by sending a single byte of value 1 tagged with EOI. When the end of volume error occurs, the target address is set to block 0.

A length specification of all 1's implies a transfer size equal to the selected volume. If the unit has 7700000 bytes, then the power-on length specification of -1, or 4 bytes of FFH, is 7700000 bytes. The power-on target address is block 0. However, if the length specification of -1 is given and the target address is not zero, the -1 means a length up to the end of the volume. The host may use the Length command to specify a full volume transfer. This transfer will begin at the target address and continue to the end of the media. Whenever the length is set to all 1's, the end of volume error is suppressed. This is the way CS/80 does it and we have to be compatible. This means that on a read, write, or verify when the length is all 1's and the end of volume is encountered, no error will be set. The problem with this is that data will be lost with no errors generated if the peripheral doesn't have sufficient capacity to hold the data. Whenever the host sets the length to all 1's, especially during a write command, it should make sure that the capacities and target addresses are such that data will not be lost, because the end of volume error will not be set.

A full volume write really doesn't make much sense and is a feature which shouldn't be used. When the peripheral detects the end of the volume, it will start sinking bytes, but the host doesn't know when to stop sending them. On reads, the peripheral will stop the host by sending a byte of value 1 tagged with EOI when it reaches the end of the volume.

**SET LENGTH**  
**(Continued)**

Thus, if the host wants to do a verify from the target address to the end of volume, it can set the length to -1 and give the verify command. If there was no error, the data from the target address to the end of the volume was good and the new target address is now block 0.

A length specification of 0 causes a Locate and Read or Locate and Write to become a seek and have no execution phase. No data is transferred in this case. If the host mistakenly has the length set to 0 and tries to do an execution message during a read or write, a Message Sequence error will be generated.

**VARIATIONS FROM CS/80:**

- CS/80 has different definitions of how long the Set Length change is in effect. See "Complementary Commands," p. 3-18.

## SET RELEASE [NO OP]

FORMAT: Set Release opcode = <00111011>

Parameter byte = <TZ000000>

T = 1 Suppress release time-out

Z = 1 Release automatically during idle time

TYPE: Complementary

TRANSACTION FLOW: Command, Report.

### DESCRIPTION:

SS/80 devices will never set any bits which request release. This will simplify both the host driver and peripheral firmware. When this command is received by an SS/80 device it is treated as a No Op. The command is accepted and ignored. No error status bits are set. No range checking is done on the parameter byte.

### VARIATIONS FROM CS/80:

- See the *CS/80 Instruction Set Programming Manual* to understand the meaning of this command in CS/80. In SS/80 it is treated as a No Op. It is included in the command set for compatibility reasons.



## **SET RETURN ADDRESSING MODE**

**FORMAT:** Set Return Addressing Mode opcode = <01001000>

Parameter byte = <00000TTT>

TTT = Addressing mode

000 = single vector

001 = three-vector

**TYPE:** Complementary command

**TRANSACTION FLOW:** Command, Report.

**DESCRIPTION:**

In SS/80, only single vector addressing is allowed. If you give this command and specify three-vector addressing, a Parameter Bounds error will be generated. The power-on value of the addressing mode is single vector. This command is included in SS/80 only for compatibility reasons and since the power-on addressing mode is already single vector, new drivers should not use this command.

**VARIATIONS FROM CS/80:**

- SS/80 supports only the single vector addressing mode.



**FORMAT:** Set RPS opcode = <00111001>

Parameter byte 1 = < Time 1 >

Parameter byte 2 = < Time 2 >

Time 1 = time-to-target in hundreds of microseconds

Time 2 = window size in hundreds of microseconds

**TYPE:** Complementary

**TRANSACTION FLOW:** Command, Report.

**DESCRIPTION:**

In SS/80 this command is treated as a No Op. SS/80 devices will accept the command and will set no error bits. However, the timing specified will not be met.

**VARIATIONS FROM CS/80:**

- See the *CS/80 Instruction Set Programming Manual* to understand the meaning of this command in CS/80. In SS/80 it is treated as a No Op. It is included in the command set for compatibility reasons.



## SET STATUS MASK

FORMAT:

Set Status Mask opcode = <00111110>

Parameter bytes < P1 > . . . < P8 >  
8 parameter bytes

Parameter format: Bit positions in parameter bytes correspond to error bit positions in the error reporting fields of the status report. "1" means to mask the error.

TYPE:

Complementary

TRANSACTION FLOW:

Command, Report.

DESCRIPTION:

This opcode is followed by 8 bytes containing the status bits to be masked. Each bit position corresponds to an error bit position in the Request Status message. A "1" in a given bit position will cause that error to be masked. All error conditions except fault errors may be masked. Refer to the Request Status command for bit positions.

If any non-maskable status bits are set, a Parameter Bounds error will result. The power-on value has no error conditions masked.

The masked bits will not be reported by either Request Status or QSTAT. If a status bit is not masked, it reports a hard error (QSTAT=1) when set. The only exception to this is the Power Fail status bit. This bit reports a power-on status (QSTAT=2) when set.

In actual operation, when a peripheral detects an error, it will call a module to set the appropriate status error bit and set the appropriate QSTAT. If the bit is masked, the status error bit and the QSTAT will not be changed. The peripheral's response to the error is the same except for not setting the error bit and QSTAT. New masks are not applied to bits that are already set, nor will old errors that had been masked become visible when the mask is removed. The mask is applied to the status bit only at the time when the peripheral goes to set the status bit in its RAM. The status bits set will reflect the mask setting at the time the error occurred.

Most host drivers mask the informational status bits so they won't see QSTAT=1 if the peripheral has to do a retry on some data. If the host does mask the informational status bits, it must remember that a Universal Clear, Channel Independent Clear, or Amigo Clear resets the status mask back to having no bits masked. Thus such a host must do a Set Status Mask command after every clear to re-mask the informational status bits.

VARIATIONS FROM CS/80:

- In SS/80 the Set Status Mask remains in effect until another Set Status Mask command changes the mask or one of the clears sets the mask back to the power-on default state of no bits being masked. In CS/80 the duration of a complementary depended on whether the complementary was given with a non-complementary command or not. See "Complementary Commands," p. 3-18.



**FORMAT:** Set Unit opcode = <0010YYYY>

YYYY = Unit number (1111=Device controller)

**TYPE:** Complementary

**TRANSACTION FLOW:** Command, Report.

**DESCRIPTION:**

This command is used to specify a specific unit within a mass storage device. For example, the device could consist of a flexible disc drive that is Unit 1 and a fixed disc drive that is Unit 0. Commands directed to Unit 1 have YYYY = 0001. If YYYY = 1111, the command will be directed to the device controller.

Since this is a complementary command, it may be given alone, with a group of other complementary commands, or as part of a Real Time, General Purpose, or Diagnostic command message. No matter how it is sent, this command changes the value of the device's "target" unit. Each device keeps a target unit in RAM. At power-on or after a Universal Clear, Amigo Clear, or Channel Independent Clear to Unit 15, the target unit is Unit 0. After that the target unit is always what the last Set Unit command specified.

If a command is given which does not include a Set Unit complementary, the command is for the target unit previously set.

Since the power-on default value of unit is 0, if the device has only 1 unit, the host never has to specify a unit number for a single unit device.

Note that the Set Unit opcode, if present, must be the first byte in the command message. If the opcode appears elsewhere in the message an Illegal Opcode error will be reported.

A unit has a full set of target parameters including a target volume, target address, target length, target status mask, full 20 bytes of status and a QSTAT value. If the host stops giving commands to Unit 0, gives some commands to Unit 1, and then returns to Unit 0, the target parameters will be unchanged from their previous values.

Normally in a device with flexible and fixed discs, the fixed disc is Unit 0 and the flexible disc is Unit 1. This is not always the case; some devices have switches that can change which unit is 0. This allows a host that can boot only from Unit 0 to boot up from either the flexible or the fixed disc drive.

**VARIATIONS FROM CS/80:** None

**STATUS BITS SET:**

An illegal unit number (one that does not reside on the addressed device) produces a Module Addressing status error. In this case the target unit will not be changed.



**FORMAT:** Set Volume opcode = <01000YYY>

      YYY = Volume number

**TYPE:** Complementary

**TRANSACTION FLOW:** Command, Report.

**DESCRIPTION:**

This command is used to specify a specific volume within a unit. Most SS/80 fixed discs can be divided into multiple volumes. This allows for different file systems on the same fixed disc, different interleaves on the different volumes, and smaller directories. A unit can be divided into as many as 8 volumes, numbered 0 through 7.

The Describe command tells the host how many volumes the unit is divided into.

Since this is a complementary command, it may be given alone, with a group of other complementary commands, or as part of a Real Time, General Purpose, or Diagnostic command message. No matter how it is sent, this command changes the value of the unit's target "volume". Each unit has to keep a target volume in RAM. At power-on the target volume is 0. After that it is the last volume specified for that unit in the Set Volume command or any clear command. Any of the clear commands set the target volume back to Volume 0.

If a command is given which does not include a Set Volume complementary, the command is for the target volume previously set.

If an illegal volume number is specified (not included on the particular device) a Module Addressing status error is generated. In this case the target volume is not changed.

The host should be very careful about using volumes and always must include the full Set Unit, Set Volume, Set Address, and Set Length with Locate and Read and Locate and Write commands. A volume is different from a unit in that all the volumes of a unit share the same target address, target length, status mask, status bits, and QSTAT. Suppose the host does a Set Address to block 10 of Volume 0, Unit 0, and then a Set Address to block 8 of Volume 1, Unit 0. If the host reselects Volume 0, Unit 0 and does a single block read, it will be reading block 8. A unit has a full set of target parameters including target volume, target address, target length, target status mask, full 20 bytes of status and a QSTAT value. Each unit has its own RAM locations to hold these target values. Volumes share the unit's targets. Volume 0 would use the same RAM location for the target address as would Volume 1 or any other volume.

Instead of doing a Set Volume alone, it is always better to do Set Unit, Set Volume, Set Address, Set Length and then the host will know the current address and length for the target volume.

Since multiple volumes in SS/80 are usually single fixed discs that are divided into volumes, there is only one head actuator and switching from doing reads on one volume to doing reads on another volume will cause a seek.

**VARIATIONS FROM CS/80:**

- Most CS/80 units are single volume units.



**FORMAT:** Spare Block opcode = <00000110>

Parameter byte = <0000000T> sparing mode byte

T = 0 Retain data on reformatted track

T = 1 Do not retain data on reformatted track

**TYPE:** General Purpose

**TRANSACTION FLOW:** Command, Report.

**DESCRIPTION:**

This command involves the host in sparing. Since this does not follow the philosophy of SS/80, this command is not recommended for use and should not be used in new SS/80 drivers. **New drivers should not use this command.**

Currently most SS/80 devices will respond to this command by setting the No Spares Available status error bit.

**VARIATIONS FROM CS/80:**

- SS/80 does not support this command.



# UNIVERSAL DEVICE CLEAR

## HP-IB SEQUENCE:

< P 0 0 1 0 1 0 0 >	ATN	Universal Clear command P is the parity bit.
.....		Device recognizes command
	DPPR	Device disables parallel poll response
.....		Device is doing the clear operations here.
.....	EPPR	When done, the device enables parallel poll response.
		An optional reporting phase may follow.

TYPE: HP-IB Universal command

TRANSACTION FLOW: HP-IB Universal, Report (optional)

## DESCRIPTION:

This command instructs all devices on the bus to do a clear. After receiving this command the device disables parallel poll, does the clear, and then enables parallel poll and goes into the reporting phase. A reporting message following a Clear command is optional and the device will accept and execute a command if sent.

After a Clear command, the host must wait for parallel poll before sending any other command. This will avoid channel timeouts if there is a delay in processing the Clear command.

SS/80 devices will respond to Clear and Cancel commands as soon as they can, but it won't be immediate because the number of interrupts available is limited. Some SS/80 devices will periodically check the bus for clears or cancels when doing the Initialize Media command. The frequency at which the device checks for a clear or Cancel is different for the different SS/80 peripherals.

The Universal Clear command does the following:

1. It aborts the current operation at the earliest opportunity such that no data corruption can take place.
2. The channel options set by the HP-IB Parity Checking command are not cleared during a clear.
3. All complementary parameters and targets will be reset to their power-on values:
  - A. The selected unit becomes Unit 0.
  - B. The selected volume becomes Volume 0.

## **UNIVERSAL DEVICE CLEAR**

(Continued)

- C. The target address becomes block 0.
  - D. The target length becomes -1.
  - E. The Status Mask is cleared so that no bits are masked.
  - F. Each unit which has Set Format options sets these back to the power-on values:
4. The status report bits are cleared. All status bits are cleared unless the Diagnostic Result bit is set. If the Diagnostic Result bit is set, then no status bits will be cleared. The only way to clear the status if the Diagnostic Result bit is set is to do a Request Status command. This ensures that the host will see that the peripheral failed a diagnostic which means something is wrong. The Power Fail bit is always cleared, so a clear will make QSTAT go from 2 to either 0 or 1.
  5. QSTAT is set to indicate whether or not status should be requested. QSTAT will be 1 after a Clear only if the Diagnostic Result bit is set.
  6. The Universal Clear is a "hard" clear so the device should perform a recalibration or restore to cylinder 0. This step is time consuming and may take many seconds.
  7. The peripheral device may want to reset its HP-IB chip. This is risky and not recommended. The Untalk and Unlisten after the Amigo Clear may hang if the chip is reset before it can complete the handshake. It may be good to reset the chip in case it is causing the problem; however, resetting the chip can cause problems itself. SS/80 peripherals are better off not resetting their HP-IB chips. The HP-IB address switches do not have to be updated during a clear.
  8. The peripheral device lastly enters an optional reporting state. It will accept any command from the host. It is recommended that the host always do a reporting message at the end of every command.

On an Initialize Media command, an SS/80 device will periodically check the bus for a clear or a Cancel command, which can be used to abort the initialization.

### **VARIATIONS FROM CS/80:**

- Some CS/80 devices like the HP 7908 will allow a clear to clear the Diagnostic Result bit if it has been returned to the host by some other unit of the device.

**FORMAT:**

Validate Key opcode = <00110001>

2 Parameter bytes = <P1> <P2>

<00110001> is the opcode for an Initiate Utility command where the host sends an execution message.

P1 = 11110001 = F1H  
P2 = 00000010 = 02H

The HP-IB sequence is the standard format shown in the section "A TYPICAL SS/80 COMMAND".

**TYPE:**

General Purpose

**TRANSACTION FLOW:**

Command, Execution, Report

**DESCRIPTION:**

A 12-byte key is sent from the host to the peripheral in the execution phase of this command.

There are 42 keys per volume. The key passed in the execution phase of the command is compared against all of the possible keys stored on the volume. If any of the keys compare, QSTAT will be 0. If none of the keys compare, or if no keys have been written, QSTAT will be 1 with a No Data Found error bit set. The keys are stored in a place that is unavailable to the user. When a volume is initialized, the key area of the volume is cleared of all keys.

This command enables a host to implement some form of security for certain software products. A software package can issue the Validate Key command, passing to the device the presumed key stored on the disc. The key stored on the disc will be examined and whether or not the No Data Found error bit is set will indicate if any key on the selected volume matches the key passed to the peripheral in this command. Presumably the applications software will abort the program if the keys don't match.

**VARIATIONS FROM CS/80:**

- This command is the device dependent Initiate Utility command in CS/80.



**See the command listing for Read Loopback.**

**The End**