

HP7908, HP7911, HP7912, AND HP7914

TAPE DRIVE & CARTRIDGE

EXTERNAL

REFERENCE

SPECIFICATION

October 15, 1984

**THE LINUS, THIRD ANNUAL, FINAL ERS (REV. G)**

**Component Specification -- Cartridge Tape Drive Oct 15, 1982**

<b>1.0 SCOPE . . . . .</b>	<b>1</b>
<b>2.0 DESCRIPTION AND FEATURES</b>	
2.1 General Description . . . . .	1
2.2 Product Features . . . . .	1
2.2 Productsm Description . . . . .	1
2.4 Cartridge Description . . . . .	2
<b>3.0 OPERATOR INTERACTION</b>	
3.1 Controls and Indicators . . . . .	3
3.2 Cartridge Loading Sequence . . . . .	4
3.3 Cartridge Unloading . . . . .	4
<b>4.0 HOST SYSTEM INTERACTION</b>	
4.1 Description of Command Sequence . . . . .	5
4.2 Linus File Mark Capability . . . . .	5
4.3 Defect Handling . . . . .	6
4.4 Commands Supported by Linus	
4.4.1 Real Time Commands . . . . .	6
4.4.2 Complimentary Commands . . . . .	8
4.4.3 General Commands . . . . .	9
4.5 Linus Utilities and Diagnostics . . . . .	11
4.5.1 Request Status . . . . .	12
4.5.2 Initiate Diagnostic . . . . .	12
4.5.3 Execute Utility . . . . .	13
4.6 Power On Sequence . . . . .	19
<b>5.0 PERFORMANCE SPECIFICATIONS . . . . .</b>	<b>21</b>
<b>6.0 ENVIRONMENTAL . . . . .</b>	<b>22</b>
<b>7.0 RELIABILITY . . . . .</b>	<b>23</b>
<b>8.0 PHYSICAL SPECIFICATIONS . . . . .</b>	<b>24</b>
<b>9.0 EXAMPLES OF LINUS MEDIA DEFECT HANDLING . . . . .</b>	<b>25</b>
<b>10.0 LIF COMPATIBILITY . . . . .</b>	<b>28</b>

## 1.0 SCOPE

This specification provides engineering documentation and system user information of the operating features and performance characteristics designed into this product.

## 2.0 DESCRIPTION AND FEATURES

### 2.1 General Description

Linus is a low cost cartridge tape drive intended for back-up of low to medium range discs. Secondary uses will be a data exchange medium between HP systems, software distribution, in data/transaction logging applications, and for personal I/O.

The Linus subsystem consists of a "smart" mechanism, a half size Amigo P.C.Assy., and preformatted, removable tape cartridges. As a component, this subsystem will be integrated into the 16 Mbyte 7908, the 28 Mbyte 7911, the 65 Mbyte 7912, and the 132 Mbyte 7914 disc systems. In these products, the packaging, power supplies, and controller will be shared with the discs.

### 2.2 Product Features

Up to 67.0 Mbytes of formatted capacity on tape cartridges preformatted into 1K byte data blocks.

"Exclusive Or" error correction.

Linus and its associated disc are capable of direct data transfers without involving the HP-IB.

Mechanical cartridge interlock engaged during tape operations.

Data blocks on any portion of tape are randomly accessible and over-writable.

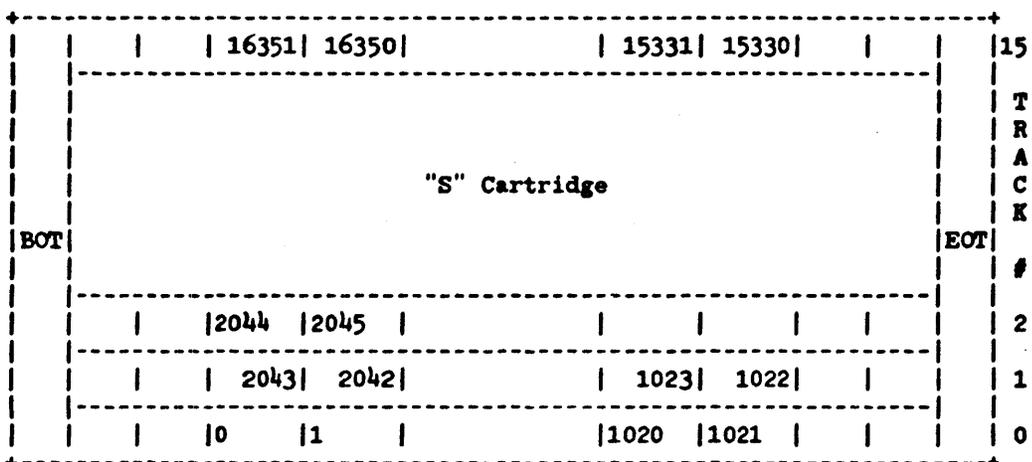
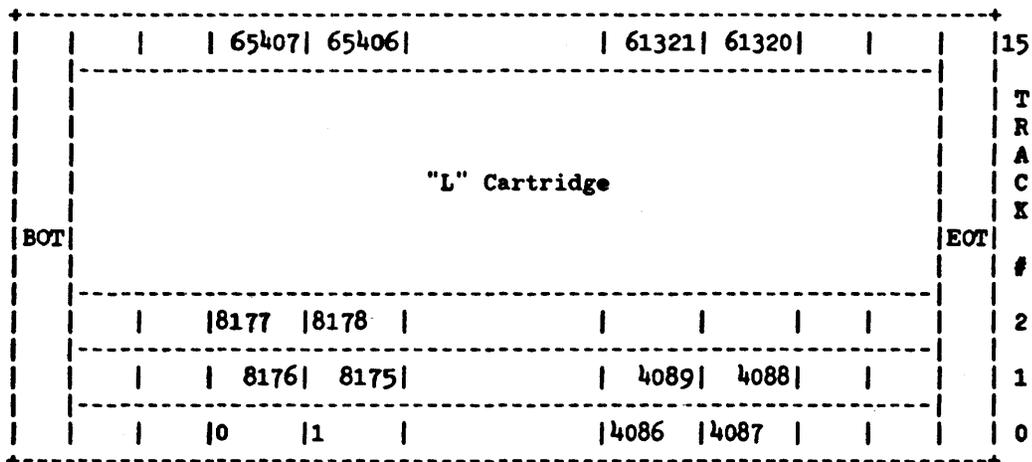
Uses command set '80 disc protocol (ie locate and read, locate and write, etc.).

### 2.3 Mechanism Description

The tape drive module consists of the drive mechanism plus motor and head control electronics packaged together within the mechanism frame. The high capacity is achieved by positioning a single track, ferrite head via a stepper motor/lead screw to 16 track locations across the tape. The data is recorded serpentine fashion (each track recorded the opposite direction as the adjacent track) to eliminate rewind times between tracks. Head positioning and capstan motor digital servo control are accomplished via a microprocessor resident in the mechanism. The microprocessor also tests and calibrates the mechanism every time a cartridge is installed.

## 2.4 Cartridge Description

The cartridge is a DC600 preformatted data cartridge. (This is a 4x6 in. version of the HP mini-cartridge used in HP calculators and terminals!) Two sizes are available: The 67MB "L" cartridge with 600 feet of tape, and the 16.7MB "S" type with 150ft. of tape. The formatted blocks include space for 1K of data, error correction, block addresses, and CRC's. A mechanical write protect key is built into the cartridge also.



<----->	<-----User Space----->	<----->
spare table	(Including Spares)	use logs
error log &		
diag.blocks		

### 3.0 OPERATOR INTERACTION

#### 3.1 Controls and Indicators

UNLOAD button - This button is mounted on the front panel of the tape mechanism and provides the user a way of indicating to the controller that the tape should be unloaded. When the controller senses that the button has been depressed it will:

- a) During an autoloader - stop the autoloader and UNLOAD.
- b) During a SAVE or RESTORE operation - temporarily stop those operations. If it is then pushed a second time and while the busy light is flickering, an UNLOAD will occur. If it is not pushed in 5 sec., the SAVE or RES will resume.
- c) During any other active operation - invoke the unload sequence (below).

The unload sequence first requests release from the host system when release is granted, the controller updates the error logs (near the front of tape), rewinds the tape to "End of Tape" (EOT), updates the Use log, and unlocks the cartridge with an audible buzz.

EJECT lever - This slide lever ejects the cartridge out of the drive mechanism. A mechanical interlock prevents its actuation unless the cartridge has "unloaded" as described above.

\*SAVE button - This button is located behind the front panel. It is intended for use by service personnel and users when the host system is down and a copy of the disc is desired. Backup is normally done with system commands.

Pushing the SAVE button will initiate a full volume transfer of the disc's contents to the in-place tape cartridge. The transfer will start from logical block zero on the disc and tape, and proceed until the disc volume has been completely transferred, or end of volume occurs on the tape. After the last block is written, a file mark is added. Mismatched volume sizes between the disc and tape are allowed within the above definition.

The 7914 will, after the first tape, prompt the user to load another tape. With this second tape the controller restarts the copy from where it left off on the disc. The address of the first disc sector stored on each tape is recorded in a tape maintenance block.

The SAVE operation starts with a request for release from the host. After release has been granted the busy light is flickered. If the button is pushed again within 5 seconds, the controller will start copying the disc data onto the tape. If no button or the wrong button is pushed, the subsystem returns to its idle state. If a data error is encountered on the disc, the best guess is sent to the tape, and the SAVE continues to completion. However, the tape is not unloaded and the flashing busy light indicates a fault. Pushing the unload button now, unloads the cartridge and clears the fault conditions. Unreadable keys on the tape are marked in the spare table and skipped. The SAVE continues to conclusion with no fault indicated, and the cartridge unloaded. If a hardware fault

interrupts the SAVE, the busy light flashes (if possible) and the tape is stopped. At the completion of the first tape on the 7914 the busy LED is flashed alternately fast and slow to prompt the user for another tape. If end of volume is reached on the disc a file mark is written on the tape and the tape unload sequence is invoked. If the end of the tape (second tape on the 7914) is encountered first the tape unload sequence only is invoked.

\*RESTORE button - This button is also mounted behind the front panel. It functions the same as SAVE but in reverse (tape to disc). Additionally, RESTORE will stop at file marks and blank blocks. Data and hardware faults are handled analogously to SAVE. The 7914 uses the disc address written on the tape for restoration of both tapes.

BUSY indicator - This LED visible from the front panel indicates that a tape operation is in progress when it is lit. During Save and RES sequences this LED is flickered (8/sec.) while waiting for a button, flashed (1/sec) to indicate a fault, and strobed while flickering to indicate that the 7914 is waiting for a second tape.

WRITE PROTECT indicator - when lit this LED, visible from the front panel, indicates that the cartridge is write protected, and that writes to the tape are not allowed. When write protected, the controller cannot modify the system blocks on a tape so sparing, logging of errors, and tallying of total tape use are disabled.

\*These switches are only used on the shared controller versions of Linus.

### 3.2 Cartridge Loading

The loading sequence is initiated whenever a cartridge is plugged in. The tape is moved to the beginning of tape (BOT), the drive tests itself, then calibrates itself to the particular cartridge in the mechanism. This process takes about two minutes for the "L", or about 55 seconds for the "S" cartridge. At the conclusion of this, it is ready to be accessed by the host.

### 3.3 Cartridge Unloading

The unload sequence (described under the UNLOAD button) has to be executed to remove a cartridge from the drive. It can take up to 80 seconds to rewind the tape to EOT for a "L" cartridge, and 20 seconds for an "S" cartridge.

## 4.0 HOST SYSTEM INTERACTION

### 4.1 Functional Description of Command Sequence

To access the tape, a starting block address, burst size and transfer length must be specified. The physical block size is 1024 bytes, and up to four blocks may be buffered by the controller. The starting address is converted internally to a two dimensional key and track address. A locate and read (or write or verify) command initiates the shortest seek to this key and track location. Seek times are 195 msec per track, about 100 msec accel. and decel. times per seek, plus about 20 msec per key traversed. (ie, to go from block 0 to block 4087 takes 80 seconds. To go from Block 0 to Block 8177 takes about 390 msec.). Seeks can take relatively long periods of time, especially if done without regard to the inherent personality of the tape.

Once the target address is reached, writes, reads and verifies are commenced in much the same manner as with discs. If the transaction overflows a track, auto seeks to the next track are done. Retries are also accomodated during reads and verifies.

Verifies can be done at the conclusion of a write transfer. They should not in general be done on a block by block basis, however, as this results in extremely poor performance, and excessive wear on the drive and cartridge.

### 4.2 Linus File Mark Capability

3M sells a controller for a stand alone Linus configuration which implements file mark capability. The 3M controller uses data block headers to record file marks which are requested by the user via a write file mark comand. In order to allow HP controllers to be compatible with 3M tapes, an equivalent function will be implemented on Linus.

The rules for handling file marks involve a new write file mark command, and modification to locate and read to allow termination of a transfer when a file mark is encountered. File marks are not involved in tape positioning prior to a read, write, or verify command. No skip file mark commands are implemented.

Whenever a file mark is encountered during a read, the data transfer is terminated. The End of File indication in the status request message becomes asserted when the host attempts to access the byte after the last byte in a file. Whenever EOF is encountered, a message terminator byte, "01", is sent to the host. The contents of this byte are not part of the requested data.

Unless a file mark is encountered, the length of the execution message is determined by the current value of the length parameter. If a data transfer ends on the last byte of a file, the EOF indication is saved until the host attempts to read the next byte. In this case the message terminator byte is sent in a single byte execution message. If an EOF is encountered in

the middle of a data transfer, that transfer is terminated and EOF occurs immediately. In this case the message terminator byte is appended to the data message. The length of the execution message is never longer than the length parameter specification. Burst mode operation is independent of the rules involving file marks.

#### 4.3 Defect Handling

Two types of media errors are possible with the Linus tape : an unreadable key for a given block and an unrecoverable data error. Bad keys are relatively common (0 - 10 per "L" cartridge) and can be detected on a virgin tape during any operation. With the internal error correction, uncorrectable data errors are rather infrequent (<1 in 10 "L" cartridges) but can only be detected during read and verify operations. To ease the handling of errors the Linus controller sets aside one block in 512 as a spare (total of 32 blocks on the "S" tape and 128 on the "L" tape). These blocks are not in the users addressing space and are only used to replace bad blocks.

Two methods of using the spare blocks are provided by the Linus controller. In the first method, referred to here as skip sparing, a bad block is spared by adding its address to a table of blocks to be skipped and removing the next available spare from the same table. This type of sparing results in minimal latency but the addresses of the blocks between the bad block and its spare are altered. In the second method, jump sparing, the bad block is directly replaced by the closest, by seek time, available spare. Any future reference to the block generates a seek to its spare requiring on the average a latency of 2.5 seconds to get to it and nearly the same time to resume with the next sequential block. This method, however, does not alter the address of other blocks and is safe for use when needed data resides beyond the bad block. In the special case when a skip spare is asked for but no spare blocks are available beyond the bad block a jump spare is substituted for the skip. There is a limit of 32 jump spares for any Linus tape.

Sparing is invoked by the host sending a Spare Block command, with a parameter byte indicating the method, or by the host writing to a known bad block after enabling Auto Sparing via the Set Device Specific Options command. Each spare operation requires the controller to rewrite a sparing information block near BOT, this will be done at the end of any transaction causing the spare but before status is returned.

#### 4.4 Commands Supported by Linus

##### 4.4.1 Real Time Commands

###### Locate and Write

Linus will execute the locate and write command in the same manner as the disc. Complete disc operations may be executed in parallel with the 'locate' portion of a Linus transaction, as described under parallel operations (to be

implemented at a later date). Also note the sparing rules in section 4.4.

After completion of a write operation, the tape will automatically be repositioned to a point in front of the next block to be written. This way the seek required in preparation for a sequential write will take place in parallel with the receiving and decoding of the next command.

#### Locate and Read

The read and cold load read operations are executed in the same manner as the disc, with the following exceptions.

File marks will be treated as mentioned earlier.

The user will be able to configure the drive to take advantage of character count capability during reads. This is necessary for 3M controller compatibility. If character counts are ignored, partial blocks will be padded with 0's to their full length during a read operation. If the character count function is enabled, the controller will skip the unfilled part of a partially written block, and proceed with the next block. The default is to ignore this powerful capability.

Upon completion of a read command, the controller will instruct the tape drive to read the next block, in case a sequential read command should follow. This is done in parallel with other controller activity, and does not impact the controller's interaction with the host.

Attempting to read a block which has never been written will terminate the transfer with a No Data Found media error.

#### Verify Command

The verify command on Linus operates like the disc verify command. The host should consider the following characteristics when using this command:

- 1 - The command will verify n logical blocks where n equals the transfer length divided by 1024. There is no adjustment for empty, partial or filemark blocks within this range.
- 2 - Verify will terminate immediately on an unrecoverable data error unless the bad block has been spared.
- 3 - Verify will not terminate on a file mark.

## Copy Data

Copy data is described in the CS 80 (Feb 1981) reference manual. The following comments should be noted when using this command:

- 1 - When Linus is the source unit and an End of File (EOF) is encountered the Copy transaction stops and EOF status is reported.
- 2 - If an unwritten block is encountered while Linus is the source device, the transfer will be terminated with a No Data Found media error.
- 3 - The full 7914 disc image cannot be backed up on a single tape cartridge.
- 4 - The command is not executable on the dual controller version of the 7911, 7912 and 7914.

### 4.4.2 Complimentary Commands

The following complimentary commands are implemented:

```
SET Block Address
SET Block Displacement
SET Length
SET Burst
SET Retry Time
SET Status Mask
SET Device Specific Options
```

All of these commands function in exactly the same manner as described in the CS 80 reference manual with the following extension:

Set Device Specific Options                    C,R

```
-----
:00111000: :00000ASC:
-----
```

```
C = 0: Disable character count capability.
C = 1: Enable character count capability.

A = 0: Auto sparing disabled.
A = 1: Auto sparing enabled.

S = 0: Auto spare invokes Jump Sparing.
S = 1: Auto spare invokes Skip Sparing.
```

The C bit in this complementary command configures the controller to utilize the tape drive's character count capability. This does not impact write operations. Character counts are always written in block headers. If this function is disabled for a read operation, partially filled blocks will be padded with 0's during a read. If the function is enabled, only the filled portions of blocks will be sent to the host. The

character count will be used to cause Linus to skip to the next block when all of the user data in a given block has been sent. No dummy data is synthesized.

When the A bit is set the Linus controller has permission to spare any bad blocks encountered while writing user blocks. Bad blocks are limited to those with an unreadable key and any blocks previously put in the error log. If the S bit is set Auto sparing will use the skip rather than the jump method of sparing. The skip method is recommended when the user doesn't care about any data which may reside beyond the current target address.

#### 4.4.3 General Commands

##### Describe

The format of the describe message does not require modification for Linus. The number of user blocks given by describe reflect the size of the current cartridge loaded in the drive. It will be zero when no cartridge is loaded, 16352 for a "S" cartridge with spares, and 65408 for a "L" cartridge with spares. A cartridge which has never been written to or has been initialized to have no spares would be describes as having 16384 or 65536 blocks.

##### Release, Release Denied

The Linus controller requests release for the following events:

- 1 - A tape cartridge finishes auto load. Release is required to obtain controller resources to execute Linus Self Test and Table initialization routines.
- 2 - The operator pushes the Store, Restore, Unload or Selftest buttons. Release is of course required before any of these actions can be taken as controller resources are needed and the host may have file or directory maintenance to perform before allowing such drastic actions.
- 3 - The run time error table in the controller fills. The controller then requests release for maintenance so it can post the new errors.

The host grants or denies release with the appropriate command. If a controller defined timeout (~2 seconds) elapses before the host acts on a request for release the Linus controller grants itself release and proceeds as it desires.

**Write File Mark**

C,R

```

-----
:01001001:
-----

```

This Real Time command causes a file mark to be written at the current position of the tape. The actual file mark indication appears as a bit in the header of a data block (1K). No data is stored in the block containing the file mark. This is a real time command.

**Spare Block Command**

C,R

```

-----
:00000110: :000000s1:
-----

```

s = 0: Skip spare.  
s = 1: Jump spare.

This command causes the controller to spare the current target block by the method indicated in the option byte. This action includes updating the sparing table near BOT thus tape motion is involved and this command may take 90 seconds to execute. This command does not retain data in the spare block.

**Initialize Tape**

C,R

```

-----
:00110111: :00000CWZ: :00000000:
-----

```

Z = 0: Rewrite sparing table with no jump spares.  
Z = 1: Reset sparing table to initial spares.

W = 0: Initial spares are every 512th block plus a track offset.  
W = 1: Initial spares are no spares.

C = 0: Runs certification if necessary.  
C = 1: Does not force a certify test.

With all the option bits (Z,W,C) cleared, this command certifies a tape (if it hasn't already been certified), sets up a spare table on the tape (if one hasn't already been set up), and spares out any defective blocks discovered during the certify test. If the Z bit is cleared, but a spare table does exist on the tape, all jump spares are converted to skip spares. If the Z bit is set, the spare table is cleared and rewritten to an initial state as per bit W. In this case if N is zero there will be an unused spare for each 512 blocks and these available spares are staggered from block to block. No spares are used; thus all bad blocks are to be rediscovered. For W = 1 a null spare table is written allocating no spares on the tape. The certify test is also skipped. Sparing becomes the

responsibility of the user as the controller is left with no resources for it. This option should be used only for compatibility with non-HP systems. When the C bit is set the certify test is not done. The media will return an initialized media status when loaded into a drive in this condition, even though the spare table is initialized. The command takes 15-60 minutes to execute on an uncertified tape.

The following table defines the controller activity for all possible combinations of previous tape state and format parameters.

Tape State	C	0	0	0	0	1	1	1	1
	W	0	0	1	1	0	0	1	1
Z	0	1	0	1	0	1	0	1	1
No Spare Table									
Not Certified	C	C	X	X	B	B	X	X	
Spare Table									
Not Certified	C	C	X	X	O	B	X	X	
Spare Table									
Certified	O	C	X	X	O	B	X	X	

C - Certify the tape  
 O - Optimize the spare table  
 B - Build a new spare table  
 X - Clear the spare table

Unload

C,R

-----  
 :01001010:  
 -----

When the Linus controller receives an unload tape command, the error log is written, if necessary, and the tape is unloaded. This action may require rewriting a block at BOT as well as EOT so prepare for a possible three minute wait until status is returned. The controller supports parallel disc operations during this operation.

#### 4.5 Diagnostic Commands

The three following Diagnostic Commands are supported:

Request Status  
 Initiate Diagnostic  
 Execute Utility

With these categories of commands, it is possible to initiate

tests and gather results at both the interface and functional levels. In addition, the utilities provide a mechanism for retrieval of logs, diagnostic results, and characterization of the head/media interface. The Error Rate Test Log and the Error Log will be stored in the system blocks found near the BOT (Between KEYS 9&10 and 10&11 respectively). As part of the unload sequence the "USE" Log will be updated. The "USE" log resides between physical 4107 & 4108 on the "L" tape and 1035 & 1036 on the "S" tape. A short discussion of the purpose and content of the above mentioned logs are included in the explanations of the utilities which pertain to them.

#### 4.5.1 Request Status

```
-----
:00001101:
-----
```

The Request Status command operates exactly as described in the disk ERS. In summary, the Request Status command returns a 20 byte status report (in an execution message) indicating the status of the last transaction. The status request consists of a 2-byte identification field, an 8-byte error classification field, and 10 bytes of additional error dependent parameters. A summary of the format of the status report is found in the CS-80 Instruction Set Manual. A more detailed discussion of this command is provided in the CS-80 ERS.

#### 4.5.2 Initiate Diagnostic

```
-----      -----      -----
:00110011:   : LOOP PARAMETER :   :SECTION#:
-----      -----      -----
      OPCODE           2 bytes           1 byte
```

The Initiate Diagnostic command will perform the phase of the internal diagnostic which is indicated by the diagnostic section number. The internal diagnostic is divided into several sections, the following two of which pertain to Linus. These tests are invoked at Power-On, and may be run individually through the Initiate Diagnostic command. TCP two Linus related diagnostic sections are specified as follows.

- ```
--
: 8: - TIB MICRO Diagnostic:
--   Functional testing of the Tape Interface Board,
      including special diagnostic mode testing.

--
:11: (A)-Initiate AUTO-LOAD Sequence:
--   Performs the OEM Self Test, tensioning, and Head
      alignment functions with the cartridge in place.

      (B)-DMA/TIB MACRO Diagnostic:
          Tests all control lines and the serial data
          path between the DMA and Tape Interface Boards

      (C)-TIB/LINUS MACRO Diagnostic:
          Tests control and data paths between the Tape
```

Interface Board and the LINUS mechanism... includes a short R/W and command/response tests. The R/W test will seek to a diagnostic key near BOT and attempt to write a test pattern on 2 of 16 tracks. Tape length is also retrieved from the "manufacturer's block".

#### 4.5.3 Execute Utility

```

-----
:001100XX: :u OPCODE: : : : :
-----
1 BYTE 0-4 PARAMETER BYTES

```

XX = Execution Message Qualifier  
 00 -> No Execution Message  
 01 -> Receive Execution Message  
 10 -> Send Execution Message

The Execute Utility Command is followed by a micro-opcode which defines a variety of disc and tape utilities. Depending on the utility selected, a pre-defined number of parameter bytes may be expected to follow the utility number. Although, the interface to this family of utilities tends to be device independent, the internal algorithms do depend on the specific device. The following Linus utilities present a consistent disc/tape interface.

#### Pattern Error Rate Test:

```

-----
:11001000: : LOOP : : TYPE : :TST AREA: :DATA SRC:
-----
u OPCODE 1 byte 1 byte 1 byte 1 byte

```

#### Parameter Definition:

LOOP -> 1-255 = Loop Count  
 0 = NOT Allowed

TYPE -> 00 = Read only ERT  
 01 = Write/Read ERT  
 10 = Certification  
 (Sparing KEY errors and  
 blks with >=2 frame errors)

TST AREA -> 00 = Use Current Address &  
 Current Length.(Logical)  
 01 = Specified & Next Track  
 (Phy)-(XXXX0001),  
 Where XXXX=Specified Track  
 10 = Entire Tape(Phy)

DATA SRC -> Defines data source  
 00 = Use Internal Pattern Tbl.  
 01 = Use User defined pattern.  
 10 = Use Random Data

This error rate test and certification utility is provided to aid in characterizing the head/media interface. Write/Read and certification tests are destructive to stored data. Tape certification includes a write/read pass on all physical blocks while sparing all blocks with a key error or two or more frames in error. Therefore, the test area is ignored when certification is selected. The test area may be either logical, using the current address and length, or physical, specifying a pair of tracks or the entire tape. The data source may be from a pseudo-random data generator, an internal Pattern Table, or user defined by the "RECEIVE USER PATTERN" utility described later. The results of this may be recovered using the "Read ERT Log Utility". \*Permanent correctable and \*uncorrectable data errors are logged, including their logical addresses where possible. Since reserved spare blocks may also exhibit errors, the appropriate error counts will be incremented, but no records will be returned for those blocks because there is no corresponding logical address. The sum of the error counts (permanent and key), may therefore, be larger than the number of records returned. \*Transient data errors and key errors are counted. Since the error log is written to a system block on the tape, rather than returned directly to the user, the tape must not be write protected even for a Read Only ERT. The parameters bytes for this test are position specific. Therefore, a parameter bounds error will not be set when undefined bit positions in the byte are used.

Read Error Rate Test Log:

```

-----
:11000110: :00000000:
-----
u OPCODE      1 byte

```

The ERT Data Error Log is the standard log for all errors found while executing the Pattern Error Rate Test. The single byte "null" parameter indicates that the results will cover all blocks specified during the ERT run. The log is composed of a header which contains the number of blocks accessed and error counts, and records which contain relevant addresses and error qualifiers. This log contains data and key errors, and is kept separate from the "Run Time" Error log so cumulative errors are not a concern during testing. Records are kept for permanent errors (correctable and uncorrectable) and key errors.

Log Format:

LOG HEADER

```

# of records _____ 1 byte
# of blocks accessed _____ 4 bytes
* # of permanent errors _____ 2 bytes
* # of transient errors _____ 2 bytes
* # of uncorrectable errors _____ 1 byte

```

# of key errors \_\_\_\_\_ 1 byte

LOG RECORD

ADDRESS PORTION:

Logical Block address \_\_\_\_\_ 2 bytes

ERROR PORTION:

ERROR BYTE(See definition below) \_\_ 1 byte

ERROR BYTE - Bit 0 = 0 frame 1 NO CRC error  
                  = 1 frame 1 CRC error  
          Bit 1 = 0 frame 2 NO CRC error  
                  = 1 frame 2 CRC error  
          Bit 2 = 0 frame 3 NO CRC error  
                  = 1 frame 3 CRC error  
          Bit 3 = 0 frame 4 NO CRC error  
                  = 1 frame 4 CRC error  
          Bit 4 = 0 ECC 5 - NO CRC error  
                  = 1 ECC 5 - CRC error  
          Bit 5 = 0 ECC 6 - NO CRC error  
                  = 1 ECC 6 - CRC error  
          Bit 6 = 0 correctable error  
                  = 1 uncorrectable error  
          Bit 7 = 0 Frame error  
                  = 1 Key error

\* A PERMANENT ERROR is a block with one or more defects in the data area. Most defects are correctable. A block with 2+ defects in non-adjacent frames is an UNCORRECTABLE ERROR. A TRANSIENT ERROR is any frame error that disappears in less than three retries.

As a suggested output format for the ERROR byte, display only those categories of errors which are "active"(Bit position = 1).

Read Error Log

```
-----
:11000101: :00000000:
-----
u OPCODE   1 byte
```

This Run Time Log provides a history of performance for a given tape. The "null" parameter byte indicates the results are over all tracks and all blocks. The Clear Log Utility (Described later) will clear this log of all records and counts. The only records kept are for uncorrectable and key errors. This log is stored near BOT, between KEYS 10 & 11, in the following format. Certification is zero for uncertified tapes; otherwise greater than zero.

LOG FORMAT:

LOG HEADER

# of records \_\_\_\_\_ 1 byte

# of uncorrectable errors \_\_\_\_\_ 1 byte  
# of key errors \_\_\_\_\_ 1 byte  
Type of certification\* \_\_\_\_\_ 1 byte

LOG RECORD

THE LOG RECORD IS OF THE SAME FORMAT AS THAT  
KEPT FOR THE ERT LOG.

\*Possible values for the "Type of certification" are:

0 = Not certified  
1 = 3M certified  
2 = HP factory certified  
4 = Certified on-line

Read "USE" Log

-----  
:11000111:  
-----

The "USE" log provides a history of the total tape activity. It contains two entries, auto-load count and number of blocks accessed. This log resides at physical block 1035 on the "S" tape and 4107 on the "L" tape. The log is only updated during the unload sequence and will therefore remain unchanged during any single session where a tape is inserted. To preserve an accurate record of tape activity, this log may not be cleared. The "USE" log has the following format:

LOG RECORD

Count of Auto-load sequences  
performed on this tape \_\_\_\_\_ 2 bytes  
Count of blocks accessed \_\_\_\_\_ 4 bytes

Read Drive Tables

-----                   -----  
:11000100:               :Table Number:  
-----                   -----

This command will return RAM values stored in the drive table specified by the parameter value. This information can be used to achieve a better understanding of the drive state. The possible parameter byte values and the associated tables are:

Table Number = 10 Manufacturer's Block table  
              = 11 Spare Block Table  
              = 12 Copy Start Address



most significant byte sent first.

#### Receive User Pattern

```
-----  
:11010001:  
-----
```

This utility will accept 64 bytes of data (passed from the host) and write them to the controller RAM area for later use with the error rate test.

#### Clear Logs

```
-----      -----  
:11001101:    : LOG CODE :  
-----      -----
```

This utility will clear logs indicated by the parameter byte.

Log Code = 0 Clear the Error Log and ERT log.  
          = 1 Clear the ERT log only.

#### Read Revision Numbers

```
-----  
:11000011:  
-----
```

This utility reads the firmware revision numbers for disc/tape/and controller firmware. The first byte returned is equal to the number of bytes which will follow.

#### Preset Drive

```
-----  
:11001110:  
-----
```

This command will force all periodic upkeep required by the device controller. This includes updating logs containing run time data errors or drive faults. This command should be issued periodically by the host, so it has control over the timing of periodic logging.

#### Read Error Summary

```
-----      -----  
:11000001: :Recent or Previous:  
-----      -----  
u OPCODE   1 byte
```

#### Parameter:

0 = Most resent results (1 summary)  
1 = Last 4 summaries with errors.  
(Does not include current summary.)

This utility returns 1 or 4 (parameter dependent) 64 byte "bit maps" which represent the results of all tests executed. Where the Linus Drive is the only unit besides the controller, the "0" parameter is the only valid parameter. When the controller is common to disc and tape, the previous 4 error summaries may be cleared with the CLEAR LOGS (ALL) command issued to the disc. Since the bit positions correspond to actual errors, their definition is now in progress. One test's successful completion may depend on the results of another test, and therefore, the host may intelligently combine a list of test dependencies with these results to perform fault isolation. This utility is provided so field testing tools may intelligently adapt to field experience.

#### 4.6 Power-On Sequence

Four separate diagnostic phases are entered during Linus System Power-On.

They are:

- (1) Controller (Including the Tape Interface board) power on diagnostics.
- (2) Linus Auto-Load Sequence.
- (3) Tape Interface Board to Linus interface tests (Macro-diagnostics)
- (4) System blocks are loaded into controller RAM.

The Linus auto-load sequence (2), begins when a tape is detected in the drive. If a tape is present at power-on, the auto-load sequence and the controller power-on diagnostics execute in parallel with the disc coming on-line before the more lengthy auto-load is complete. In the normal situation, the controller will be "up" when the tape is inserted. For this case, phase (2), (3), and (4) summarize the Linus Tests before coming on line.

In the first phase, the controller and disc (if present) "board and interface" tests are executed. This includes controller to Tape Interface Board Data and Control Path Testing. This testing occurs whether or not a tape is present in the drive. In the second phase, the Linus drive performs tape tensioning and head alignment functions, as well as checking internal RAM and registers for proper operation. This phase is initiated when a tape is inserted or found (during system power-on) in the drive. The functions are all performed with the Linus's internal micro-processor.

In the third phase, the controller attempts to communicate with the cartridge using a short Read/Write test. This test is non-destructive, using blocks invisible to the user. In the last phase, the controller reads system information from the Linus cartridge into the controller RAM. This information includes the manufacture's block and Spare tables.



5.0 PERFORMANCE SPECIFICATIONS

5.1 DATA CAPACITY

Formatted Data Capacity: 67.0 Mbytes Per "L" Cartridge  
16.7 Mbytes Per "S" Cartridge

5.2 Data Transfer Rate

Average Data Transfer Rate: 35.0K bytes/sec. (2.1Mbyte/min)  
Burst Transfer Rate: ~ 900K bytes/sec.

5.3 Access Time

Total access time is the sum of the following factors:

Track to Track Seek 195 msec/track  
Key to Key Search 19.8 msec/key (1Kbyte/key)

(Worst case access is 83 sec.)

5.4 Tape Speed

Read/Write/Verify 60 ips  
Search (Key) 90 ips

5.5 Recording Parameters

Encoding Technique MFM  
Bit Density 10000 bpi  
Number of Tracks 16  
\*Number of User Blocks 65408 for "L", 16352 for "S"  
Data/Block 1024 bytes

5.6 Error Rate

< 2 errors in 10E10 bits transfered, unrecoverable e.r.

\* For initialized tape

## 6.0 ENVIRONMENTAL

The Linus tape subsystem is designed to meet the Class B requirements of the HP Corporate Environmental Specification A-6950-5344-1. Exceptions to this specification are noted.

### 6.1 Temperature

OPERATING - five degrees centigrade to forty five degrees centigrade.

NON-OPERATING - negative forty degrees centigrade to sixty five degrees centigrade.

### 6.2 Humidity

OPERATING - 8% to 80% non-condensing.

NON-OPERATING - 5% to 95% non-condensing.

### 6.3 Vibration

Operating and non-operating per DMD dwg A-5955-3439-1

### 6.4 Shock

11 msec, 1/2 sine wave, 30 g's non-operating.

### 6.5 Altitude

The unit will withstand altitudes from 300 meters below sea level to 4600 meters above sea level, operating; and non-operating to 15000 meters.

7.0 RELIABILITY

7.1 Failure Rate

The failure rate goal is 6.6% /1000 hours of power on time, and tape moving at 17% duty cycle.

7.2 Repair Time

The mean time to repair (MTTR) goal is <1 hour excluding travel time.

7.3 Average Use Estimates

Power on - 50% duty cycle

Tape in motion - 2 hours/day average

7.4 Service

No field preventative maintenance is required. The head and capstan can be cleaned by the customer using a chemical head cleaner. (Every week)

7.5 Cartridge Life

Average life of the cartridge is 2500 end to end cycles of the tape. One cycle is moving from EOT to BOT to EOT.

8.0 PHYSICAL SPECIFICATIONS

8.1 Size

Drive Module - 117mm. high x 178mm. wide x 210mm. deep

Tape Interface Bd. - 178mm. x 305mm. (1/2 Amigo size)

8.2 Input Power

5v - 3.6 amps

12v - .75 amps (4.0 amp surge)

## 9.0 Examples of Linus media defect handling

The following examples show how the host should interact with the Linus controller to minimize error handling problems. For these examples the tape has a bad key for block 5, a bad data area in block 10; blocks 7 and 15 are allocated as spares. The host is working with a 18 kbyte file starting at block 0.

```

L 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
+---+
| | | | | * | |ss| | |**| | | | |ss| | | | | |
+---+
P 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

```

\* - defect over key  
 \*\* - defect in block  
 ss - available spare block  
 xx - block deleted by sparing

NOTE: P-## indicates physical block ##. Physical addresses are not accessible to the host.

L-## indicates logical block ##. All host addresses are Logical addresses.

Case 1 - Writing the file on virgin tape.

The host issues a write at 0 for 18 kbytes.

Linus writes blocks L-0 to L-4.  
 The Linus controller adds block 5 to the Error Log.  
 Linus accepts and sinks blocks L-5 to L-17.  
 The returned status indicates that block 5 is of no use.

The host should issue a command to skip spare block L-5.  
 The host then reissues a write at 0 for 18 kbytes.

Linus writes blocks L-0 to L-4.  
 Linus skips block P-5.  
 Linus writes block L-5 (in block P-6).  
 Linus writes blocks L-6 to L-13. (P-7 becomes a data block)  
 Linus skips block P-15.  
 Linus writes blocks L-14 to L-17. (in blocks P-16 to P-20)  
 Linus returns normal status as the error at block 10 cannot be detected yet.

The tape now contains...

```

L 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
+---+
| | | | | *xx| | | | |**| | | | |ss| | | | | |
+---+
P 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

```

Case 2 - Reading the above file.

The host issues a read at 0 for 18 kbytes.

Linus returns data from blocks L-0 to L-4.

Linus skips block P-5.

Linus reads data from blocks L-5 to L-8.

The Linus controller enters block P-10 in the Error Log.

Linus returns it's best guess for L-9, and reads L-10 to L-13.

Linus skips block P-15.

Linus reads blocks L-14 to L-17 (from blocks P-16 to P-20).

The returned status indicates unrecoverable data in block L-9.

Case 3 - Rewriting the above file.

The host reads the error log and is told about block L-9.

The host issues a Spare Block L-9 command, with jump option.

Linus logs P-10 in its Jump block table with a link to P-15.

The host issues a write at 0 for 18 kbytes.

Linus writes blocks L-0 to L-4.

Linus skips to block P-6.

Linus writes blocks L-5 to L-8.

Linus seeks to block P-15.

Linus writes block L-9.

Linus seeks to block P-11.

Linus writes blocks L-10 to L-13.

Linus skips block P-15.

Linus writes blocks L-14 to L-17.

Linus returns a normal completion.

The tape now contains...

```

L 0 1 2 3 4 5 6 7 8 10 11 12 13 9 14 15 16 17 18
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| | | | | *xx| | | | xx| | | | | | | | | | | |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
P 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

```

Case 4 - Reformatting the Tape

The host issues a Initialize Tape command, saving spares.

Linus logs blocks P-5 and P-10 in the skip table.

Linus clears the Jump table.

Linus returns good completion.

The host issues a write to 0 for 18 kbytes.

Linus writes blocks L-0 to L-4.

Linus skips block P-5.

Linus writes blocks L-5 to L-8 (in blocks P-6 to P-9).

Linus skips blocks P-10.

Linus writes blocks L-9 to L-17 (in blocks P-17 to P-19).

Linus reports good completion.



## 10.0 LOGICAL INTERCHANGE COMPATIBILITY

Linus is compatible with the HP Logical Mass Memory Format Standard (LIF) as revised in January '81. The following notes apply to Linus when used for LIF:

1. Sector (physical block) = 1024 bytes
2. The number of usable sectors per take is readable from the DESCRIBE command (16352 for initialized short tape, 65408 for initialized long tape).
3. For writing, use skip spacing, SPARE BLOCK command S = 0. (Auto skip spacing is recommended, device specific options A = 1, S = 1).
4. On used tapes convert jump spares to skip spares, INITIALIZE TAPE, E = 1.
5. It is also recommended that a file mark be written at the third sector to prevent the disc from accidentally being "image" restored by a LIF tape.