SUBSET/80
FOR
SMALL HARD DISCS
AND FLEXIBLE DISCS

July 1, 1983

GENERAL INFORMATION ON SUBSET/80

SUBSET/80 is a subset of the CS/80 protocol which has been used at HP on their high capacity hard discs. SUBSET/80 will be used on the low end Floppies and Hard discs.

SUBSET/80 and CS/80 are essentially the same so it will be possible of the lowest cost desktop computer at HP to use the entire spectrum of HP mass storage products from the Micro Floppy up to the 400 Megabyte Winchesters.

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

Older command sets at HP required the host to have prior knowledge of the Floppy or Hard disc 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 SUBSET/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 Floppy or hard disc 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. There is no mention of heads, and cylinders. The host sees each device as a linear address space. The peripheral manufacturer can then upgrade his peripherals ,increase their storage capacity and performance, and they will function correctly without the host computers making any changes to its software. Older computers already in place can then buy the latest peripheral.

Another improvement of SUBSET/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 command set itself is bus independent. At HP it is being implemented on HP-IB and HP-IL.

Commands are split up into Complementaries and Real Time commands. The complementaries are things like Set Unit, Set Volume, Set Address, and Set Length.. These need only be set up once so that if you are always reading in 1K byte lengths, and want to read at the target address, the read command consists of only 1 byte. This minimize bus overhead. The target address is automatically incremented.

The error bits are all predefined so there is no confusion for the host as to what a certain status byte means.

Although the command set has a wealth of commands, only about 5 are needed by the host, so a very simple driver is possible.

Roger Faaborg

WHY SUBSET/80?

September 9, 1983
Roger Faaborg
GLD 1-356-2141

1. Why is the Floppy and small hard disc section of Greeley division switching from Amigo protocol to SUBSET/80?

With the Amigo protocol, each time a new peripheral came out, the host driver software had to be changed or the new product had to appear to be an old product. The 5 Megabyte hard disc in the 9134A was made to look like 4 9895A Floppies so it could work with existing drivers. This method works occasionally, but usually performance or capacity is sacrificed. To correctly support a new Amigo device would require a revision to the host software. Each Amigo device returns 2 bytes in the Identify command. Only if the host code had prior knowledge of this Identify code would it be able to function with this device.

With SUBSET/80 Floppies and hard discs, the host can write one parametric driver. It will work with all SUBSET/80 hard discs and Floppies and it will work for all future hard discs and Floppies. Greeley could then introduce new peripherals and they could be sold immediately to the existing base of computers. No driver software would have to be changed. The SUBSET/80 peripheral tells the host pertinent information in response to the DESCRIBE command.

2. Why did we choose SUBSET/80 rather than CS/80?

Over a period of about 6 months, driver writers and peripheral firmware writers from all interested divisions gave inputs as to what the low end mass storage protocol should be. It was decided to do CS/80 with some slight modifications. Call me if you want a copy of the final document.

SUBSET/80 is identical to CS/80 except for the following:

1. Initialization of the medium is done totally by the peripheral device in SUBSET/80. To initialize a SUBSET/80 Floppy or hard disc, the host need only give the Initialize Media command. When this command has completed, the host can write a directory and used the medium. In CS/80 the initialization routines involved the host software in doing verifying, reading error rate tables, and sparing. We decided this code is better placed in the peripheral firmware rather than in the host firmware. This is necessary if new peripherals are going to work with old host SUBSET/80 drivers.

2. To simplify the peripheral's firmware and hardware, the following CS/80 commands are treated as No ops by SUBSET/80:

CS/80	SUBSET/80
RELEASE	NO OP
RELEASE DENIED	NO OP
SET RPS	NO OP
SET RELEASE	NO OP

3. The CS/80 commands listed below are not supported in SUBSET/80. They result in an illegal opcode error if given to a SUBSET/80 device. The reason they are not supported is also given.

CS/80	REASON NOT SUPPORTED
Cold Load Read	Identical to Locate and Read and not used by low cost hosts.
Set Block Displacement	Not used by low cost hosts. Adds complexity to peripheral code.
Set Burst	Not used by low cost hosts. Adds complexity to peripheral code.
Set Retry Time	SUBSET/80 peripherals do a sufficient number of retries so that the host dc not have to do any.
Copy Data	Added considerable complexity to the low cost peripheral code. Can easily be done through read and write commands.
Selected Device Clear	Not possible with currently used low cost HP-IB chip. (8291A)

4. Now we would have a true subset, except in getting rid of the release commands we had to add two commands not in CS/80. These are DOOR LOCK and DOOR UNLOCK.

SUBSET/80 is so close to CS/80 that most CS/80 drivers will only have to change their Initialization procedure to support SUBSET/80. This initialization procedure involves only giving the Initialize Media command.

It is very important for host divisions to support SUBSET/80 as soon as an opportunity to change their mass storage driver occurs. The change to a CS/80 driver is simple and the modified driver will then be able to support all future Floppies and hard discs from Greeley division without any more changes.

Please call me if you have any questions.

Roger Faaborg
1-356-2141

SUBSET/80
FOR HARD DISCS AND FLOPPIES

FROM: Roger Faaborg, GLD 1-356-2141 July 1,1983
(Final version)

DISTRIBUTION:

Greg Brake, GLD
Doug Clifford, GLD
Scott Dehart, GLD
Mike Niquette, GLD
Bob Sobol, GLD
Mike Tremblay, GLD
Dave Joslin, GLD
Gordon Nuttall, GLD
Jeff Osborne, GLD
Lynn Formanek, GLD

Doug Voigt, DMD
Mark Anderson, DMD
Pat Walker, DMD
Steve King, DMD
Steve Lieske, DMD
Bob Pentecost, DMD
Kent Wilken, DMD

Joe Cowan, FSD
Ted Lappin, FSD
Marcel Meier, FSD
David Pinedo, FSD
Jeff Hirschl, FSD

Eiko Johnson, DSD
Chris Lehner, DSD
Dave Snow, DSD
Bill Ashcraft, DSD

Mike Mason, CSY (Roseville)

John Lee, POD
Greg Woods, POD
Willy Lassiter, POD

Jim Andreas, PCD

John Shada, (Colorado Springs)

TABLE OF CONTENTS

INTRODUCTION

HP has made a significant commitment to develop low cost personal computers and to become a significant force within that market place. This commitment has focused our efforts on lower cost designs that can be competitive in price and performance.

The implementation of CS/80 used on the 7908/Linus project is too complex to be used on low end systems. Too much firmware and software is required making development times long. What is needed is an implementation of CS/80 which uses only the necessary commands and restricts the number of features used. We are definitely not defining a new command set. We are only placing 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 will also decrease the time it takes to decode a command.

This cutdown version of CS/80 is called "SUBSET/80".

Before reading about SUBSET/80, everyone should first study the CS/80 INSTRUCTION SET programming manual, part[illegible]5955-3442 available from DMD. This document contains invaluable information which we have not tried to duplicate here.

Over the last few months there have been a number of formal and informal discussions of what SUBSET/80 should include and what it should leave out. The following are the basic objectives for SUBSET/80:

1. SUBSET/80 should preserve the basic functionality of CS/80. We want to leverage the enormous effort undertaken in devloping CS/80, and definitely do not want to define a new command set. The SUBSET/80 commands will be implemented in the same manner as CS/80 commands whenever possible.

2. SUBSET/80 like CS/80 should have as its goal device independent drivers, easily upgradeable for new peripherals. THIS IS VERY IMPORTANT. The Floppy in the Sparrow Hawk drive will probably be introduced with one capacity and shortly after be upgraded. We expect host drivers to function without modification for the higher capacity Floppy.

3. SUBSET/80 should spell out exactly what commands a SUBSET/80 peripheral must support.

4. SUBSET/80 will be tailored to the lower cost devices because of the higher sales volume and greater sensitivity to cost versus additional features.

5. Each low cost peripheral may support some commands outside of those defined by SUBSET/80. This support would last for at least one generation of peripherals to allow existing products to use new peripherals. These commands would be negotiated on a product by product basis. New drivers, however, must use only the SUBSET/80 commands specified in this document. Obviously, the driver does not need to use all the SUBSET/80 commands, but only those it needs.

Although CS/80 is a channel independent protocol, we will assume in this document that the channel is HP-IB.

The basic differences between the implementation of CS/80 proposed in this document and that implemented by 7908/Linus are:

1. Initialization of media is done totally by the device in SUBSET/80. The Linus/7908 implementation has a procedure the host must follow. See the Initialize Media command.

2. SUBSET/80 devices do not support the extensive diagnostic commands which were supported on the 7908/Linus products. The responsibility of diagnostics is left to service routines using the Download command. This is because SUBSET/80 drivers and devices want to use minimum ROM.

This is the final document on SUBSET/80 for hard discs and floppies. Products are being developed based on this document. It is now much more difficult to make changes.

The part of SUBSET/80 which deals with QIC tapes is still being worked on. It will be incorporated into this document at a later time.

Roger Faaborg
July 1,1983
telnet 1-356-2141

DEVICE ADDRESSING HIERARCHY

In SUBSET/80, as in CS/80, the hierarchy in descending order is: device, unit, and volume. A device on HP-IB is characterized by an HP-IB address, usually switch selectable. For example, a hard disc and floppy may be packaged in one box with one HP-IB port out the back and one set of 3 switches to set the address.

In this floppy and hard disc combination, each is a unit. The hard disc might be unit 0 and the floppy unit 1. There can be up to 7 units in a device.

A unit may be split up into 8 possible volumes. Currently, all CS/80 and SUBSET/80 devices have only volume 0. A possible example of using volumes might be if a combination fixed and removable hard disc is produced. The fixed disc might be unit 0 and volume 0. The removable disc could then be unit 0 and volume 1. One could as well call the fixed disc unit 0 and the removable disc unit 1.

TRANSACTION STRUCTURE

A transaction is a logically complete operation between a system host 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.

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 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. 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. 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. The QSTAT byte reflects the current status of the device.

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 floppy or tape has been loaded. Any incomplete transactions were aborted and should be repeated. The host should reconfigure any programmable operating parameters.

The sequence of command, execution, and report must be followed. The peripheral controls when the execution phase and report phase are entered by enabling parallel poll. 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.

In SUBSET/80 any background tasks are done in the command ready phase and can be interrupted out of if there is a command on the bus. Thus the background diagnostics, if they exist, are totally transparent to the host. SUBSET/80 devices never request release.

Commands which may not follow this transaction procedure are the Transparent Commands. The transaction model does not always apply and each particular command should be checked to see the interactions between the host and peripheral device.

POWER-ON STATE AND THE UNSEEN QSTAT = 2 HOLDOFF

When power is first applied, SUBSET/80 devices will initialize their HP-IB chips and will not initially respond to polls. Any bus activity directed to the device's HP-IB chips may be lost when it is reset and initialized. A selftest will then be done by all units and all parameters will be set to their power-on state. See the table below for power-on values. Each unit will set it's Power Fail status bit which causes 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.

The HP-IB chips on SUBSET/80 devices usually 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 a SUBSET/80 device while it is doing selftest will not be handshaked off 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 a SUBSET/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 SUBSET/80 device is doing selftest. The host can remove the secondary from the bus if he wants to 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, 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 him from doing certain commands. These are listed on the EXECUTION REQUIREMENTS table given on a following page.

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 loaded, 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 is returned in the first reporting message the host gives. Returning the QSTAT=2 byte to the host eliminates the holdoff but does not clear the QSTAT byte.

To clear the QSTAT byte you must give one of the clear commands or do a Request Status command. An Amigo Clear, Universal Clear or Channel Independent Clear to the controller (unit 15) will clear all the status bits (in all units) except the Diagnostic Result bit.

In SUBSET/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.

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 necessarily 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 which you intend to give commands to.

Thus the host gets the QSTAT=2 in the reporting phase message. It is allowable for the host to do what is called a "Stand alone QSTAT" where the host does a reporting phase message not associated with a transaction. This will also give the host the QSTAT = 2. However, the host must send a Request Status or do a Clear to actually 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. When the host sees the QSTAT = 2, he should clear the QSTAT by doing one of the Clears or Request Status. It is imperative that the host, upon seeing the QSTAT=2, do a Describe and reconfigure his system for the new medium.

There are two recommended procedures for a host to see what CS/80, Amigo, or SUBSET/80 devices are connected to the bus.

METHOD A

1. The host scans the bus by doing an Identify command to each of the 8 addresses and waits 25msec 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 msec 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. Next the host can do a Describe command if the device is a CS/80 or SUBSET/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 8 bus addresses.

2. The host then waits for parallel poll from the devices which are 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 SUBSET/80 or CS/80 devices.

Both methods require the host to 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 the section on timeouts.

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

The host can't just wait for a device to enable parallel poll at power-on because that device may have been on and have its parallel poll disabled. The host may have been turned off and then on while the peripheral device remained on.

POWER-ON VALUES

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

EXECUTION REQUIREMENTS

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

This first column lists all the commands done by a SUBSET/80 floppy or hard disc.

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

The third column gives the state the peripheral must be in before it will receive and execute the command. Except for some transparent commands, the peripheral must be in the command ready state (command phase) to execute a command. The host must follow the transaction sequence for SUBSET/80 peripherals to work.

The fourth column specifies what phase the peripheral will be in at the end of the command. Normally the peripheral starts at command phase, goes to to execution phase (if there is one), then to report phase, and back to command phase. Command phase is also called command ready phase. At the end of the clears there is an optional reporting phase so the peripheral will accept a reporting message or execute a command. Some transparent commands like HP-IB Parity checking have no affect on the peripheral's phase.

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). The transparent commands are independent of what unit is selected.

UNIT 15

The controller on the peripheral device is referred to as unit 15. For example, if the device consists of a floppy and hard disc, the hard disc might be unit 0, the floppy 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 a Copy Data. The copy will involve two units, so the command itself goes to the device controller.

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.

The EXECUTION REQUIREMENTS table on the preceeding page 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.

PARALLEL POLL

The parallel poll response is used to indicate that the device has finished a phase of the transaction and is ready for the next phase. Another way of saying this is that when the device enables parallel poll, he is asking for service from the host. The device must then be ready to accept the next phase of the transaction. Parallel poll 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 to actually disable parallel poll before the secondary is taken off the bus, thereby keeping a fast host from giving you data, seeing parallel poll enabled (not yet disabled), and assuming the device is done when it actually hasn't yet started.

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, the low cost HP-IB chips 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 HP-IB to have secondaries hanging around. If the host chooses to send secondaries without waiting for parallel poll he 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 the Locate and Read command description.

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 only set 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.)

TIMEOUTS

To have truly parametric drivers, the timeouts they use on devices must also be parametric or be fixed for all SUBSET/80 devices. When a 5 Megabyte hard disc is upgraded to a 10 Megabyte hard disc, no timeout problems should be encountered and no changes to the driver should be necessary. In order to simplify things, we have used the timeouts already specified in the Describe command and set acceptable values for all others.

SUBSET/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. It is best to do a Loopback command or some simple command before doing the long command to make sure the device is there. 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 a SUBSET/80 device will respond to a command phase secondary and take the parameter bytes in in a few milliseconds. However, to allow some protection for future devices whose characteristics are presently unknown, the timeouts are set high. When the timeout the host should use is specified as 5 seconds, the SUBSET/80 peripheral will actually complete in 4 seconds so there is some margin. The host timeout of 25 milliseconds will be done by the peripheral in 23 milliseonds to allow some margin.

SUBSET/80 TIMEOUTS

Command message
A SUBSET/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.

Command phase additional time.
The next time is from when the parameter bytes and opcodes are accepted until parallel poll is enabled. This varies with the command and is shown in a table which follows. Those commands which specify 5 seconds total mean that the command message above will be accepted, the device will do what it has to do and parallel poll will be enabled in 5 seconds worst case.

COMMAND PHASE ADDITIONAL TIME.

COMMAND	TIMEOUT TO USE	TYPICAL RESPONSE TIME
Any complementary given alone or a group of	5 sec. total	Milliseconds

complementaries.

Locate and Read	U15-U16, Access time parameter in Describe	
Locate and Write	U15-U16, Access time parameter in Describe	
Validate Key	U15-U16, Access time parameter in Describe	
Describe	U15-U16, Access time parameter in Describe	
Request Status	5 sec. total	Milliseconds
Release	5 sec. total	Milliseconds
Release Denied	5 sec. total	Milliseconds
Download	Not specified, depends on code run.	
Spare Block	Infinite	Depends on device.
Initialize Media	Infinite	Depends on device.
Initiate Diagnostic (0,1,0)	5 minutes	Depends on device.
Locate and Verify	Infinite	Depends on length and device.

Execution phase

A SUBSET/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. I used U13-U14 so all execution phase timeouts will be the same and to allow time for any cleanup tasks after a Locate and read.

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

Report phase

A SUBSET/80 peripheral will accept the reporting phase secondary, disable parallel poll and send the QSTAT byte within 25 milliseconds worst case.

The remaining commands are different and 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 SUBSET/80 peripheral will accept the secondary, disable parallel poll and accept the two bytes in 5 seconds worst case. Typical response time would be in milliseconds.

Read and Write Loopback

The SUBSET/80 peripheral will accept the secondary, disable parallel poll and accept any bytes sent or send the number specified within 5 seconds worst case. Typical response time would depend on the number of bytes sent.

Universal Clear, Channel Independent Clear, and Amigo Clear

The SUBSET/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 time to actually do the clear is long and depends on the device so a host driver should give an infinite timeout while waiting for the peripheral to enable parallel poll.

Identify

The SUBSET/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 SUBSET/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.

NOTE: During some long commands like Initialize Media which may take many minutes, the SUBSET/80 device may be checking for a Clear or Cancel command at least every 5 seconds. This will allow the user to abort a format operation if he decides to do so. Support for this feature is device dependent and if the device can not allow a format to be aborted, it will not allow you to Clear or Cancel out of it.

Power-up — A SUBSET/80 device will be ready to repond to commands from the host after power is turned on in 5 minutes worst case. Typical times depend on the device.

The timeouts used in SUBSET/80 are then 25 milliseconds, 5 seconds, 5 minutes, and infinite.

FORMATTING PROCEDURE FOR SUBSET/80 DEVICES

The recommended procedure for initializing a medium in SUBSET/80 is to give the Initialize Media command. In SUBSET/80 this command 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 = Ø, then the medium is ready for the host to use. The host can then write a directory or whatever, no more 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. Possible status errors returned from the Initialize Media command are:

No Spares Available	Medium had too many defective areas to pass initialization.
Media Wear	Medium initialized successfully but almost all the spares were used up.
Not Ready	There is no medium in the drive. (Floppy)
Unit fault	There was some hardware failure with the unit.

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.

Since the timeout on the Initialize command is infinite, the host should first do some command to make sure the peripheral is alive and the bus works. Then the Initialize Media command can be given. The peripheral's processor will timeout all hardware so there is little possibility of the bus becoming hung.

On a hard disc, the host could do a Loopback and an Initiate Diagnostic (0,1,0) command before the Initialize Media, but these are not required.

To summarize, to prepare an unformatted drive for use by the host, the host should give the following commands:

1.Any command to make sure the peripheral is alive.
2.Initialize Media

SPARING STRATEGY FOR SUBSET/80 FLOPPIES AND HARD DISCS

Sparing strategy is a question of data integrity. The user of a low cost system expects it to work without errors. We must guarantee that our products are as reliable or more reliable than the competition.

Since SUBSET/80 peripherals are aimed at the low cost host, the driver will probably be in ROM. Thus any complicated procedures will either be in the peripheral's ROM or host's ROM. Our general strategy is to place drive related items in the peripheral's ROM and leave system related operations for the host ROM.

Sparing strategy is one of those areas which could be handled by either the peripheral or the host. We feel it is more related to the peripheral hardware and should be taken care of by the peripheral whenever possible.

SUBSET/80 Floppies or hard discs which need to spare will then do autosparing. Except for the Auto Sparing Invoked bit being set, autosparing is completely transparent to the host.

Blocks can be autospared only if their data is recoverable. If an unrecoverable block is encountered during a Locate and Read or Locate and Write operation the host will have to be involved. Also, as a protection in case the device doesn't do autosparing, some SUBSET/80 devices may set the Marginal Data status bit. Both the Unrecoverable Data and Marginal Data status bits are flags to to the host that sparing should be done.

UNRECOVERABLE DATA. If the Unrecoverable Data bit is set during a Read or Write, the host could do retries. This is not necessary, however, since the peripheral will already have done a sufficiently large number of retries. THE HOST DOES NOT HAVE TO DO RETRIES ON DATA ERRORS. Upon seeing the Unrecoverable Data error bit set, the host should notify the user that the data was lost. The user should then be allowed two options:

A. The user could re-initialize the entire disc. All user data would be lost unless it has been backed-up. During initialization the bad areas will be spared. This is a possible solution for hard discs assuming it happens very infrequently. For Floppies, the disc should not be re-initialized since the defect may be caused by wear and more sectors are likely to follow. This is a problem because a Floppy and a removable cartridge disc are not distinguishable. They are both generic type 1. The better solution is given in B. below.

B. The user should somehow issue the Spare Block command, with the option to retain data on the reformatted track. The data in the bad block is lost, but this operation retains all other data and substitutes a new area of medium for the bad one. It is preferrable to re-initializing the entire disc because less data is lost, but it is still unfortunate. A device which doesn't spare will set the No Spares Available status error bit. If the other data on the track cannot be retained, an Unrecoverable Data error occurs again.

Getting an Unrecoverable Data error is very bad and should never be seen by most users.

MARGINAL DATA. Just to cover ourselves, some SUBSET/80 devices may set the Marginal Data error bit. This is a flag that sparing should be done by the host. In this case, the data is recoverable. If the Marginal Data bit is set, the driver should do the following:

1. Ask the user if sparing should be done.
2. If the answer is yes, the driver should read the data at the address shown in P1 through P6 and save it.
3. Next the driver should issue the Spare Block command retaining all data except the bad area and sparing it.
4. Finally the driver should re-write the data back into the new area of the disc. The single vector address is the same but the physical location will be different. The sector has been spared and no data was lost.

Hopefully, most devices will not set Marginal Data and will do only autosparing. A device which doesn't spare will return No Spares Available when the Spare Block command is issued.

Other status bits like Latency Induced and Recoverable Data DO NOT indicate that sparing is needed. The difficulty involved in reading the data was not great enough for sparing.

FLOPPY LOADING AND REMOVAL
(ONLY SUBSET/80 FLOPPIES)

This section is only for SUBSET/80 floppies. The SUBSET/80 QIC tapes may NOT be handled the same way. The QIC tape documentation will be included at a later time.

When a floppy disc is loaded into a SUBSET/80 unit, that unit will set the Power Fail status bit which causes a QSTAT of 2. This will happen when the floppy controller firsts notices that there is a new medium. The floppy controller will only look to see if there is a new disc or not 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 just does the Set Unit command and doesn't check for new media. However, if the host then gives a Locate and Read to the unit which has the new medium, the floppy 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 he must reconfigure for a new medium was loaded.

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

The following commands go out and check if the medium is there and will result in QSTAT = 2 if a new medium is detected.

Locate and Read Locate and Write Locate and Verify Spare Block Validate Key Initiate Diagnostic Describe Initialize Media	THESE CHECK FOR 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, Download, HP-IB Parity Checking, Loopback, Cancel, any clear, or identify. Any command can return a QSTAT of 2, but only certain commands result in checking to see if a new medium is present. This sets QSTAT=2 and it will remain set until a Clear or Request Status clears it.

If the host wants to see if the medium is there, he should give a Locate and Read of length 0 to the target address. This is a seek to the track you are 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.

For writes, the host should give a Locate and Write of length 0 to the target address. This is again a seek to the track you are 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 HOST SHOULD NOT USE THE DESCRIBE COMMAND TO SEE IF A MEDIUM IS THERE OR NOT. 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 his 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.

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.

Use the read or write specified above if you want to check that the medium is there before giving a command which accesses the floppy. The Describe command wasn't intended to become a "medium in" command.

When the host gets back a QSTAT=2, he 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 a QSTAT which 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.

MEDIA WEAR STATUS BIT

A SUBSET/80 device which has media which 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 it is important and should never 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 a dangerous 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 his medium.

SUBSET/80 COMMANDS

*************************** CORE SET *******************************

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
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
	LOOPBACK
	HP-IB PARITY CHECKING
DIAGNOSTIC:	INITIATE DIAGNOSTIC
	REQUEST STATUS

********************** DEPENDENT ON DEVICE TYPE ************************

DEPENDENT ON DEVICE TYPE:	VALIDATE KEY (FLOPPY AND HARD DISC)
	DOWNLOAD (FLOPPY AND HARD DISC)
	COPY DATA (TAPE AND HARD DISC)

****

Each peripheral must respond to all the commands in the core set and those device dependent commands which apply to it. For example, a Floppy must respond to all the commands in the core set and also VALIDATE KEY and DOWNLOAD. It needs to respond to no other commands but these.

On the other hand, the host driver for a Floppy can use only the core commands or VALIDATE KEY or DOWNLOAD. He can't expect the Floppy to respond to any other commands.

Certain commands like Set RPS are accepted by the SUBSET/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 SUBSET/80 devices should not use these 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 SUBSET/80.

Just as constraints are placed on which commands can be used, SUBSET/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 SUBSET/80 devices.

Any complementaries given should be included only once per transaction. The complementaries should be included in the following order:

Set Unit
Set Volume
Set Address
Set Length
(Next comes a Locate and Read, Locate and Write or
Locate and Verify)

No ops can be placed between any of these complementaries. Any of the complementaries can be left out, but those which remain should be in this order. This is the order for fastest decoding. Another order will work, but it will be decoded more slowly. The complementaries other than those listed above can be placed in any order.

COMMAND DESCRIPTIONS

Each command is described individually in the following pages. The commands are listed in alphabetical order. The outline used for each command is explained below:

```
***********************************************
****************  COMMAND NAME  ****************
***********************************************
```

FORMAT: This gives the opcode and parameters of the command. For a detailed byte by byte look at the HP-IB bus interactions, see Table 4-6 in the CS/80 INSTRUCTION SET programming manual, part number 5955-3442. For a typical transaction, see the section which follows on TYPICAL MESSAGE STRUCTURE.

HPIB SEQUENCE: For certain commands I included the complete sequence of events over the HP-IB bus for a command with good syntax and normal completion. Bytes from the host to the peripheral are shown as < X X X X X X X X >. Bytes from the peripheral to the host are shown as - X X X X X X X X -. "P" refers to the parity bit on HP-IB. "A D D R S" is the devices HP-IB address in binary. "ATN" is the attention line. "EOI" is the termination line. An "X" is a "don't care" value of a bit.

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

SUPPORT: This section states whether the command is supported by Floppies, Hard discs, Tapes, or some combination.

TRANSACTION FLOW: This specifies whether the command has Command, Execution, or Report phases.

DESCRIPTION: The description is a paragraph or two explaining how the command works.

VARIATIONS FROM CS/80: This section spells out any differences between the SUBSET/80 command and the same command in CS/80.

TYPICAL HP-IB COMMAND MESSAGE STRUCTURE

Many SUBSET/80 commands follow the HP-IB command message structure shown below. Any command which doesn't follow this structure will have its HP-IB SEQUENCE included with the description of the command. I chose the Locate and Read command as an example. On some commands, the execution message is not present. All real time, diagnostic, and general purpose SUBSET/80 commands follow this message structure.

********** COMMAND MESSAGE **********

< P 0 1 A D D R S >	ATN	Primary listen from host
< P 1 1 0 0 1 0 1 >	ATN	Command message secondary from the host.
	DPPR	Device disables parallel poll response.
< . . . . > < . . . . > < . . . . > < . . . . > < . . . . > etc.		Host sends 0 to N complementary commands.
< 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.
< P 0 1 1 1 1 1 1 >	ATN	Host unlistens 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.

. . .
. . .
. . .
. . .
. . .
		Peripheral device executes any complementaries, checks to see if medium is ready and seeks to the correct track.

	EPPR	Device enables parallel poll response which ends the command message. By enabling parallel poll, the device is asking for the execution

```
                                        message.

       ********** EXECUTION MESSAGE **********

< P 1 0 A D D R S >     ATN       Primary talk from the host.

< P 1 1 0 1 1 1 0 >     ATN       Execution message secondary
                                  from the host.

                        DPPR      Peripheral device disables
                                  parallel poll response.

                                  Peripheral device reads
       . . .                      first sector.
       . . .
       . . .

- X X X X X X X X -               Peripheral sends data to
- X X X X X X X X -               the host.
       . . .
- X X X X X X X X -

                                  If more data is needed,
       . . .                      peripheral device reads
       . . .                      another sector.
       . . .

- X X X X X X X X -               Peripheral sends data to
- X X X X X X X X -               host, tagging last byte
       . . .                      sent with EOI. 1 to n bytes
- X X X X X X X X -     EOI       can be sent to host.

                        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.

       ********** REPORTING MESSAGE **********

< P 1 0 A D D R S >     ATN       Primary talk from host.

< P 1 1 1 0 0 0 0 >     ATN       Reporting message secondary
                                  from the host.

                        DPPR      Peripheral device disables
                                  parallel poll response.

-    Q S T A T    -     EOI       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.

```
********************************************
***************  AMIGO CLEAR  ***************
********************************************
```

HPIB SEQUENCE:

< P 0 1 A D D R S >	ATN	Primary listen from host
< P 1 1 1 0 0 0 0 >	ATN	Amigo Clear secondary from the host
	DPPR	Device disables parallel poll reponse

< 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 1 >	ATN	Unlisten from host.(The host could send this unlisten any time after the SDC primary. It doesn't have to occur here.

TYPE: Transparent

SUPPORT: Core command, all peripherals must support.

DESCRIPTION:

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

See the Universal Clear command for details as to what is done during a clear.

SUBSET/80 devices will do parity checking on commands only if their hardware allows them to. If it does not, the A bit shown above will be ignored.

SUBSET/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 SUBSET/80 devices.

On any long operations like format, SUBSET/80 devices may periodically check the bus for a clear or cancel command.

VARIATIONS FROM CS/80:

The HP-IB parity check bit in the control byte is treated as a No op if the SUBSET/80 device's HP-IB chip can't do parity checking.

****
*************** CANCEL ***************
****

HP-IB SEQUENCE:

<P01ADDRS>	ATN	Primary listen from host.
<P1110010>	ATN	Secondary for a transparent command.
	DPPR	Peripheral disables parallel poll response.
[0010YYYY]		Optional Set Unit command from the host. YYYY = unit number. "[]" means optional.
<00001001>	EOI	Opcode for a Cancel command from the host.
<P0111111>	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.

********** REPORTING MESSAGE **********

<P10ADDRS>	ATN	Primary talk from the host.
<P1110000>	ATN	Reporting phase message secondary from the host.
	DPPR	Peripheral disables parallel poll.
- QSTAT -	EOI	Peripheral sends the QSTAT byte to the host.
<P1011111>	ATN	Host untalks the peripheral. Can occur anytime after QSTAT is accepted by host.

TYPE: Transparent

SUPPORT: Core command, all peripherals must support.

TRANSACTION FLOW: Command, Report.

DESCRIPTION:

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

An example of when this command is used might be useful. Suppose that the user decides to format a hard disc. This process may take 30 minutes. Just after he has started the format procedure, he decides he didn't want to do that just now and wants the peripheral device to stop. Somehow he 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 go to report phase. 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 recommended way to terminate a transaction is to terminate the message link, then send the Cancel command.

Cancel commands are polled for during long operations, 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 at least every 5 seconds.

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 are Message Sequence and Message Length.

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.

There is the possibility that some future device may not allow you to cancel out of a format. Because of this, the ability to Cancel a long command is a feature which most SUBSET/80 devices will have, but is not a definite requirement.

VARIATIONS FROM CS/80: None

****
*************** CHANNEL INDEPENDENT CLEAR ***************
****

HP-IB SEQUENCE:

<P01ADDRS>	ATN	Primary listen from host.
<P1110010>	ATN	Secondary for a transparent command.
	DPPR	Peripheral disables parallel poll response.
[0010YYYY]		Optional Set Unit command from the host. YYYY = unit number. "[]" means optional. YYYY = 1111 is device controller.
<00001000>	EOI	Opcode for a Channel Independent Clear command from the host.
<P0111111>	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.

********** REPORTING MESSAGE **********

<P10ADDRS>	ATN	Primary talk from the host.
<P1110000>	ATN	Reporting phase message secondary from the host.
	DPPR	Peripheral disables parallel poll response.
- QSTAT -	EOI	Peripheral sends the QSTAT byte to the host.
<P1011111>	ATN	Host untalks the peripheral. This can occur anytime after QSTAT is accepted by the host.

TYPE: Transparent

SUPPORT: Core command, all peripherals must support.

TRANSACTION FLOW: Command, Report (report is optional)

DESCRIPTION:

This is the recommended clear command in SUBSET/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 steps taken in doing the clear are identical to those undertaken in a Universal Device Clear. See that command for the details.

After the clear operation, the unit specified in the command will remain the selected unit. However, if the unit specified is the controller (unit 15), unit 0 will be the selected unit.

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.

On any long operations like a format, a SUBSET/80 device may periodically check the bus for a clear or cancel command.

VARIATIONS FROM CS/80: None

****
*************** DESCRIBE ***************
****

FORMAT: <00110101>

TYPE: General Purpose

SUPPORT: Core command, all peripherals must support.

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 2 units, each with a single volume, and the Describe command is given to unit 15, the order of the bytes returned is:

controller field
unit 0 field
volume 0 field of unit 0
unit 1 field
volume 0 field of unit 1

If the device has 2 units, each with a single volume, and the Describe command is given to unit 0, the order of the bytes returned is:

controller field
unit 0 field
volume 0 field of unit 0

NOTE: The Describe command will always return the correct number of bytes. However, if the QSTAT returned in the reporting message of a Describe command is 2, then a new medium has been detected and the host should reconfigure for this new medium. The host should clear the QSTAT, reconfigure all operating parameters and give the Describe command again. This time it should return a QSTAT of 0. 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 SUBSET/80 is 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 has timeouts similar to a Locate and Read command. See the section on TIMEOUTS.

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

C1,C2 = Installed unit byte; 1 bit for each unit. (unit 0 = LSB. e.g. if there are only 2 units, unit 0 and unit 1, then C1 = 00000000 and C2 = 00000011 in binary.

C3,C4 = Maximum instantaneous transfer rate in thousands of bytes per second.

C5 = Controller Type

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

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

U1 = Generic Device Type

0 = Fixed disc
1 = Removable disc or combination. (includes floppy)
2 = Tape (Random access format like Linus or Buffalo)
4 = Tape (serial tape, QIC format)

U2-U4 = Device number. Represents actual HP product number: XX XX XY (BCD Coded, 2 digits per byte). XXXXX = product number. Y = option.

U5-U6 = Number of bytes per block.

U7 = Number of blocks which can be buffered.

U8 = 0 Recommended burst size. SUBSET/80 devices will set U8 = 0. SUBSET/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. The peripheral should compute this transfer rate using an effect interleave factor of 1, even if that effective interleave factor may not be possible with the given media/controller. See the discussion on V13 below.

U13-U14 = Optimal retry time in 10's of milliseconds.

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 section on Timeouts.

U17 = Maximum Interleave factor. If the controller or media do not support interleave factors, then U17 should be returned as 0.

U18 = Fixed volume byte; one bit per volume (set if fixed); Volume 0 = LSB.

U19 = Removable volume byte; one bit per volume (set if removable); Volume 0 = LSB.

********** 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.

SUBSET/80 devices use only single vector addressing. The maximum values for three vector addressing will be returned in V1 through V6 but the host should never do a Set Address or Set Return Addressing Mode in any mode except single vector.

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.

This byte is defined in terms of the need to calculate timeouts for Locate and Read and Locate and Write requests. The following rules should apply to the designers of discs:

1, The field V13 should return the media interleave factor if known.

2, If the media interleave factor is not known, then 0 or the maximum effective interleave factor should be returned.

3, If the disc drive (media/controller) can not maintain a transfer at media interleave factor 1, then the effective interleave factor should be returned (0 or the maximum effective interleave if the actual effective interleave is not known by the controller).
For example, if the drive can only support media interleave factor 4 and the media interleave factor is 2, then the effective interleave factor should be <the maximum media interleave factor> + <4 - 2>. Similarly if the media interleave factor is greater than 4, then the media interleave factor should be returned as the effective interleave factor.

V13 = the effective interleave factor. "Media interleave factor" is defined as the skewing factor used on the media (the classical interleave factor). "Effective interleave factor" is defined as the value that should divide the continuous average transfer rate U11-U12 to give the actual transfer rate of the media/controller.

The intent of the V13 and U11-U12 fields of Describe are to allow the driver to calculate the time required to transfer N bytes of data after the worst case access time (U15-U16) and the retry time (U13-U14) factors are accounted for. The rules above allow for the margin of error to prevent a pre-mature timeout from occurring.

VARIATIONS FROM CS/80:

1. In the controller description field, SUBSET/80 devices are denoted by having Controller Type 4,5, or 6. This means bit 2 is set for SUBSET/80 devices. This is the way you tell SUBSET/80 devices from CS/80 devices.

2. CS/80 specifies that the access time parameter in U15-U16 applies for read and write commands only. In SUBSET/80 this access time parameter is used in other commands which access the disc. See the section on TIMEOUTS.

```
************************************************************
***************        DOOR    LOCK        ***************
************************************************************
```

FORMAT: <01001101>

TYPE: General Purpose

SUPPORT: Core command, all peripherals must support.

TRANSACTION FLOW: Command, Report

DESCRIPTION:

In SUBSET/80 this command instructions the target unit of the device to lock its door. If the device does not have a door lock, the command is a No op.

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 SUBSET/80 devices. You can tell a device is a SUBSET/80 device by doing a Describe command and checking that bit 2 of byte C5 is set. See the Describe command.

```
*******************************************************
***************    DOOR  UNLOCK         ***************
*******************************************************
```

FORMAT: <01001100>

TYPE: General Purpose

SUPPORT: Core command, all peripherals must support.

TRANSACTION FLOW: Command, Report

DESCRIPTION:

In SUBSET/80 this command instructs the target unit of the device to unlock its door. If the device does not have a door lock, the command is a No op.

This command can be given to unit 0,1,2,..., but not to unit 15. If this command is given to unit 15 (the controller), an illegal Opcode error will result. The controller doesn't have any doors.

VARIATIONS FROM CS/80:

This command does not exist in CS/80. This command should only be given to SUBSET/80 devices. You can tell a SUBSET/80 device because bit 2 in C5 of the data returned in the Describe command will be set. See the Describe command.

```
******************************************
***************  DOWNLOAD  ***************
******************************************
```

HPIB SEQUENCE:

< P 0 1 A D D R S >	ATN	Primary listen from host
< 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 product number. AB CD EY is BCD coded, 2 digits per byte. ABCDE = product number. Y = option.
< F F F F F F F F >	EOI	The Download revision number in unsigned binary. 6 parameter bytes followed the opcode.
. . .		Device decodes command
	EPPR	Device enables parallel poll response when ready for execution message.
< P 0 1 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 0 1 A D D R S >	ATN	Primary listen from host
< P 1 1 0 1 1 1 0 >	ATN	Execution phase secondary
	DPPR	Device disables parallel poll response
< from 1 to a buffers worth of data >		Device received up to a buffers worth of data from

< last byte >	EOI	the host. 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.

TYPE: Diagnostic

SUPPORT: All Floppies and Hard discs must support this command.

TRANSACTION FLOW: Command, Execution, and then whatever.

DESCRIPTION:

Host drivers should not use this command.

The Download command is a command supported by only certain device types. 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 buffer. 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 and execution from then on is dependent on this downloaded code.

SUBSET/80 devices use the parameter field bytes P7 through P10 to put diagnostic information.

The purpose of the Download command is to allow service and test engineers complete freedom to write any routines 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 will set, and the report phase will be entered.

The Download command can be given to any unit including unit 15.

VARIATIONS FROM CS/80: This is the CS/80 Initiate Utility command.

```
*********************************************************
****************  HP-IB PARITY CHECKING  ****************
*********************************************************
```

HP-IB SEQUENCE:

< P 0 1 A D D R S >	ATN	Primary listen from host
< 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 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

SUPPORT: Core command, all peripherals must support.

TRANSACTION FLOW: Not applicable

DESCRIPTION:

SUBSET/80 devices will do parity checking on commands only if their hardware allows them to. If it does not, the V bit will be ignored.

If the S bit is set, the Service Request (SRQ) line will be asserted whenever the device enables parallel poll.

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

VARIATIONS FROM CS/80: None

```
********************************************
***************  IDENTIFY  ***************
********************************************
```

HPIB SEQUENCE:

< P 1 0 1 1 1 1 1 >	ATN	Untalk
< 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 1 1 1 1 0 >	ATN	Talk controller. The host sends MTA (My talk address) which automatically untalks everyone but himself.

TYPE: Unique

SUPPORT: Core command, all peripherals must support.

TRANSACTION FLOW: Not applicable.

DESCRIPTION:

Identify is a special-case HP-IB 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 SUBSET/80 devices return the value of 2 (00000010) in ID byte 1, and the product type code in ID byte 2.

Normally in a command which has the peripheral device talk, the host would end 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 may not unaddress SUBSET/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 himself to talk. Since there can be only one talker on HP-IB, all other devices are untalked.

VARIATIONS FROM CS/80: None

```
**************************************************
***************  INITIALIZE MEDIA  ***************
**************************************************

***************************************************
*                  CAUTION !!!                    *
*       Execution of the Initialize Media command *
* will destroy all user data on the selected unit.*
* Before executing the Initialize Media command,  *
* make certain that the proper unit has been      *
* selected. Failure to do so may result in the    *
* loss of needed data.                            *
***************************************************
```

FORMAT: <00110111> <00000YYY> <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 should 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

SUPPORT: Core command, all peripherals must support.

TRANSACTION FLOW: Command, Report.

DESCRIPTION:

SUBSET/80 devices will accept the initialize options byte and do what is appropriate for that device. Host drivers should only use the "000" option so no factory or field spares are lost. The loss of factory spares could render a device unusable. The "001" and "010" options should be used only by service engineers or others who understand what they are doing.

In SUBSET/80, this command does the complete initialization process which prepares the medium for use by the host. SUBSET/80 devices do multiple passes of writing, verifying and sparing and return only when all sparing has been done. The host can then issue this command and if the QSTAT returned is 0, he knows the medium is ready for use. The host does not have to 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 the section on TIMEOUTS and the section on FORMATTING PROCEDURE.

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 better understands the sensitivities and failure modes of the device than does the host driver writer.

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 his buffer memory correctly. This makes the device less sensitive to the interleave factor. How this is done depends on the device and how sophisticated his low cost hardware is. The ability to transfer data to the host over HP-IB while at the same time the device is filling his 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.

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 maximum interleave.

VARIATIONS FROM CS/80:

1. SUBSET/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 was not the case for the 7908/Linus which had an involved Initialization procedure. See the section on FORMATTING PROCEDURE for the recommended procedure for initializing a SUBSET/80 device.

****

*************** INITIATE DIAGNOSTIC ***************

****

FORMAT: <00110011> <00000000> <00000001> <00000000>
three parameter bytes, 0,1,0.

TYPE: Diagnostic

SUPPORT: Core command, all peripherals will support.

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 SUBSET/80 this command directs the device or unit to perform a complete diagnostic selftest. If the medium is present, the device will do a test which involves reads and writes on the medium. Writes will only be done 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 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 was 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 information for test and service telling which component failed. The codes are device specific and are not the same as those used on 7908/Linus.

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 was not used in CS/80 so this is a departure. We felt it was important for the host to at least know which device failed, but we still don't want the host driver to get involved in specific diagnostic information.

The bytes P7 through P10 will contain addition diagnostic information which should be ignored by the host driver.

See the section on TIMEOUTS 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 selftest performed is a subset of that performed at power-on. Tests on the processor, ROM, RAM, and HP-IB chip may only be done at power-on. 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 probably easier to give the Initiate Diagnostic (0,1,0) to the unit you are interested in. You will see if 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 SUBSET/80 and we hope it is clear enough so that host drivers will use it.

The idea of this command in SUBSET/80 is to do a "go" or "no go" check of the device.

The Initiate Diagnostic command does no clearing of complementaries or status.

VARIATIONS FROM CS/80:

1. With the 7908,11,12 this command can be given only to unit 15. Nickel/Buffalo allows this command to a specific unit. SUBSET/80 decided to allow it to any unit. In fact it is simpler to use if given to a specific unit other than unit 15.

2. In SUBSET/80 the Initiate Diagnostic (0,1,0) is the basic selftest. Most device will support no other selfest.

3. In SUBSET/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 SUBSET/80 driver using a CS/80 device like the 7908 might display that unit [P6] failed when P6 did not contain this information. CS/80 devices on the 7908 ,11,12 have P5 and P6 equal to 0. The driver could display this only with a SUBSET/80 device if he expects to run with SUBSET/80 and CS/80 devices. You tell whether the device is SUBSET/80 by the byte C5 in the Describe command.

4. In CS/80 for the 7908,11,12, the Clear commands would clear the Diagnostic result bit if it had been seen once by the host. This method was not used on Nickel/Buffalo because they allowed the command to go to units other than unit 15. In SUBSET/80 THE CLEAR COMMANDS NEVER CLEAR THE DIAGNOSTIC RESULT BIT. It must be cleared by doing a Request Status command.

```
************************************************
****************  LOCATE AND READ  ****************
************************************************
```

FORMAT: <00000000>

TYPE: Real time command

SUPPORT: Core command, All peripherals must support.

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 reenables parallel poll telling the host that it is ready to enter the execution phase.

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

If a problem is found by the device, like an illegal opcode, it will go to the report phase, 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 reenabled at this point since the transaction is done.

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 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. SUBSET/80 devices will set the Unrecoverable Data status error bit if there is a data error. If another block if found to be bad, the Unrecoverable Data Overflow bit is set. Regardless of how many data errors, the device will try to read all blocks and send back its best 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.

The host can always issue the command again if it wants to do retries. THE HOST DOES NOT HAVE TO DO RETRIES AS THESE ARE ALREADY 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 only read 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.

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

Set Unit
Set Volume
Set Address
Set Length

No op's 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.

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.

VARIATIONS FROM CS/80:

1. SUBSET/80 devices do not use RPS (Rotational position sensing). They accept this command as a No op.

2. SUBSET/80 devices do not use burst mode.

3. SUBSET/80 devices do not allow the host to set the retry time. It is a fixed time set by each unit.

```
*****************************************************
***************  LOCATE AND VERIFY  ***************
*****************************************************
```

FORMAT: <00000100>

TYPE: General Purpose

SUPPORT: Core command, all peripherals must support.

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 unit 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 retires. In any case, the idea is to let the host know if the previous write did not take so he 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.

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 the sections on SPARING STRATEGY and FORMATTING PROCEDURE.

Since in SUBSET/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 if 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 if the data is recoverable without retries or error correction and it is fast since no data is transferred over HP-IB.

VARIATIONS FROM CS/80:

SUBSET/80 devices do not keep error logs.

****
*************** LOCATE AND WRITE ***************
****

FORMAT: <00000010>

TYPE: Real time command

SUPPORT: Core command, All peripherals must support.

TRANSACTION FLOW: Normally Command, Execution, Report. For a 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 media is checked at this time to see if 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 Address, and Set Length for explanations as to how these work.

Assuming there was 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 if 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 should be filled by duplicating the last data byte, or by filling it 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 can not 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 him. 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 bad block. The address of the bad block will be in parameter bytes P1-P6. Otherwise, the target address always points to the next block to be done, which in this case is the one after the bad block.

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 write protected, without actually writing on the medium, he should do a Locate and Write of length 0 to the target address. This is a seek to where the device already is. It doesn't move anything but it will check to see that the medium present and it will fail with a Write Protect error bit set (QSTAT=1) if the medium is write protected. See the section on FLOPPY LOADING AND REMOVAL.

VARIATIONS FROM CS/80:

1. SUBSET/80 devices do not use burst mode.

```
******************************************
*************** LOOPBACK ***************
******************************************
```

HP-IB SEQUENCE: (READ LOOPBACK)

< P 0 1 A D D R S >	ATN	Primary listen from host.
< 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 > < P 2 > < P 3 > < P 4 >		4 byte length parameter (unsigned binary) Length in bytes to use in Loopback test.
< P 0 1 1 1 1 1 1 >	ATN	Unlisten from the host.

< 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.
-< X X X X X X X X >- -< X X X X X X X X >- . . . . -< X X X X X X X X >- -< X X X X X X X X >-		Read Loopback Data goes from the peripheral to the host. Write Loopback Data goes from the host to the peripheral. The number of bytes sent is specified in the length parameter shown above.
-< X X X X X X X X >-	EOI	Last byte is tagged with EOI.
< P J K 1 1 1 1 1 >	ATN	Host unlistens (JK=01) peripheral in a Write Loopback, or Host untalks (JK=10) peripheral in a Read Loopback.

```
********** REPORTING MESSAGE **********
```

< P 1 0 A D D R S >	ATN	Primary talk from the host.
< 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 1 >	ATN	Host untalks the peripheral.

TYPE: Transparent

SUPPORT: Core command, all peripherals must support.

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,. . . 0FFH,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.

If the loopback was successful, the device never asserts parallel poll. If something went 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 screws up 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.

If any of the bytes in a write loopback were wrong, a Channel Parity error is reported.

If the length of the loopback message was wrong, a Message Length error is reported.

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

The length of the loopback can be any 4 byte unsigned binary number. SUBSET/80 devices will be able to handle any loopback length specified.

VARIATIONS FROM CS/80: None

**************** NO OP ****************

FORMAT: <00110100>

TYPE: Complementary

SUPPORT: Core command, all peripherals must support.

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 ops can be placed at the beginning of a string of complementary commands,inbetween complementary commands, and between the last complementary command and a Real Time, General Purpose, or Diagnostic Command. However, No ops should not be placed after the last command or an Illegal Parameter error will be set.

VARIATIONS FROM CS/80: None

```
*************************************************
***************  REQUEST STATUS  ***************
*************************************************
```

FORMAT: <00001101>

TYPE: DIAGNOSTIC

SUPPORT: Core command, all peripherals must support.

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 the Section on FLOPPY LOADING AND REMOVAL.

In SUBSET/80, the Request Status command returns a 20-byte status report indicating the cumulative status of all transactions for a unit which have occurred since the status report was last cleared. The status report can only be cleared 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 ten bytes of additional information in the parameter field. The table on the next page shows the complete format of the status report for SUBSET/80 devices.

The identification field has 2 bytes. The MSB nibble of the first byte is the volume field, the LSB nibble of the second byte is the unit field, and the second entire byte is always set to all 1's by SUBSET/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 SUBSET/80. See the status description which follows for more information on what P1 through P10 can contain.

Reject errors include status bits which 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.

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, then QSTAT=1. If no status bits are set, then QSTAT=0.

VARIATIONS FROM CS/80:

1. In SUBSET/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.

2. The following fault error bits will NEVER be set by a SUBSET/80 device:

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

3. 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.

IDENTIFICATION FIELD <VVVVUUUU> <11111111>

VVVV = Volume number
UUUU = Unit number

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.

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 reject or fault errors have occurred prior to sequence error.)

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.

FAULT ERROR 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.

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.

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 Floppy 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 SUBSET/80 for future devices which may use a low cost version of the ABI chip which has a FIFO. Current SUBSET/80 devices use the 8291A and don't set this bit.

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 loaded.

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.

35 = Not Ready

The device is not ready, probably because the medium has not been loaded.

36 = Write Protect

The selected volume is write protected and the current command failed because of that.

37 = No Data Found

For Tapes this means a block accessed during a read has not been written. For hard discs and floppies, 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.

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.

INFORMATION FIELD ERRORS < 0 0 0 51 52 0 0 55 > < 0 57 58 59 0 0 0 0 >

51 = Media Wear

The Floppy 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 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. Currently it is set by

a hard disc if, during the execution message, the data transfer was not keeping up with interleave. The host could then use this bit when he is trying to find the optimum interleave to use on the disc.
Since most low cost devices like floppies can't do this, they won't set this bit if the interleave is not being met. The host will have to use another method to determine if the interleave is being met.
The other way the transfer could be impaired during the execution phase, is if the device had to do incremental seek retries. This would also degrade performance, but is less likely to happen. Incremental seek retries during execution phase would also cause this bit to be set.

55 = Auto Sparing Invoked — The device has automatically spared a defective block.

57 = Recoverable Data Overflow — More than one recoverable data error was encountered since the last status request.

58 = Marginal Data — The block whose address is in the parameter field P1 to P6 was recovered, but with difficulty. More than 1 retry was necessary to read the data. See the SPARING STRATEGY.

59 = Recoverable Data — A latency was introduced in order to correct a data error. We had to do 1 retry to read the data.

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

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

1. If there have been no errors, P1 through P6 contains the target address. It will always be single vector.

2. When the Unrecoverable Data bit is set, P1 through P6 contains the single vector address of the bad

block.

3. When the Marginal Data bit is set, P1 through P6 contains the single vector address of the marginal block.

4. When the Recoverable Data bit is set, P1 through P6 contains the single vector address of the recoverable block.

5. After a Spare Block command, P1 through P6 contains the beginning single vector address of the reformatted area.

6. When the Cross-unit (copy error, tapes only) bit is set, P1 through P6 contain the encoded values of each unit which has experienced an error. using 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. He gets gack a QSTAT=1. The host gives a Request Status command. He sees the Cross-unit bit set. Next he 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.

7. 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 SUBSET/80 this byte contains the unit which failed.

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

P7 - P10 In SUBSET/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.

```
********************************************************
***************  RELEASE (  NO OP    )  ***************
********************************************************
```

FORMAT: <00001110>

TYPE: General Purpose

SUPPORT: Core command, all peripherals must support.

TRANSACTION FLOW: Command, Report

DESCRIPTION:

In SUBSET/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 SUBSET/80. See the CS/80 INSTRUCTION SET programming manual for an explanation of Release in CS/80.

```
***************************************************************
****************  RELEASE DENIED (  NO OP  ) ***************
***************************************************************
```

FORMAT: <00001111>

TYPE: General Purpose

SUPPORT: Core command, all peripherals must support.

TRANSACTION FLOW: Command, Report

DESCRIPTION:

In SUBSET/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 SUBSET/80. See the CS/80 INSTRUCTION SET programming manual for an explanation of Release Denied in CS/80.

```
    ************************************************
    ***************  SET ADDRESS  ***************
    ************************************************
```

```
FORMAT: <00010000>  <P1> <P2> <P3> <P4> <P5> <P6>
                      ( 6 parameter bytes )
                      MSB ...................LSB
```

Only single vector addressing is allowed.
Single vector format: 6-byte unsigned binary number

TYPE: Complementary

SUPPORT: Core command, All peripherals must support.

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.

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 Target address is unlike other complementary commands in that it is updated by any command which accesses data, and does not revert back 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 true in Locate and Verify. The target address will point to the next block to be read. If there was 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.

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 SUBSET/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).

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 SUBSET/80.

In the response to Request Status, a SUBSET/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 out side 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 LENGTH  ***************
*********************************************
```

FORMAT: <00011000> < P1 > < P2 > < P3 > < P4 >

(4 parameter bytes)

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

TYPE: Complementary

SUPPORT: Core command, All peripherals must support.

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 ones 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 host does not realise 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 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 did it and we have to be compatible. This means that on a read, verify, write, or copy, 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 or copy command, he 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.

On a tape like the QIC tapes, the length should never be set to all 1's during a read or copy command. The QIC tape's capacity depends on how many sectors are spared during writing and is not a fixed number.

Thus, if the host wants to do a verify from the target address tq the end of volume, he 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 seeks and have no execution phase. No data is transferred in this case.

VARIATIONS FROM CS/80:

1. The new CS/80 does not use "set" or "current" values of length so SUBSET/80 is trying to emulate this improvement to CS/80.

```
***************************************************************
**************  SET RELEASE (NO OP)  **************
***************************************************************
```

FORMAT: <00111011> <TZ000000>

```
     T = 1  Suppress release time-out
     Z = 1  Release automatically during idle time
```

TYPE: Complementary

SUPPORT: Core command, all peripherals must support.

TRANSACTION FLOW: Command, Report.

DESCRIPTION:

SUBSET/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 a SUBSET/80 device it is treated as a No op. The command is accepted and ignored. No error status bits are set.

VARIATIONS FROM CS/80:

See the CS/80 INSTRUCTION SET programming manual to understand the meaning of this command in CS/80. In SUBSET/80 it is treated as a No op. It is included in the command set for compatibility reasons.

```
*************************************************************
*************** SET RETURN ADDRESSING MODE ***************
*************************************************************
```

FORMAT: <01001000> <00000TTT>

```
     TTT = Addressing mode
           000 = single-vector
           001 = three-vector
```

TYPE: Complementary command

SUPPORT: Core command, all peripherals must support.

TRANSACTION FLOW: Command, Report.

DESCRIPTION:

In SUBSET/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 SUBSET/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:

1. SUBSET/80 supports only the single vector addressing mode.

```
************************************************
***************  SET RPS (NO OP)  ***************
************************************************
```

FORMAT: <00111001> < time 1 > < time 2 >

Time 1 = time -to-target in hundreds of microseconds
Time 2 = window size in hundreds microseconds

TYPE: Complementary

SUPPORT: Core command, all peripherals must support.

TRANSACTION FLOW: Command, Report.

DESCRIPTION:

In SUBSET/80 this command is treated as a No op. SUBSET/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 SUBSET/80 it is treated as a No op. It is included in the command set for compatibility reasons.

```
*************************************************
***************  SET STATUS MASK  ***************
*************************************************
```

FORMAT: <00111110> < P1 > . . . < P2 >
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

SUPPORT: Core command, all peripherals must support.

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 "one" 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 his RAM. The status bits set will reflect the mask setting at the time the error occurred.

VARIATIONS FROM CS/80: None

****
*************** SET UNIT ***************
****

FORMAT: <0010YYYY>

YYYY = Unit number   (1111=Device controller)

TYPE: Complementary

SUPPORT: Core command, All peripherals must support.

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 Floppy which is unit 1 and a hard disc which 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 by itself 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.

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.

****

*************** SET VOLUME ***************

****

FORMAT: <01000YYY>

YYY = Volume number

TYPE: Complementary

SUPPORT: Core command, All peripherals must support.

TRANSACTION FLOW: Command, Report.

DESCRIPTION:

This command is used to specify a specific volume within a unit. Most SUBSET/80 devices will have only volume 0. Since volume 0 is the power-on volume, the host never has to give this command. In case this is not true, the Describe command will tell you how many volumes exist on each unit.

Since this is a complementary command, it may be given by itself 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.

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.

VARIATIONS FROM CS/80: None

SPARE BLOCK

FORMAT: <00000110> <0000000T>
sparing mode byte

T = 0 Retain data on reformatted track

T = 1 Do not retain data on reformatted track

TYPE: General Purpose

SUPPORT: Core command, all peripherals must support.

TRANSACTION FLOW: Command, Report.

DESCRIPTION:

This command is used to spare a defective block. The data in the spared block is ALWAYS lost. What happens to the rest of the blocks on the track depends on the T bit.

If T = 0, all blocks on the track are preserved except the spared block. The spared block is moved somewhere. This is taken care of by the device and transparent to the host. In this case the only data lost was the spared block and after the command is finished, a new block has replaced the defective one. The host will then have to fill it up with the appropriate data. The parameter bytes P1 through P6 will contain the address of the spared block and P7 through P10 will contain the number 00H,00H,00H,01H. If the host is not able to retain all the blocks but the defective one, an Unrecoverable Data error will be set and QSTAT will be 1. The host will then have to give the command again with T = 1.

If T = 1, the data on the entire track will be lost. It is just like the whole track was reformatted. The parameter bytes P1 through P7 will contain the begining address of the track and the bytes P7 through P10 will contain the number of blocks on the track.

Some possible status error bits which could be set in response to this command are:

No Spares Available	There are not enough spares left, or this device doesn't do any sparing.
Media Wear	The sparing was done but almost all the spares have been used up.
Unrecoverable Data	T=0 but the peripheral couldn't

retain all the data in the blocks.
(not counting the spared one which
is never retained)

Because of the information contained in the parameter field of the status message, the Request Status command should always be given after a Spare Block command. This will return information about the area affected by the sparing operation and will also clear the status message bits.

For information as to when to use this command, see the section on SPARING STRATEGY.

VARIATIONS FROM CS/80: None

****

*************** 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

SUPPORT: Core command, all peripherals must support.

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.

SUBSET/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 SUBSET/80 devices will periodically check the bus for clears or cancels when doing long procedures like a format. The bus will be checked for these commands at least every 5 seconds.

The sequence of operations performed by a clear is as follows:

1. Abort the current operation at the earliest opportunity such that no data corruption can take place.

2. Clear all clearable device or interface conditions currently asserted. The channel options set by the HP-IB Parity Checking command are not cleared during a clear.

3. Reset all complementary parameters to their power-on values. The selected unit for a Universal Clear, Amigo clear or Channel Independent Clear to unit 15 will be unit 0. For Channel Independent Clears given to unit 0 or 1, unit 0 or 1 , respectively, is still the target unit after the clear.

4. Reset the status report. All status bits are cleared except the Diagnostic Result bit. NO CLEARS EVER CLEAR THE DIAGNOSTIC RESULT BIT. The only way to clear it is to do a Request Status command. Reset the power-on status.

5. Set QSTAT value 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 device should perform whatever recalibration operation it feels it should do, like hard discs restoring the heads to cylinder 0. If the restore takes a long time, it may be possible to speed it up by doing a seek to cylinder 0, then restore.

7. The device may want to reset its HP-IB chip. This is risky if the clear is an Amigo Clear or a Channel Independent Clear. The Untalk and Unlisten after these commands 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. Our recommendation is to not reset the HP-IB chip but to do whatever clearing is possible short of taking it off line. The HP-IB address switches do not have to be updated during a clear.

8. The device lastly enters an optional reporting state. It will accept any command from the host.

VARIATIONS FROM CS/80: None

VALIDATE KEY

FORMAT: <00110001> <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 "TYPICAL HP-IB COMMAND MESSAGE STRUCTURE".

TYPE: Diagnostic

SUPPORT: Floppies and Hard discs must respond to this command.

TRANSACTION FLOW: Command, Execution, Report

DESCRIPTION:

This command results in the 12 byte key sent from the host to the peripheral being compared with the key stored in a secure (non-customer accessable) part of the disc. If the keys are the same QSTAT will be 0. If the keys are not identical, the status error bit 37, No Data Found is set causing a QSTAT of 1.

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

This command enables a host to implement some form of security for certain software products. A downloadable binary program will be made available to software vendors selected by the host divisions that will enable a key to be written in the privileged area on a disc. A software package can then 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 the status bit errors set, or not set will indicate if the key on the disc matches the key passed in the 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.

COMMANDS NOT SUPPORTED

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

COMMAND:	COMMENTS:
1. Cold Load Read	Identical to Locate and Read so it was not needed in the subset.
2. Write File Mark	Tape command, will be supported by low cost tape.
3. Set Block Displacement	
4. Set Burst	
5. Set Retry Time	
6. Set Options	Tape command, may be supported by low cost tape.
7. Copy Data	This definitely will be supported with a low cost tape/hard disc device.
8. Selected Device Clear	Not possible with low cost HP-IB chip. (Since phases of a transaction are secondariès, the 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 8291A doesn't see it because without a secondary address it is not yet correctly addressed and ignores the SDC.
9. Unload	May be supported with low cost tape drive.

```
*******************************************************
***************  SET FORMAT OPTIONS  ***************
*******************************************************
```

FORMAT: <00110001> <P1> <P2>

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

P1 = 11110011 = F3H
P2 = 01011111 = 5FH

The HP-IB sequence is the standard format shown in the section "TYPICAL HP-IB MESSAGE STRUCTURE".

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

TYPE: Utility

SUPPORT: All SUBSET/80 devices will respond to this command. All SUBSET/80 devices will accept and decode this command, whether or not they have any options.

TRANSACTION FLOW: Command, Execution, Report

DESCRIPTION:

This command is used to tell the peripheral to format in a special way. This command is different from most other SUBSET/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 is sent from the host to the peripheral. This single 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 Clear command or the power is cycled to the peripheral.

For example, if you want to format a 9122 microfloppy with 512 byte sectors, you would send this command to the 9122 with the option byte set to 2. If you want to format a 9122 microfloppy with 1024 byte sectors, you would send this command with the option byte set to 3. To determine which byte to send you must consult the user's manual or documentation included with the 9122. This command is product specific.

WARNING! SINCE THIS COMMAND IS PRODUCT SPECIFIC, IF YOU INCLUDE AN OPTION NUMBER IN YOUR SOFTWARE, THAT OPTION NUMBER MAY NOT BE SUPPORTED ON THE NEXT GENERATION PERIPHERAL. SENDING FORMAT OPTIONS REQUIRES KNOWLEDGE OF THE SPECIFIC PRODUCT. A DRIVER IS NOT PARAMETRIC IF IT HAS TO KNOW THAT THE FORMAT OPTION #2 MEANS A FORMAT WITH 512 BYTES ON A 9122. DON'T FIX

THESE OPTION VALUES INTO YOUR PARAMETRIC DRIVER. 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 you send the Set Format Options command with the option byte of OFFH, the SUBSET/80 device will accept this and set no errors if the device has format options. If the SUBSET/80 device does not have any selection of format options, then the device will set a Parameter Bounds error. All SUBSET/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 = OFFH) by setting the Parameter Bounds error. If the peripheral does have some options, it will respond to the Set Format Options command (options byte = OFFH) by setting no errors. Once the driver knows the device has some format options he 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 OOH also specifies the default value. Default values, of course, may be different for different products.

Format options byte	Meaning
OOH	Use product default. See product documentation.
01H - OEFH	These may be valid format options. Their meaning is product specific.
FOH - OFEH	Not used.
FFH	This option byte is used to see if the SUBSET/80 device has any format options.

The meaning of the options byte is defined by a product and you 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 floppy in different ways. It is very dangerous from a command set perspective, for the options byte is only meaningful for certain products and not others. If you put the options bytes into code, that code may have to be changed on the next SUBSET/80 product. 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, you have to be very careful how you use it. If you are not careful, we will be back in the Amigo situation of having to update host code for each new peripheral.

VARIATIONS FROM CS/80: This command is the device dependent Initiate Utility command in CS/80.

for example, 9122 Microfloppy

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	9121 format (256 byte blocks, single sided)

HP-IL MESSAGES

The device's response to each HP-IL message is described below. In SUBSET/80, some of these messages could be classified as Transparent messages since they are used to implement a particular type of channel. However, in this description they will be described separately from the SUBSET/80 messages since their transaction sequence is different from the transaction sequence of SUBSET/80 messages. The loop sequences used should follow the examples given in "The HP-IL System: An Introductory Guide to the Hewlett-Packard Interface Loop".

DATA GROUP

Data Byte : If a talker, will check to see that the byte received was the same as the byte sent. If the frame is not the same as the one sent, then an ETE frame will be sent. If the error occured during the execution phase of a SUBSET/80 command, the device will go to the reporting phase. If a listener, the device accepts the data byte and passes it to the next device. NOTE: In order to make data transfers as fast as possible, the data byte will not be checked until after the next data byte is sent, so the ETE might be sent one frame later.

End Byte : If a listener, accepts the byte and passes it on. In a SUBSET/80 command sequence, the controller of the loop uses an End byte to indicate to the device that the last parameter or opcode is being sent or that the last byte is being sent in a data transfer. The device sends the End Byte in SUBSET/80 to indicate that the last byte during the execution phase or report phase is being sent. The device will not send End Bytes when responding to the HP-IL Ready commmands Send Status, Send Device ID, or Send Accessory ID.

COMMAND GROUP

Interface Clear: The device is removed from talker or listener status.

Device Clear: The device will clear itself. This command is the same as the SUBSET/80 Universal Device Clear command. For a complete description of the sequence of operations performed by a clear, refer to the SUBSET/80 documentation.

Selected Device Clear. If listener, will have same response as the Device Clear command.

Go To Local: No response.

Local Lockout: No response.

Remote Enable: No response.

Not Remote Enable: No response.

Parallel Poll Enable 0-15: If a listener, the device will enable its parallel poll response capability according to the last four bits of the message. The device's power-on state is its parallel poll response capability is disabled.

Parallel Poll Disable: If a listener, the device will disable its parallel poll response capability.

Parallel Poll Unconfigure: The device will disable its parallel poll response capability.

Group Execute Trigger: No response.

Loop Power Down: Pompeii will immediately go into its power down mode.

Enable Asynchronous Requests: No response.

Auto Address Unconfigure: The device's address is set to the default value of 2.

Listen Address 0-30: If the address matches, the device is removed from talker status and becomes a listener.

Unlisten: Device is removed from listener status.

Talk Address 0-30: If address matches, device is removed from listener status and becomes a talker. If the address doesn't match, the device is removed from talker status.

Untalk: The device is removed from talker status.

Device Dependent Listener 0-31: These commands are used by the FILBERT command set only. The device will respond to these commands as described in the FILBERT command set description.

Device Dependent Talker 0-31: These commands are used by the FILBERT command set only. The device will respond to these commands as described in the FILBERT command set description.

Secondary Address 0-31: These commands are used by SUBSET/80 only. They are used to indicate the phase of a transaction. If the device is a listener or a talker, then the device will interpret the command and take the proper action.

Null: No response.

READY GROUP

Take Control: No response.

Ready For Command: The HP-IL chip will handle the automatic response to this command.

Send Data: If the device is a talker and has received the proper secondary command or device dependent command, the device will begin sending data. If the device is a talker and has not received the proper secondary or device dependent commmand, it will send back an ETO message.

Send Status: If talker, sends the FILBERT status bytes as outlined in the FILBERT command set description.

Send Device ID: If talker, sends the bytes for device ID.

Send Accessory ID: If talker, sends one byte with value of 010H as the accessory ID. SUBSET/80 hosts must do a SDI or a SUBSET/80 command to determine if the device also implements the SUBSET/80 command set.

Not Ready For Data: This command is not fully supported in SUBSET/80. In SUBSET/80, if the device receives a NRD message while it is sending data, it will retransmit the NRD message and then send a EOT message when it receives the last data byte it sent. This will cause a message length error if this occurs during a SUBSET/80 data transfer, unless the data byte just sent was the last byte of the transfer. The device will not start sending data where it left off if it receives another SDA message. In FILBERT, this command will be supported in full.

End Of Transmission-OK: If talker, sent after a data transfer has completed successfully.

End Of Transmission-Error: If talker, sent after a bad HP-IL error check.

Auto Address 0-31: If device has earlier address or the address is 31, then no resoponse. Otherwise, the device accepts the address as its own, increments the message by one, and retransmits it.

Auto Extended Primary 0-31: No response.

Auto Extended Secondary 0-31: No response.

Auto Multiple Primary 0-31: No response.

IDENTIFY GROUP

Identify (no service request): The device automatically retransmits the message with the SRQ bit and the parallel poll bits set according to the devices current status.

Identify (with service request): Same as Identify with no service request.

SUBSET/80 COMMAND SET

The CS/80 and SUBSET/80 instruction set programming manuals contains an overview of the command set and should be referred to for a complete description of the command set. This section will describe the specific actions the device will take in response to each of the commands. First, a list of the commands is given and then a brief description of each command in alphabetical order is given.

As mentioned above, the HP-IL messages are not included in SUBSET/80 description. The transparent commands included in this description all have the same type of transaction sequence as the other SUBSET/80 messages. The SUBSET/80 Transparent messages Universal Device Clear, Selected Device Clear, and Identify are not implemented since they have corresponding HP-IL messages. The Loopback commands are not implemented since with HP-IL the loop is checked with every message sent. The Channel Independent Clear command is used to allow only one unit within a device to be cleared, and the HP-IB Parity Checking command is used to allow the SRQ response to be enabled and disabled. The SUBSET/80 command set does not require that the Transparent commands be implemented since they are channel dependent, so the command set implemented here meets the standards.

SUBSET/80 COMMAND TABLE

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
Complementary:	Set Unit
	Set Volume
	Set Address
	Set Length
	Set Mask
	Set RPS (No Op)
	Set Release (No Op)
	Set Return Addressing Mode
	No Op
Transparent:	Cancel
	Channnel Independent Clear
	HP-IB Parity Checking
Diagnostic:	Initiate Diagnostic
	Request Status
Device Dependent:	Validate Key
	Download
	Set Format Options

CANCEL

Opcode: 09H

Parameters: None

Description: This command causes a graceful termination of long transactions, leaving the device in the reporting phase.

The device will look for a cancel during long command executions such as Initialize Media. Cancel will not cause an immediate response usually, but eventually it will be seen and acted upon.

The Cancel command supresses message length errors.

CHANNEL INDEPENDENT CLEAR

Opcode: 08H

Parameters: None

Description: If this command is directed to the controller or to the microfloppy, the same actions are taken as described in the HP-IL Device Clear command.

DESCRIBE

Opcode: 35H

Parameters: None

Description: The controller fields returned for the 9114A are:

C1, C2 = installed unit byte. C1 will always be
80H, C2 will be 1 corresponding to 1 drive.
C3, C4 = 7, maximum instantaneous transfer rate.
C5 = 4 for one drive being connected.

The unit description for the microfloppy will be:

U1 = Generic device type 129, since disc change is
not supported
U2-U4 = 091140, product number
U5-U6 = variable, depending on bytes per block
of current disk
U7 = 1, number of blocks which can be buffered
U8 = 0, burst mode not implemented
U9-U10 = variable, depending on current disk
U11-U12 = 7 Kbytes/sec maximum continuous average
transfer rate
U13-U14 = 3500, 35 seconds read retry time
U15-U16 = 8000, 30 seconds maximum access time
U17 = variable, depending on current disk
U18 = 0, no fixed volumes
U19 = 1, one removable volume

The volume description field is:

V1-V3 = 0
V4 = 1, maximum head address for double-sided, 0
for single-sided discs
V5-V6 = variable, sectors per track
V7-V12 = variable, maximum sector address
V13 = current interleave factor or maximum value
if unknown

To determine the correct values for the volume description field, the disc is accessed. Cylinder 0 on head will be read to get the current interleave, disc type ID, etc. If this track is unformatted, the bottom surface of the disc will beaccessed. If it's formatted as single-sided, the volume description field parameters will be returned as such. If neither side of the disc is formatted, the disc will be regarded as the default type of disc. If no disc is in the drive, V1-V6 will reflect the default format, but the addr. field, V7-V12, will be zero.

DOOR LOCK

Opcode. 4DH

Parameters: None

Description: This command will cause the illegal opcode error bit to be set. Drivers can use this command to see that the devices do not have door locks.

DOOR UNLOCK

Opcode: 4CH

Parameters: None

Description: This command will cause the illegal opcode error bit to be set. Drivers can use this command to see that the devices do not have door locks.

DOWNLOAD

Opcode: 31H

Parameters: P1 = F2H
P2 = A5H
P3-P5 = Device number
P6 = Download revision number in unsigned binary

Description: This command is used to download code into the RAM of the device, which will then be executed. This command is mainly used for special service routines and diagnostics.

HP-IB PARITY CHECKING

Opcode: 01H

Parameters: P1 = 00H to disable SRQ response
02H to enable SRQ response

Description: This command is used to enable and disable the device's SRQ response. The device's power-on state is SRQ disabled.

INITIALIZE MEDIA

Opcode: 37H

Parameters: P1 = byte indicating how sparing should be handled
P2 = block interleave number

Description: This command will initialize a media, one track at a time. The procedure consists of:

1) writing a worst case pattern
2) verifying the pattern
3) formatting the track with the desired interleave or sparing the track if defective
4) verifying the pattern
5) step to the next track

None of the spares present on the disc at format time are preserved. If a track is truly defective, it will be spared again by this Initialize Media procedure.

There are two spare tracks on each side of the double-sided disc. If more tracks need to be spared than provided for by these set-aside tracks, the command will end in error.

The disc can be formatted with an interleave from 1 to the maximum sectors per track. An interleave of 0 is the same as an interleave of 1. If the the interleave parameter passed is greater than the maximum number sectors per track, the maximum will be used.

Periodically during the execution of the Initialize Media command, the HP-IL will be scanned. If a Clear or Cancel command is recognized, the Initialize Media command will be terminated with no errors.

Some possible status errors are:

35 = Not Ready	No disc in drive
36 = Write Protect	Disc is write protected
34 = No Spares Available	Ran out of spare tracks
55 = Auto Sparing Invoked	At least one track was spared

INITIATE DIAGNOSTIC

Opcode: 33H

Parameters: P1 = 00H
P2 = 01H
P3 = 00H

Description: This command instructs the device to perform its diagnostic routine. This routine is the same as that performed at power-on.

LOCATE AND READ

Opcode: 00H

Parameters: None

Description: This command finds the data at the target address and transmits it to the host.

If the present location of the read/write head is not at the target track, the head is stepped to the proper cylinder and the sector ID is read to verify the location of the head. The actual sector(s) ID is not verified at this time, only the track ID. If any error occurs up to this point, the proper status bits are set up, QSTAT is set to 1, and the reporting phase is entered, skipping the execution phase.

Once the proper track is located, the execution message is requested, unless the current length is 0, in which case no execution message is requested.

At least one sector (block) of data is read from the disc and then passed to the host. The cycle of reading a sector or data and then passing it to the host over the HP-IL will continue until the number of bytes as set by the current length parameter is passed.

If the correct sector for a read cannot be found, a restore to track 0 followed by a seek for the correct track is performed. Re-seeking will be performed up to five times.

If the data is read with a CRC error, up to two retries will be performed before the read ends in error. In all cases, all the data requested as defined by the length parameter will be passed to the host.

Some possible status errors:

35 = Not Ready	No disc in drive
33 = Unitialized Media	Unformatted disc present
44 = End of Volume	Reads extend to end of disc
41 = Unrecoverable Data	Seek fails or data error
40 = Unrecov. Data Overflow	More than one error 41
59 = Recoverable Data	Seek or read succeeds after retries
52 = Latency Induced	Seek or read succeeds after retries
57 = Recov. Data Overflow	More than one error 59

LOCATE AND VERIFY

Format: 04H

Parameters: None

Description: This command instructs the device to perform an internal verification of a section of data to ensure that it can be read.

This command is basically the same as a Locate and Read except that 1) the data is not made available to the host, 2) retries on reading the data are not performed, 3) the device stops reading after the first error.

Some possible status errors:

35 = Not Ready	No disc in drive
33 = Uninitialized Media	Unformatted disc
44 = End of Volume	Verify went to end of disc
41 = Unrecoverable Data	Seek fails or data error
40 = Unrec. Data Overflow	More than one error 41

LOCATE AND WRITE

Opcode: 02H

Parameters: None

Description: This command transfers data from the host to the microfloppy, starting at the target address.

If the present location of the read/write head is not at the target track, the head is stepped to the proper cylinder and the sector ID is read to verify the location of the head. The actual sector(s) ID is not verified at this time, only the track ID. If any error occurs up to this point, the proper status bits are set up, QSTAT is set to 1, and the reprorting phase is entered, skipping the execution phase.

Once the proper track is located, the execution message is requested, unless the current length is 0, in which case no execution message is requested.

The actual writing of the data occurs once the sector buffer is filled with data from the host. This cycle of filling the buffer and then transferring the data to the microfloppy will continue until the number of bytes as specified in the length parameter is written on the microfloppy. If only a portion of a sector's worth of data is passed by the host, the remainder of the sector is filled with zeros.

Some possible status errors:

35 = Not Ready	No disc in drive
33 = Uninitialized Media	Unformatted disc present
36 = Write Protect	Disc is write protected
44 = End of Volume	Reads extend to end of disc
41 = Unrecoverable Data	Seek fails or data errror
40 = Unrecov. Data Overflow	More than one error 41
59 = Recoverable Data	Seek succeeds after retries
52 = Latency Induced	Seek succeeds after retries
57 = Recov. Data Overflow	More than one error 59

NO OP

Opcode: 34H

Parameters: None

Description: This command is ignored.

RELEASE (NO OP)

Opcode: 0EH

Parameters: None

Description: This command will be treated as a No Op.

RELEASE DENIED (NO OP)

Opcode: 0FH

Parameters: None

Description: This command will be treated as a No Op.

REQUEST STATUS

Opcode: 0DH

Parameters: None

Description: This command instructs the device to return the status report.

See te SUBSET/80 command set description for details.

There are certain staus bits that can be set by all the commands. These are:

5 = Illegal Opcode
9 = Illegal Parameter
10 = Message Sequence
12 = Message Length
19 = Controller Fault
22 = Unit Fault
30 = Power Fail

The Power Fail status will indicate that a new disc has been inserted into the drive.

SET ADDRESS

Opcode: 24H

Parameters: 6-byte unsigned binary number for single vector address, MSB first

Description: This command sets the value of the single vector target address.

The target address is incremented after each read, whether it was successful or not and will point to the block after the one just read.

Some possible status errors:

7 = Address Bounds	Address passed is too large

SET FORMAT OPTIONS

Opcode: 31H

Parameters: P1 = F3H
P2 = 5FH

Description: This command is used to tell the peripheral to format the media in a special way. The execution message consists of a single byte that specifies the format. This format option remains selected until power is cycled or a clear command is executed. The following format options are supported.

Format option byte	Format written
0 (default)	512 byte sectors 9 sectors/track 2 sides 77 tracks of data/side
1	256 bytes/sector 16 sectors/track 2 sides 77 tracks of data/side
2	Same as 0
3	1024 bytes/sector 5 sectors/track 2 sides 77 tracks of data/side
4	256 bytes/sector 16 sectors/track 1 side 66 tracks of data/side (this is HP single_sided format)

SET LENGTH

Opcode: 18H

Parameters: P1-P4 = unsigned binary number, MSB first (except all 1's means entire volume)

Description: This byte defines the number of bytes in a data transfer. No matter what the current length parameter is set to, all read and write operations to the micr floppy will be done on a full sector basis. The number of data bytes transferred over the HP-IL will be as defined by the length parameter.

SET STATUS MASK

Opcode: 3EH

Parameters: P1-P8 = Bit positions in each byte corresponds to the error bit position in the error reporting fields of the status report. "1" means to mask the error.

Description: This command allows masking of error conditions reported by the Request Status command. The 8 bytes following the command opcode indicate which error bits are to be masked. At power-on, no bits are masked.

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

Some possible status errors:

8 = Parameter Bounds	Trying to mask an unmaskable bit

SET RELEASE

Opcode: 3BH

Parameters: 2 bytes

Description: This command will be treated as a NO Op.

SET RETURN ADDRESSING MODE

Opcode: 48H

Parameters:

Description: This command should not be used by new drivers.

SET RPS

Opcode: 39H

Parameters: 2 bytes

Description: This command will be treated as a No Op.

SET UNIT

Opcode: 2XH

Parameters: None

Description: This command is used to specify a specific unit within the device. The controller is always unit 15. The microfloppy will be unit 0.

Some possible status errors:

6 = Module Addressing	Illegal unit number

SET VOLUME

Opcode: 8XH

Parameters: None

Description: This command is used to specify the desired storage volume of a specified mass storage device. In this device there is only one volume, so the only valid volume number is 0.

Some possible status errors:

6 = Module Addressing	Volume not equal to 0

SPARE BLOCK

Opcode: 06H

Parameters: One byte

Description: This command instructs the device to spare out the track indicated by the target address.

The device's response will be:

34 = No spares available

VALIDATE KEY

Upcode: 32H

Parameters: P1 = f1H
P2 = 02H

Description: This command causes the device to compare the key passed during the execution phase to the key stored on the microfloppy. If the keys agree QSTAT will be 0.

This command enables a host to implement some form of security for certain software products. A downloadable binary program will be made available to software vendors selected by the host divisions that will enable a key to be written in the privileged area on a disc. A software package can then 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 QSTAT will indicate if the key on the disc matches the key passed in th command. Presumably, the applications software will abort the program if the keys don't match.

Some possible status errors:

37 = No Data Found	Incorrect key or no key

FILBERT COMMAND SET

The extended Filbert command set will also be supported on both Pompeii and Buzzard. This can be done since the two command sets use different HP-IL commands. SUBSET/80 uses secondary addresses to indicate phases in a transaction and Filbert uses device dependent commands.

DEVICE COMMMANDS

The device commands in the extended Filbert command set are listed below. The commands are split into two groups. Listen commands are executed when addressed as a listener, and talk commands are executed when addressed as a talker. No action is taken on any device dependent command until the RFC following it has been received. If any other frame follows the device dependent command, then the command is not executed. Any DDL or DDT commands other than the ones listed below will cause all states to be cleared

LISTEN COMMANDS

WRITE TO BUFFER 0	A0
WRITE TO BUFFER 1	A1
WRITE	A2
SET BYTE POINTER	A3
SEEK	A4
FORMAT	A5
PARTIAL WRITE	A6
HOME (REWIND)	A7
CLOSE RECORD	A8
TRANSFER BUFFER 0-1	A9
EXCHANGE BUFFER	AA
VERIFY RECORDS	AB
DOWNLOAD	AC

TALKER COMMMANDS

SEND BUFFER 0	C0
SEND BUFFER 1	C1
READ	C2
SEND ADDRESS	C3
EXCHANGE BUFFERS	C4
TRANSFER BUFFER 0-1	C5
SEND PHYSICAL ATTRIBUTES	C6
SEND MAXIMUM ADDRESS	C7

LISTEN COMMANDS DESCRIPTIONS

WRITE BUFFER 0

Data bytes following this command are stored in BUFFER 0 starting at the location pointed to by the BYTE POINTER. When the buffer fills, the contents of the buffer are written to the NEXT RECORD. Following bytes fill the buffer from the beginning again. This command is the same as the WRITE command except that the BYTE POINTER and PARTIAL WRITE flag are not cleared when the command is received.

WRITE BUFFER 1

Bytes are stored in BUFFER 1. No action is taken when the buffer fills. The BYTE POINTER wraps around and extra bytes are written over the bytes at the beginning of the buffer. The PARTIAL WRITE flag is cleared.

WRITE

This command sets flags that cause the buffer contents to be written to the NEXT RECORD when an END frame or CLOSE RECORD is received, or the buffer fills. After the buffer is written to the NEXT RECORD, the record after the one just written becomes the NEXT RECORD. The BYTE POINTER and PARTIAL WRITE flag are set to zero when this command is received. If the last record on the media is written, the L STATE mode is cleared and NEXT RECORD is set to 0. If another DAB is received, size error is set. If a CLOSE RECORD is received just after the buffer has filled, the contents of the buffer will be written to the NEXT RECORD resulting in two sequential records containing the same data (the buffer contents have not been changed since the previous record was written).

SET BYTE POINTER

Data bytes received after this command are put in the BYTE POINTER register. The last byte received is the only valid byte. The data frame is taken as a binary value. The bytes of each buffer are numbered from 0 to 255D.

SEEK

This command sets the NEXT RECORD. The PARTIAL WRITE flag is cleared. A two byte binary number specifies the NEXT RECORD. The first byte received is the most significant. If only one byte is sent the command is aborted. Bytes sent after the first two are disregarded. If a record number larger than the last addressable record is received, a size error is generated and the NEXT RECORD remains the same.

FORMAT

Immediately after receiving this command the entire media is initialized. The format used will be 256 bytes/sector, 16 sectors per track, 2 sides, and 77 data tracks per side. The Subset/80 command Set Format Options can be used to change the format to single_sided HP format if desired. The interleave used will be 9. This is the only time that record headers are written on the media. The PARTIAL WRITE flag is cleared and the contents of BUFFER 0 and 1 are destroyed. Format will fail and write protect the media if motor speed is not within specifications. Format will take between 1 and 2 minutes to complete with double-sided drives. Due to errors in the format routines of certain hosts, a check is done after a write to the first sector to make sure that if LIF format is being written on the disc after the format that the correct extension fields and directory size are written. If not, then these are overwritten with the correct data.

PARTIAL WRITE

When this command is received the NEXT RECORD is read. The NEXT RECORD remains the same. The head remains over the track containing the NEXT RECORD. The PARTIAL WRITE flag is set. The following rules apply when in PARTIAL WRITE mode:

A) After every read the record just read remains the NEXT RECORD.

B) After every write:

if the buffer was filled the NEXT RECORD number is incremented, the new NEXT RECORD is read, and the head remains over the track containing the new NEXT RECORD.

if the buffer was not filled the NEXT RECORD remains the same and the head remains on the same track.

Writes occur when the buffer fills, or when an END frame or CLOSE RECORD is received.

The buffer is filled when the BYTE POINTER rolls over to zero when no END frame or CLOSE RECORD is received, or when the BYTE POINTER is pointing to the last byte in the buffer and an END frame is received. (The END frame is placed in the last byte of the buffer resulting in a full buffer.)

The buffer is not filled if the BYTE POINTER is not at the end of the buffer when an END frame is received or if a CLOSE

RECORD is received. (The buffer will never be full when a CLOSE RECORD is received.)

If the BYTE POINTER is at the first byte of the buffer and a CLOSE RECORD is received, the buffer is written back to the NEXT RECORD. (The buffer contents are the same as the NEXT RECORD.)

If the maximum addressable record is written and the NEXT RECORD gets incremented then the LSTATE mode will be cleared, the NEXT RECORD will be set to 0, and size error will be generated if another DOF frame is received.

HOME (Previously REWIND)

This command sets the NEXT RECORD to 0 and restores the drive to track 0.

CLOSE RECORD

This command causes the buffer contents to be written to the NEXT RECORD NEXT RECORD. If PARTIAL WRITE mode is active the NEXT RECORD number is number remains the same. If not active the NEXT RECORD number is incremented. If the WRITE BUF 0 flag has been cleared then no action is taken. END frames act the same as CLOSE RECORD unless PARTIAL WRITE mode is active.

TRANSFER BUFFER 0-1

BUFFER 0 is copied into BUFFER 1. The contents of BUFFER 1 are lost. BUFFER 0 is used for buffering records on and off the media.

EXCHANGE BUFFERS

The contents of BUFFER 0 and BUFFER 1 are exchanged.

VERIFY RECORDS

The sector headers and data are read and verified with their checksums. The first record checked is the NEXT RECORD. This should be set by the SEEK command before verifying records. A two byte binary number specifies how many record to verify. The first byte received is the most significant. If only one byte is sent the command is aborted. If records greater than the maximum addressable record are to be verified then the verification stops with the last record on the media and NEXT RECORD is set to 0. If the verify fails, the NEXT RECORD will contain the address of the sector that failed.

DOWNLOAD

The first DAB received after this command is the starting address MSB, the second byte is the LSB. Successive bytes are loaded into memory starting at this address. When an END frame is received a subroutine call is made to the location specified by the starting address. This command is intended for service use only. No range checking is done.

TALK COMMANDS DESCRIPTION

SEND BUFFER 0

When this command and a subsequent SDA are received the BUFFER 0 contents are sent starting with the byte pointed to by the BYTE POINTER. When the entire buffer is sent the NEXT RECORD is read into the buffer and sent. THe NEXT RECORD is then incremented. This continues until the maximum addressable record is sent, an NRD sequence has been executed, or an error condition is detected. If the maximum addressable record is sent and an NRD sequence has not been executed, an EOT is sent and the NEXT RECORD is set to 0.

SEND BUFFER 1

When this command and a subsequent SDA are received the BUFFER 1 contents are sent starting with the byte pointed to by the BYTE POINTER. Sending stops when the entire buffer has been sent or an NRD sequence has been executed.

READ

The NEXT RECORD is read and placed in BUFFER 0. The NEXT RECORD number is incremented. The BYTE POINTER is set to zero. The contents of BUFFER 0 are then sent. When the last byte has been sent the new NEXT RECORD is read and sent. This continues until the maximum addressable record has been read and sent or an NRD sequence has been executed. If the maximum addressable record is sent and an NRD sequence has not been executed, an EOT is sent and the NEXT RECORD is set to 0.

SEND ADDRESS

The NEXT RECORD number (two bytes) and the value of the BYTE POINTER (one byte) are returned after this command and a subsequent SDA are received. The order is most significant byte, least significant byte, and then the BYTE POINTER.

EXCHANGE BUFFERS

This command is the same as the DDL EXCHANGE BUFFERS.

TRANSFER BUFFER 0-1

This command is the same as the DDL TRANSFER BUFFER 0-1.

SEND PHYSICAL ATTRIBUTES

Twelve bytes of the LIF Level 1 extension field are returned after this command and a subsequent SDA are received. The most significant byte of word 12 is sent first, then the least significant byte of word 12. This is done for words 12 to 17. (See LIF document page 15)

SEND MAXIMUM ADDRESS

Two bytes representing the record number (in binary) of the last (highest) physical record on the media are returned after this command and a subsequent SDA are received. The most significant byte is sent first.

STATUS

One byte of status information will be returned when the Send Status command is received. The following table outlines the possible status messages that could be returned. If the device is busy, status message of 32D will be returned.

Status Byte	Condition	Description
0	Idle	Everything is OK
15	Stall	A seek was attempted to a blank media.
20	No media	No media is in drive
21	Low battery	Low battery
23	New media	A new disc is in the drive
24	Blank	A foriegn or unformatted media is in the drive
25	Record number error	The record could not be found
26	Checksum error	A CRC error accurred on a read or too many tracks

		were spared by FORMAT
28	Size error	An access to a record greater than the maximum addressable record was attempted
29	Write protect	Disc is write-protected

MESSAGE STRUCTURES

The messages that are used were separated into two distinct groups, HP-IL messages and SUBSET/80 messages. This was done because the message sequence used when a SUBSET/80 message is sent always has a certain structure. SUBSET/80 transactions may consist of three separate phases: command, execution, and reporting. Each SUBSET/80 phase is indicated by the controller sending the proper secondary address command that corresponds to the phase of the transaction. Transactions in which only HP-IL messages are sent will not follow this structure.

The HP-IB sequences used during a SUBSET/80 transaction is outlined in the CS/80 Instruction Set Programming Manual. These same sequences will be used as they apply to HP-IL to make the HP-IL implementation as close as possible to the HP-IB implementation. The main difference occurs when the device starts and stops sending data. The controller must send the Send Data message to the device when it wants the device to send data and the device must send an End Of Transmission message when it is done sending data. The controller will always send the last data byte as an End Byte. The device will use this byte as a signal that the last parameter or opcode has been sent during the command phase of a sequence, or that the last byte has been sent during the execution phase of a sequence. The device will send its last data byte during the execution phase and report phase as an End Byte, also.

The sequences used during a Locate and Read transaction and a Locate and Write transaction are shown below. These sequences should show exactly how the HP-IB seqences map into the HP-IL sequences.

LOCATE AND READ

Command Phase:

LADn — First, the controller addresses the device as a listener.

RFC

SAD5 — Next, the controller sends out the secondary address for the command phase of the transaction.

RFC

DABxx — The controller then sends 0 to N complemetary commands to the device (xx = the value of the data byte).

.

.

.

DABxx

END00 The controller sends the opcode for Locate and Read as an End Byte.

UNL The controller unlistens the device.

RFC

Execution Phase:

TADn The controller addresses the device to be a talker.

RFC

SAD14 The controller sends out the secondary address to indicate the execution phase of the transaction.

RFC

SDA The controller sends the Send Data message which the device replaces with its first data byte.

DABxx The device sends 1 to N data byte.

.

.

DABxx

ENDxx The last data byte will be sent as an End byte.

ETO The device sends the End Of Transmission message to signal to the controller that it is done.

UNT The controller untalks the device.

RFC

Reporting Phase:

TADn The controller addresses the device to talk.

RFC

SAD16 The controller sends the secondary address for the reporting phase of the transaction.

RFC

SDA The controller sends the Send Data message which the device replaces with the QSTAT byte (xx = QSTAT) sent as an End Byte.

ENDxx

ETO The device sends the End Of Transmission message to indicate it is done.

UNT — The device is untalked.

RFC

LOCATE AND WRITE

Command Phase:

LADn — First, the controller addresses the device as a listener.

RFC

SAD5 — Next, the controller sends out the secondary address for the command phase of the transaction.

RFC

DABxx — The controller then sends 0 to N complemetary commands to the device (xx = the value of the data byte).

.

.

.

DABxx

END02 — The controller sends the opcode for Locate and Write as an End Byte.

UNL — The controller unlistens the device.

RFC

Execution Phase:

LADn — The controller addresses the device to be a listener.

RFC

SAD14 — The controller sends out the secondary address to indicate the execution phase of the transaction.

RFC

DABxx — The controller sends out 1 to N bytes of data to be written, with the last one transmitted as an End Byte.

.

.

DABxx

ENDxx

UNT — The controller untalks the device.

RFC

Reporting Phase:

TADn — The controller addresses the device to talk.

RFC

SAD16 — The controller sends the secondary address for the reporting phase of the transaction.

RFC

SDA — The controller sends the Send Data message which the device replaces with the QSTAT

ENDxx — byte (xx = QSTAT) sent as an End Byte.

ETO — The device sends the End Of Transmission message to indicate it is done.

UNT — The device is untalked.

RFC