```
                        .REM @
```

IDENTIFICATION

PRODUCT CODE:	AC - FG13A MC
PRODUCT NAME:	CZTU1A0 T81 DATA RELIAB TEST
PRODUCT DATE:	SEPTEMBER 1985
MAINTAINER:	TAPE OPTICAL DIAGNOSTIC ENGINEERING
AUTHOR:	BRIAN T. LEBLANC

TABLE OF CONTENTS

1 GENERAL INFORMATION

1.1 Program Abstract

The TU81 PDP11 Data Reliability program will exercise the TU81 and establish the performance quality of each unit through the accumulation of statistics. Predetermined sequences of operations will permit read and write compatibility (Media Interchange testing) and data reliability testing. This program will be designed to run in a PDP11 XXDP+ environment.

The Data Reliability program will detect functional faults, but will not provide diagnostic isolation to the field replaceable unit.

The PDP11 TU81 Data Reliability program is intended for the following users:

1. Quality and user audit functions,
2. F A & T at our various facilities,
3. Field service personnel,
4. DEC customers who choose to provide their own maintenance.

Program uses include but are not limited to the following:

1. Determination of a unit's specific performance (error rate)
2. Fault detection,
3. Repair verification,
4. Installation verification,
5. Preventive maintenance software tool.

This program will exercise up to 4 TU81's in a round-robin manner. It will require 28KW of memory. One default pass will be when a tape cartridge (600') has been started at the beginning of tape (BOT) marker and has passed all available tape to the end of tape (EOT) marker over the tape head, twice. One End Of Pass (EOP) will require approximately 1 hour and 10 minutes for each unit under test.

1.2 Runtime Environment Requirements

Run time environment requirements include:

1. XXDP+ Diagnostic Supervisor
2. PDP11 family CPU,
3. 28KW of memory,

4. an XXDP Load Device.

5. Console Terminal.

6. 1 to 4 TU81 drives with controllers.

7. 1 scratch tape / TU81

1.3 Related Documents And Standards

The TU81 Data Reliability program will run under the XXDP+ operating system, and will be Supervisor compatible. The program, with the supervisor will run on all PDP11 processors.

This program will conform to the following documents:

1. EL-ENDIA-11 "PDP11 Diagnostic Design Guide",

2. PDP Diagnostic Quality Assurance Checklist,

3. Software Development Policies And Procedures Manual,

4. DEC Std 100,

5. UNIBUS/Q-bus Storage Systems Port Spec Version 2.1

6. Magnetic Tape Mass Storage Control Protocol Spec Version 1.6

7. Mass Storage Control Protocol Spec Version 1.2

1.4 Pass/Fail Criteria

A unit under test will not pass the data reliability mode of testing if any of the following error conditions have occurred during the test cycle:

1. Any irrecoverable write errors detected as documented in the TU81 product specification,

2. Any irrecoverable read errors detected as documented in the TU81 product specification,

3. Irrecoverable hardware errors have occurred,

4. CRC recoverable read errors which exceed TBD errors in 10 to the 11th bits read

5. ECC recoverable read errors which exceed TBD errors in 10 to the 11th bits read

If less than the required data has been transferred, the confidence that the unit has met the error rate is diminished. That is to say if the program is run in a quick verify mode, the unit may be accepted as error free but only with a low degree of confidence.

1.5 Data Compare Function

The time required to perform 100% software data comparisons is entirely prohibitive for streaming tape drives. This problem is further exacerbated by the asynchronous nature of command execution under TMSCP and program size limitations which dictate the allocation of a single read data buffer.

To minimize the impact of all this, tests 2 and 3 (the only tests which will perform software data compares) will do software data compares on every 4th record. To avoid the problem of performing data compares on a dynamic read buffer, 3 records will be read from tape using the Access command.

1.6 Restrictions

This program is not intended for use as an isolation tool to detect a fault to the single Field Replaceable Unit (FRU). As such, it will not contain scope loops for that purpose. The parameter selection process, discussed later in this document, is meant to be used only for functional fault detection and unit isolation.

2 OPERATING INSTRUCTIONS

2.1 User Dialogue

The following user dialogue will be provided by the diagnostic to allow the operator to establish certain operational parameters of the program.

2.2 Hardware Questions

This set of questions must be answered by operator when the program is first started.

```
CHANGE HARDWARE (L) ?  no oefault

NUMBER OF UNITS (D) ?

UNIT x                                  x = Number of unit the p-table is
TKIP ADDRESS (O) 774500 ?                   being built for.
T/MSCP UNIT NUMBER (O) 0 ?
```

Unit specific prompting will continue for a maximum of 4 times, depending on the users response to the "NUMBER OF UNITS" question.

2.2.1 Definition Of Hardware Questions -

CHANGE HARDWARE - If you want to change the hardware p-table to be used in the testing this question must be answered yes. This question must be answered with a yes on the initial start of the program.

NUMBER OF UNITS - Number of units to test in decimal.

TKIP ADDRESS - The base address for this unit.

T/MSCP UNIT NUMBER - The unit number of the controller board as specified by MSCP.

2.3 Software Questions

Answering of the software questions is always optional. Default values for a specific question can be obtained simply by typing a <CR>.

```
CHANGE SW (L) ?    no default

ENABLE TIME OF DAY CLOCK (L) N ?
 INPUT HOUR IN 24 HOUR FORMAT (OMIT LEADING ZERO) (D)  0 ?
 INPUT MINUTES (OMIT LEADING ZERO) (D)  0 ?
CHANGE CONTROLLER PARAMETERS (L) N ?
 ENABLE CONTROLLER ERROR CORRECTION (L) Y ?
 ENABLE CONTROLLER ERROR RECOVERY (L) Y ?
 ENABLE PAD BLOCKING (L) Y ?
CHANGE PRINTING PARAMETERS (L) N ?
 ENABLE SOFT ERROR REPORT PRINTING (L) N ?
```

```
296                          ENABLE READ SOFT ERRORS ONLY (L) Y ?
297                         CLEAR MEDIA TABLE ON EVERY PASS (L) N ?
298                         ENABLE PRINTING OF MEDIA DEFECTS TABLE (L) N ?
299                         ENABLE PROGRAM VARIABLES DUMP ON ERROR (L) N ?
300                         ENABLE CLEAR STATS ON FATAL ERROR (L) N ?
301                        CHANGE TEST PARAMETERS (L) N ?
302                         DATA PATTERN (D) 0 ?
303                         RUN TEST 3 ONLY (L) Y ?
304                         ENABLE DATA COMPARES IN TEST 5 (L) Y ?
305                         ENABLE PRINT READ BUFFER IN TEST 5 (L) N ?
306                         CHANGE COMMAND SEQUENCE (L) N ?
307
308
309
310                    2.3.1  Definition Of Software Questions -
311
312
313                    ENABLE TIME OF DAY CLOCK (L) N ?
314
315                    The default is to not enable the  clock.   This  question  allows  the
316                    operated  to  start  a  program clock to track time on a 24 hour basis
317                    during the running of the  program.   The  clock  will  remain  fairly
318                    accurate  as  long  as  the program is running.  Any time you stop the
319                    program the clock will stop running  It  is  therefore  necessary  to
320                    reset the time whenever the program is started.
321
322
323                    INPUT HOUR IN 24 HOUR FORMAT (OMIT LEADING ZEROS) (D) 0 ?
324
325                    Input the hour in a decimal number leaving off any leading zeros.
326
327
328                    INPUT MINUTES (OMIT LEADING ZEROS) (L) 0 ?
329
330                    Input the minutes in a decimal number leaving off any leading zeros.
331
332
333
334
335                    CHANGE CONTROLLER PARAMETERS (L) N ?
336
337                    The default  answer  (no)  prohibits  the  asking  of  the  controller
338                    parameter questions.  To change the controller parameters type a Y.
339
340                    ENABLE CONTROLLER ERROR CORRECTION (L) Y ?
341
342                    If answered "yes" (default) the program will enable  the  controller's
343                    error correction algorithms for read errors.
344
345
346                    ENABLE CONTROLLER ERROR RECOVERY (L) Y ?
347
348                    If answered "yes" (default) the program will enable  the  controller's
349                    error recovery algorithms for write and read errors.
350
351
352                    ENABLE PAD BLOCKING (L) Y ?
```

If answered "yes" (default) the program will enable the controller s
pad blocking algorithms to assist in streaming

CHANGE PRINTING PARAMETERS (L) N ?

The default answer (no) prohibits the asking of the printing parameter questions. To change the printing parameters type a Y.

ENABLE SOFT ERROR REPORTS (L) N ?

The default answer (no) inhibits the printing, but not the tallying of soft errors as reported by the subsystem. Answering the question "yes" will result in detailed error reports on the terminal for each recoverable data error.

ENABLE READ SOFT ERRORS ONLY (L) N ?

This question will only be asked when the above question is answered no. This question allows the operator to enable print outs on read soft errors only. The default answer is to inhibit all soft error printouts.

CLEAR MEDIA TABLE ON EVERY PASS (L) N ?

The default answer (no) allows the tallying of media defects over multiple passes. By answering the question yes, the operator can then print the table on every pass and see how the defects are affected by passing over the heads.

ENABLE PRINTING OF MEDIA DEFECTS TABLE (L) N ?

The default answer (no) inhibits the printing, but not the tallying of media defects as reported in the soft error reports by the subsystem. If the default answer is used the table may still be printed by giving the PRINT command at the supervisor prompt (DS>) after the termination of the program. Answering the question "yes" will cause the printing of the table after every pass and after a control C (C) is issued.

ENABLE CLEAR STATS ON FATAL ERROR (L) N ?

The default answer (no) allows the accumulation of statistics from pass to pass. An answer of "yes" results in the clearing of a devices statistical matrix following any error that results in the unit's being dropped from the test sequence for the rest of the current pass. This action is intended for use primarily by Springfield volume manufacturing.

ENABLE PROGRAM VARIABLES DUMP ON FATAL ERROR (L) N ?

This question is intended as a program and subsystem debug tool. Answering the question "yes" will cause the program to print out the contents of approximately 1K words of critical memory locations. This is a time consuming process and this question should be defaulted under ordinary circumstances.

CHANGE TEST PARAMETERS (L) N ?

The default answer (no) prohibits the asking of the test parameter questions. To change the test parameters type a Y.

DATA PATTERN (O) 0 ?

This question allows the user to select a data pattern from the table of patterns provided by the program. (See the Data Pattern section below.) The default answer, "0", causes the program to cycle through all the data patterns. Answering the question with a number from 1-5 will causes the program to use that pattern only. A number higher than 5 will cause the question to be repeated.

RUN TEST 3 ONLY (L) Y ?

Answering this question "Y" (default) will automatically cause the program to run test 3 only; i.e., it will no longer be necessary to use the /TES:3 switch to the start command. Please note that this question will effectively override the /TES: switch if the user wishes to run a test other than 3. That is, if the user wants to run test 4 he must specify the /TES:4 switch AND answer this question "N".

ENABLE DATA COMPARES IN TEST 5 (L) N ?

The default answer (no) disallows the data compare function during test 5. This would have to be the case when running with a truly unknown tape. The option (yes) is given to the operator so that when a tape is written in a known manner using this program the operator can then run test 5 using data compares.

ENABLE PRINT READ BUFFER IN TEST 5 (L) N ?

Answering this question "yes" will cause a printout of all data read from tape in test 5. The data will be presented on a record basis. This is a time consuming process, and this question should be defaulted except in special cases.

CHANGE COMMAND SEQUENCE (L) N ?

Answering this question "Y" will cause the program to prompt the user for a sequence of commands to be used in Test 7. (See Test 7 below.) If defaulted, this is the last software question asked.

2.4 Conversation Mode Test Questions

Answering of these question is optional. These questions will not be asked unless the operator has answered the CHANGE COMMAND SEQUENCE question with a yes. A total of seven commands may be entered by the operator.

Test 7 is intended to give the user the ability to create a specific sequence of commands. Note that Test 7 will not support the entire TMSCP command repertoire, it is limited primarily to the tape motion commands. To run Test 7, the user must issue a STA/TES:7 and must answer "Run Test 3 Only" with a N(O). The user must also answer "Change Command Sequence" with a Y (yes). Understand that the program does not check for legality of command sequences issued by the user, the onus is on the user to perform this check.

The following questions will be asked by the program to prompt the user for his input.

CMD/1 (O) 160 ?

The user enters the octal value for the desired command from the list shown below. Please note that the command values are those defined by the diagnostic, not by TMSCP. The default value for the first command is a rewind.

DATA PATTERN (O) 1 ?

The user should enter the octal value of the desired data pattern from the table of patterns shown above. If the command does not use a data pattern, any number entered here is ignored.

PATTERN #	DESCRIPTION
0	ROTATE THROUGH ALL DATA PATTERNS
1	ALL 1'S
2	ALL 0'S
3	WORST-CASE MFM PEAK SHIFT (110)
4	ALTERNATE 1'S AND 0'S
5	RANDOM DATA
6	MM PEAK SHIFT (1110)
7	COMBINATION OF PATTERN 3 AND 5
200	NO DATA PATTERN USED

ITEM COUNT (BYTE, RECORD, OBJECT) (D) 0 ?

The purpose of this field varies with the type of command. For example, for write and read commands, the user may specify the record size, in decimal bytes. If the command is a reposition command, the user may specify the number of records, objects or file marks. There are also two special commands provided which use this value in unique

ways. For a branch command, the user would specify the command number to which (s)he wishes to branch. For the delay command, the value entered here is the relative delay length, with larger numbers producing longer delays. User experimentation may be required to produce desired delay.

ITERATION COUNT (D) 1 ?

This field allows the user to specify how many times the command should be issued before the program issues the next command. The value is entered in decimal.

Additional Commands

This same sequence of four questions will be repeated up to 6 more times, allowing the operator to create a command table with seven unique commands. The only noticeable difference in question format is that each time the command question is asked, its relative position in the Test 7 command table is identified.

2.5 Allowable Commands

The following commands are supported by Test 7. Please remember that the octal values are defined by the program and have no numerical correlation to TMSCP command opcodes. Also note that the diagnostic does not check for legality of the value entered or for valid command sequences. Operator error in either of these cases could result in bizarre program behavior.

Octal	Command	Description
10	RD	Read forward
20	WR	Write
30	CMP	Compare host data
40	ACC	Access
50	SPC	Space records
51	SCR	Space records reverse
60	SKP	Skip tape marks
61	SKR	Skip tape marks reverse
70	SPO	Space objects
71	SPR	Space objects reverse
100	WTM	Write tape mark
160	REW	Rewind
300	BR	Branch - item count specifies destination
310	DLY	Delay - item count specifies relative delay
377	END	End of sequence - necessary if sequence has less than 7 commands

2.6 Supervisor Run Time Flags

This program will support all of the PDP11 Diagnostic Supervisor flags except for those mentioned here.

LOE - Loop on Error - This flag will not be supported by this program.

```
581                     Data  reliability programs do not lend themselves to implementation of
582                     error loops.
583
584                     IDR - Inhibit Drop Units - This flag will not  be  supported  by  this
585                     program due to the devices sequential operation.  If an error of fatal
586                     extent happens on the device there is no way to  continue  running  in
587                     any meaningful way.
```

3 ERROR INFORMATION

3.1 Error Reporting

TKDR provides a variety of information in its error printouts, most of which is self-explanatory. The following information is intended to clarify certain messages and abbreviations used

3.2 Commands

All error printouts will contain a field indicating the command on which the error was detected. Refer to the TMSCP specification for detailed descriptions of these commands. Also, please note that commands currently used by TKDR are indicated by an asterisk.

RD*	read
WRT*	write
CMP	compare host data
ACC*	access
SPC*	space records (position)
SKP*	skip tape marks (position)
SPO*	space objects (position)
WTM*	write tape mark
ERS	erase
ERG	erase gap
AVL*	available
ONL*	online
SUC	set unit characteristics
REW*	rewind (position)
ABO	abort
GCS*	get command status
GUS*	get unit status
SCC*	set controller characteristics

The following two "commands" are used by TKDR for special purposes and are not actually sent as commands to the subsystem:

NUL	null - used by program to while waiting for last responses to real commands
INT	initialize - used by program to invoke the UQ-Port init sequence

3.3 Type Of Error

Each error message includes one line of text intended to describe the type of error detected. There are three distinct sources of information used by the program to generate the text message: the status field of an end packet; an error log packet; and program detected error conditions.

3.4 Status Errors

These messages are derived from the status field of an end packet and correspond directly to the status codes as defined in the TMSCP specification.

Invalid command issued
Command aborted
Unit offline
Unit available error
Unit write protected
Data compare error
Data error
Host buffer access error
Controller error
Drive error
Formatter error
BOT encountered
Tape mark encountered
Data record truncated
Position lost
Serious exception
Logical EOT encountered

3.5 Error Log Packets

Certain messages will be generated as a result of receiving the "diagnostic mode" error log packet.

Retriable Data
Hard CRC
Data Underrun
Data Overrun
ECC Corrected
CRC Error on ECC Block

3.6 Program Detected Error Conditions

In addition to reporting errors detected by the subsystem, TKDR may generate additional error reports based on problems it detects. These error conditions are presented and defined here.

Invalid status received - the contents of the status field of an end packet is not a valid status as defined by TMSCP

Port-detected error - examination of the SA register indicated an error condition exists within the controller

Program command timeout - the program received no end packet from the subsystem within the predefined command timeout.

Response out of sequence - the program received an end packet for a

sequential command other than the oldest out-
standing command.

Port initialization failed - the port failed to make an expected step transition during the UQ-Port init sequence.

Software data compare the program's data compare routine detected a miscomparison of read data to expected data.

Record length short - the data record read from tape was shorter than the record length expected.

3.7 Drive Errors

On occurrence of a Drive Error, status code of 13(8), the error log packet will now contain a status code which is the drive error byte as returned by the drive. This value will be placed in the DRV CODE field of the error log packet.

To understand the precise nature of the error condition it will be necessary to correlate the value presented in the printout against the table below.

Octal	Hex	Description
1	01	Write lock violation
2	02	Drive fault
4	04	Communication exception (timeout, etc.)
6	06	Wrong track error (following a turnaround)
10	08	No cable or drive powered off
20	10	Synchronization failure write/read
23	13	
44	22	
45	23	
47	27	
201	81	Failure to load to BOT
202	82	Failure to unload tape into cartridge
203	83	General motor or tach failure
204	84	Motor A failure
205	85	Motor B failure
206	86	Drive lost control of tape or bad tach
207	87	Excessive drag in tape transport
210	88	Failure to stop tape or remain stopped
211	89	Cartridge insert error
212	8A	Cartridge extract error
213	8B	CU attempted to move tape with drive in error
214	8C	Deceleration timeout error
215	8D	Second attempt to balance reels in init failed
220	90	8155 RAM memory failure in self-test
221	91	8155 timer failure
222	92	Read amplit (Hd 1) too low in calibrate
223	93	Read amplit (Hd 2) too low in calibrate
225	95	EOT sensed in R/W/S
226	96	BOT sensed in R/W/S
227	97	Drive block address overflow

230	98	Drive block address underflow
231	99	Servo error - excessive speed variations
231	9A	Failure in tracking currently not used
233	9B	Command error - not recognized
234	9C	Illegal command - incompatible with drive state
235	9D	Write lock error
236	9E	Write gate at wrong time
237	9F	No write gate for calibration track write
240	A0	Error sensing cal track 1 - bad head?
241	A1	Error sensing cal track 2 - bad head?
242	A2	Detection of edges of cal trk 1 out of spec
243	A3	Detection of edges of cal trk 2 out of spec
244	A4	Offset of cal trk 2 from 1 is too great
245	A5	Search for bottom edge of tape failed
246	A6	Bottom tape edge tolerance error
247	A7	Drive is overheating
250	A8	No current in LED of BOT sensor (cable?)
251	A9	Hall switch sense lines Motor A questionable
252	AA	Tachometer failure

3.8 Hard Error Reports

Hard error reports, if not user disabled, will be generated anytime an error recovery process does not successfully complete.

Hard Error reports will typically be of the following format:

```
CZTU1 HRD ERR 00014 ON UNIT 00 TST 003 SUB 000 PC: 020460
HARD DATA ERROR
COMMAND: RD     T/MSCP UNIT: 000(O)
PASS:   1(D)    DATA PAT: 01(O)
RECORD BYTE COUNT:    457(D)
OBJECT CNT : 000000026352(O)

RESPONSE PACKET
      HIGH WORD      LOW WORD
      000000(O)      026532(O)
      000000(O)      000000(O)
      000050(O)      010240(O)
      000000(O)      000733(O)
      000000(O)      000000(O)
      000000(O)      000000(O)
      000000(O)      000000(O)
      000000(O)      001413(O)
      000000(O)      000733(O)
```

NOTE

Some error reports will not include a Response Packet field. For example a Command Timeout Error, by definition, results only when no response to a command has been received prior to expiration of the programs watch dog timer.

817
818 3.9 Soft Error Reports
819
820 Soft error reports, if not user disabled, will be generated
821 anytime an error recovery process is successful. The soft error
822 report will include the number of retries necessary in order to
823 successfully complete the current operation. Soft Error reports will
824 typically be of the following format:
825
826 CZTU1 SFT RD ERR 00014 ON UNIT 00 TST 003 SUB 000 PC: 020460
827 ECC RECOVERED DATA ERROR
828 COMMAND: RD T/MSCP UNIT: 000(O)
829 PASS: 1(D) DATA PAT: 01(O)
830 OBJECT CNT : 000000026352(O)
831 TAP OBJ CNT: 000000026352(O)
832 TRK NUM: 6(D) LEVEL: 0(D) RETRIES: 1(D)
833 LOG BLK NUM: 0(D) PHYS BLK NUM: 9932(D)
834 DRV CODE: 000(O) DRV FLGS: 041(O)
835 DRV STATE: 000000(O) INTERN STATUS: 002(O)
836 TAP CNT 0: 227(O) TAP CNT 1: 015(O)
837 TAP CNT 2: 035(O) RD/WR STATE: 000000(O)
838 OPER FLGS: 000000(O)
839

4 PERFORMANCE AND PROGRESS REPORTS

4.1 Statistics Matrix

	READ CH 1	READ CH 2	WRITE CH 1	WRITE CH 2
SOFT DATA ERRORS				
RETRY RECOVERED	X	X	X	X
ECC CORRECTED	X	X	N.A.	
HARD DATA ERRORS	X	X	X	X
CRC ON ECC BLOCK	X	X	N.A.	
DATA COMPARE ERRORS	X		N.A.	
DATA UNDERRUN	N.A.		X	
DATA OVERRUN	X		N.A.	
MISPOSITIONS	X		X	
OTHERS	X			
TIMES DROPPED	X			
BYTES WRITTEN	X,XXX,XXX,XXX			
BYTES READ	X,XXX,XXX,XXX			

TRK	PHY BLK	HWR	HRD	SWR	SRD
0	26	0	0	1	0
0	2474	0	0	1	0
1	126	0	0	1	0
1	10374	0	0	1	0

4.2 Read Error Definition

1. SOFT DATA ERRORS

   0 Retry Corrected - ECC disabled or repositioning was required because >1 block in ECC group was bad.

   0 ECC Corrected - CRC error occurred on data block but ECC has corrected it

2. Hard Data Errors - Maximum retries exhausted and data not recovered.

3. CRC Error on ECC Block - Data was read successfully, but CRC error occurred

4. Data Compare - No hardware detected errors, but the data compare failed. on an associated ECC block.

5. Data Overrun - The controller did not have sufficient buffer space for read data.

898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925

4.3 Write Error Definition

1. Retry Recovered - Operational write algorithm was enabled and controller successfully recovered from a write error. (In this case, media-induced write errors will appear in this category.)

2. Hard Data Errors - Write retries exhausted and block not successfully written.

3. Underrun - Controller ran out of write data blocks prior to a record boundary.

4.4 Miscellaneous

1. Mispositions - Times the drive lost position on tape.

2. Others - This is a tally of all errors not specifically called out in the error matrix.

3. Times Dropped - Times the drive has been dropped by the program.

5 TEST DESCRIPTIONS

5.1 Test 1 Basic Function Test

This test will execute a subset of the available commands on the unit under test. It serves as a quick verify test to ascertain that the unit can move tape and write/read predictably, without error. The subset of legal commands will be issued in a coherent manner.

The testing sequence, performed once will be as follows:

1. Execute online
2. Rewind to ensure that tape is at BOT,
3. Write two tapemarks, just after BOT,
4. Backspace two tapemarks,
5. Space forward to LEOT,
6. Rewind,
7. Write, using increasing byte counts, rotating through all data patterns, using decreasing file lengths. Files to be separated by tape marks.
8. Write LEOT after previous sequence,
9. Rewind,
10. Read records of the first file,
11. Space records over the second and third files,
12. Space objects over the fourth file,
13. Read records of the fifth file,
14. Skip reverse over four tape marks,
15. Skip forward one tape mark,
16. Read the second file set,
17. Space objects over the third record set,
18. Read the fourth record set,
19. Space objects to LEOT,
20. Space objects reverse to Just after BOT,
21. Skip four tape marks,
22. Space records over the fourth record set,

23. Skip a tape mark.

24. Read the sixth record set.

25. Skip two tape marks.

26. Space objects reverse to the end of the second file set.

27. Skip a tape mark.

28. Read the third file set.

29. Rewind tape.

5.2 Test 2 Quick Verify Read/Write Test

This test rewinds the tape, then executes the following sequence:

1. Write record set.

2. Write LEOT.

3. Rewind.

4. Reposition to just written record set.

5. Read the current record set.

6. Skip to LEOT.

for 5 iterations or until fatal error is encountered. This test permits retries, fixed record length (4096 bytes decimal), fixed number of records/set (250), and predetermined data patterns. This test will execute in a round-robin manner.

5.3 Test 3 Complex Read/Write Test

This test rewinds the tape, and executes the following sequence:

1. Write N records.

2. Write a tape mark.

3. Repeat 1 and 2 until EOT is reached.

4. Write 2 tape marks (LEOT).

5. Rewind.

6. Read N records.

7. Space 1 record (should see unexpected tape mark)

8. Repeat 6 and 7 until LEOT.

Number of records (N), and record size will be randomly selected. This sequence will permit hardware retries, if enabled by the operator. This test will run until EOT, LEOT or fatal error is detected. All data patterns including random data will be used in this test.

5.4 Test 4 Write Interchange Tape

This test will rewind the tape, then write until EOT or a fatal error is encountered. This test will keep track of the number of records and files written. If a fatal error is encountered, a message will report it, the tape on the unit will be rewound, and the unit prevented from executing further write operations.

5.5 Test 5 Read Unknown Tape

This test will rewind a tape, then read until EOT, LEOT or fatal error is encountered. This test will keep track of the number of records and files read. If a fatal error is encountered, a message will report it, the tape on the unit will be rewound, and the unit prevented from executing further read operations.

NOTE

Tests 4 and 5 can be used to perform a media interchange test for multiple drives. The program will not attempt to make any determination as to whether the unit that wrote the tape or the unit reading the tape is at fault for any errors.

5.6 Test 6 Start/Stop Write/Read Test

This test rewinds the tape, then executes the following sequence:

1. Write record set, stopping between each record,
2. Write a tape mark,
3. Repeat steps one and two until two tracks have been written,
4. Write LEOT,
5. Rewind,
6. Read the record set stopping between each record,
7. Skip a tape mark,
8. Repeat steps six and seven until LEOT is detected,

```
1098
1099                                        9.  Rewind.
1100
1101                                    Until fatal error is encountered.  This test  permits  retries,  fixed
1102                                    record length (8096 bytes decimal), fixed number of records/set (250),
1103                                    and  predetermined  data  patterns.   This  test  will  execute  in  a
1104                                    round-robin manner.
1105
1106
1107
1108                                    5.7  Test 7 Conversation Test
1109
1110                                        Conversation mode will run with or without  error  reports.   The
1111                                    user can select, from a list of commands, a sequence which can be used
1112                                    to emulate a known failure  mode.   Between  commands,  the  user  can
1113                                    specify unique delays, ranging from 10 to 250 ms.  The user can follow
1114                                    each tape command  with  integer  values,  the  first  indicating  the
1115                                    byte/record/file  count and the second indicating the # of repetitions
1116                                    necessary for that command.
1117                                    @
1118
1123
1134                                    .TITLE PROGRAM HEADER AND TABLES
1135                                    .SBTTL  PROGRAM HEADER
1161
1163 000000                                .ENABL  ABS,AMA
1164                                       .DSABL  GBL
1165          002000                       .       =       2000
1167
1168 002000                                BGNMOD
1169
1170                                    ;++
1171                                    ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
1172                                    ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
1173                                    ;--
1174
1175 002000                                POINTER BGNSW,BGNSFT,BGNRPT,ERRTBL,BGNDU,BGNSETUP
1176
1184
1185
1186 002000                                HEADER  CZTU1,A,0,15000,1,0
     002000                           L$NAME::                ;DIAGNOSTIC NAME
     002000      103                          .ASCII /C/
     002001      132                          .ASCII /Z/
     002002      124                          .ASCII /T/
     002003      125                          .ASCII /U/
     002004      061                          .ASCII /1/
     002005      000                          .BYTE   0
     002006      000                          .BYTE   0
     002007      000                          .BYTE   0
     002010                           L$REV::                 ;REVISION LEVEL
     002010      101                          .ASCII  /A/
     002011                           L$DEPO::                ;0
     002011      060                          .ASCII  /0/
     002012                           L$UNIT::                ;NUMBER OF UNITS
     002012  000001                           .WORD   T$PTHV
     002014                           L$TIML::                ;LONGEST TEST TIME
```

PROGRAM HEADER

```
002014  015000                  .WORD   15000
002016
002016  046500          L$HPCP::.WORD   L$HARD        ;POINTER TO H.W. QUES.
002020
002020  046566          L$SPCP::.WORD   L$SOFT        ;POINTER TO S.W. QUES.
002022
002022  002210          L$HPTP::.WORD   L$HW          ;PTR. TO DEF. H.W. PTABLE
002024
002024  002216          L$SPTP::.WORD   L$SW          ;PTR. TO S.W. PTABLE
002026
002026  110740          L$LADP::.WORD   L$LAST        ;DIAG. END ADDRESS
002030
002030  000000          L$STA::.WORD    0             ;RESERVED FOR APT STATS
002032
002032  000000          L$CO::  .WORD   0
002034
002034  000001          L$DTYP::.WORD   1             ;DIAGNOSTIC TYPE
002036
002036  000000          L$APT:: .WORD   0             ;APT EXPANSION
002040
002040  002124          L$DTP:: .WORD   L$DISPATCH    ;PTR. TO DISPATCH TABLE
002042
002042  000000          L$PRIO::.WORD   0             ;DIAGNOSTIC RUN PRIORITY
002044
002044  000000          L$ENVI::.WORD   0             ;FLAGS DESCRIBE HOW IT WAS SETUP
002046
002046  000000          L$EXP1::.WORD   0             ;EXPANSION WORD
002050
002050     004          L$MREV::.BYTE   C$REVISION    ;SVC REV AND EDIT #
002051     000                  .BYTE   C$EDIT
002052
002052  000000          L$EF::  .WORD   0             ;DIAG. EVENT FLAGS
002054  000000                  .WORD   0
002056
002056  000000          L$SPC:: .WORD   0
002060
002060  002200          L$DEVP::.WORD   L$DVTYP       ; POINTER TO DEVICE TYPE LIST
002062
002062  034536          L$REPP::.WORD   L$RPT         ;PTR. TO REPORT CODE
002064
002064  000000          L$EXP4::.WORD   0
002066
002066  000000          L$EXP5::.WORD   0
002070
002070  000000          L$AUT:: .WORD   0             ;PTR. TO ADD UNIT CODE
002072
002072  040270          L$DUT:: .WORD   L$DU          ;PTR. TO DROP UNIT CODE
002074
002074  000000          L$LUN:: .WORD   0             ;LUN FOR EXERCISERS TO FILL
002076
002076  002142          L$DESP::.WORD   L$DESC        ;POINTER TO DIAG. DESCRIPTION
002100
002100  104035          L$LOAD::EMT     E$LOAD        ;GENERATE SPECIAL AUTOLOAD EMT
002102
002102  013166          L$ETP:: .WORD   L$ERRTBL      ;POINTER TO ERRTBL
002104
002104  036644          L$ICP:: .WORD   L$INIT        ;PTR. TO INIT CODE
```

PROGRAM HEADER

```
002106                          L$CCP::                 ;PTR. TO CLEAN-UP CODE
002106  037760                          .WORD   L$CLEAN
002110                          L$ACP::                 ;PTR. TO AUTO CODE
002110  037756                          .WORD   L$AUTO
002112                          L$PRT::                 ;PTR. TO PROTECT TABLE
002112  020640                          .WORD   L$PROT
002114                          L$TEST::                ;TEST NUMBER
002114  000000                          .WORD   0
002116                          L$DLY::                 ;DELAY COUNT
002116  000000                          .WORD   0
002120                          L$HIME::                ;PTR. TO HIGH MEM
002120  000000                          .WORD   0
```

```
                                        .SBTTL  DISPATCH TABLE

                                        ;++
                                        ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST
                                        ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST
                                        ;--

002122                                          DISPATCH 7
002122  000007                                  .WORD   7
002124                                  L$DISPATCH::
002124  040304                                  .WORD   T1
002126  042626                                  .WORD   T2
002130  043446                                  .WORD   T3
002132  044306                                  .WORD   T4
002134  044740                                  .WORD   T5
002136  045314                                  .WORD   T6
002140  046116                                  .WORD   T7

002142                                          DESCRIPT        <CZTU1A0 TU81 DATA RELIAB TEST>
002142                                  L$DESC::
002142      103     132     124                 .ASCIZ  /CZTU1A0 TU81 DATA RELIAB TEST/
                                                .EVEN

                                        ;
                                        ; NAMES OF DEVICES SUPPORTED BY PROGRAM
                                        ;

002200                                          DEVTYP  <TU81>
002200                                  L$DVTYP::
002200      124     125     070                 .ASCIZ  #TU81#
                                                .EVEN
```

```
                                .SBTTL  DEFAULT HARDWARE P TABLE

                                ;++
                                ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
                                ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
                                ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P TABLES.
                                ;--

002206                                  BGNHW   DFPTBL
002206  000002                         .WORD   L10000-L$HW/2
002210                          L$HW::
002210                          DFPTBL::

002210  174500                          174500                  ;TKIP ADDRESS
002212  000000                          0                       ;T/MSCP UNIT NUMBER

002214                                  ENDHW
002214                          L10000:
```

```
                        .SBTTL  SOFTWARE P TABLE

                        ;++
                        ; THE SOFTWARE P TABLE CONTAINS THE VALUES OF THE PROGRAM
                        ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
                        ;--

002214                          BGNSW   SFPTBL
002214  000042                 .WORD   L10001 L$SW/2
002216                  L$SW::
002216                  SFPTBL::
002216  000             CLOCK::         .BYTE   0       ;ENABLE TIME OF DAY CLOCK
002217  000             HOURS::         .BYTE   0       ;HOURS FOR TIME OF DAY CLOCK
002220  000             MINUTE::        .BYTE   0       ;MINUTES FOR TIME OF DAY CLOCK
002221  000             SECOND::        .BYTE   0       ;SECONDS FOR TIME OF DAY CLOCK
002222  000             SUBSEC::        .BYTE   0       ;SUBSECONDS FOR TIME OF DAY CLOCK

002223  000             CONTPA::        .BYTE   0       ;CHANGE CONTROLLER PARARMETERS
002224  001             SERCOR::        .BYTE   1       ;ENABLE ERROR CORRECTION FLAG
002225  001             SERREC::        .BYTE   1       ;ENABLE ERROR RECOVERY FLAG
002226  001             DENSITY::       .BYTE   1       ;INITIAL TEST DENSITY (GCR)

002227  000             PRNTPA::        .BYTE   0       ;CHANGE PRINT PARAMETERS
002230  000             SOERRP::        .BYTE   0       ;ENABLE SOFT ERROR REPORT FLAG
002231  001             RDSOER::        .BYTE   1       ;ENABLE READ SOFT ERRORS ONLY

002232                  T7TBL::                         ;COMMAND TABLE TOP  6

002232  000             NOCLR::         .BYTE   0       ;ENABLE CLEAR STATS ON FATAL ERROR
002233  000             DMPFLG::        .BYTE   0       ;ENABLE PROGRAM TABLE DUMP ON ERROR

002234  000             TESTPA::        .BYTE   0       ;CHANGE TEST PARAMETERS
002235  000             PATERN::        .BYTE   0       ;CHANGE DATA PATTERN
002236  001             T5CMP::         .BYTE   1       ;ENABLE DATA COMPARES IN TEST 5
002237  000             CHGFLG::        .BYTE   0       ;CHANGE CMD SEQ TABLE FLAG

002240  160             T7CMD1:         .BYTE   REW     ;REWIND
002241  000                             .BYTE   NULPAT
002242  000000                          .WORD   0
002244  000001                          .WORD   1

002246  020             T7CMD2:         .BYTE   WR      ;WRITE RECORDS
002247  007                             .BYTE   PAT7
002250  004000                          .WORD   2048.
002252  000310                          .WORD   200.

002254  100             T7CMD3:         .BYTE   WTM     ;WRITE TAPE MARK
002255  000                             .BYTE   NULPAT
002256  000000                          .WORD   0
002260  000002                          .WORD   2

002262  061             T7CMD4:         .BYTE   SKR     ;SKIP TAPE MARKS REVERSE
002263  000                             .BYTE   NULPAT
002264  000000                          .WORD   0
002266  000002                          .WORD   2
```

```
1276
1277 002270     160                T7CMD5:         .BYTE   REW     ;REWIND
1278 002271     000                                .BYTE   NULPAT
1279 002272  000000                                .WORD   0
1280 002274  000001                                .WORD   1
1281
1282 002276     010                T7CMD6:         .BYTE   RD      ;READ RECORDS
1283 002277     007                                .BYTE   PAT7
1284 002300  004000                                .WORD   2048.
1285 002302  000310                                .WORD   200.
1286
1287 002304     060                T7CMD7:         .BYTE   SKP     ;SKIP TAPE MARK
1288 002305     000                                .BYTE   NULPAT
1289 002306  000001                                .WORD   1
1290 002310  000002                                .WORD   2
1291
1292 002312     160                T7END:          .BYTE   REW     ;REWIND
1293 002313     000                                .BYTE   NULPAT
1294 002314  000000                                .WORD   0
1295 002316  C00001                                .WORD   1
1296
1297 002320  177777                                .WORD   -1
1298                                                .EVEN
1299
1300 002322                             ENDSW
     002322                        L10001:
1301
1302 002322                             ENDMOD
```

```
                                .TITLE GLOBAL AREAS
                                .SBTTL  GLOBAL EQUATES SECTION

002322                                  BGNMOD


                                ;/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\
                                ; 1.0   SUPERVISOR DEFINED LITERALS
                                ;/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\


                                ;++
                                ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
                                ; ARE USED IN MORE THAN ONE TEST.
                                ;--

002322                                  EQUALS
                                ;
                                ; BIT DIFINITIONS
                                ;
100000                          BIT15== 100000
040000                          BIT14== 40000
020000                          BIT13== 20000
010000                          BIT12== 10000
004000                          BIT11== 4000
002000                          BIT10== 2000
001000                          BIT09== 1000
000400                          BIT08== 400
000200                          BIT07== 200
000100                          BIT06== 100
000040                          BIT05== 40
000020                          BIT04== 20
000010                          BIT03== 10
000004                          BIT02== 4
000002                          BIT01== 2
000001                          BIT00== 1
                                ;
001000                          BIT9==  BIT09
000400                          BIT8==  BIT08
000200                          BIT7==  BIT07
000100                          BIT6==  BIT06
000040                          BIT5==  BIT05
000020                          BIT4==  BIT04
000010                          BIT3==  BIT03
000004                          BIT2==  BIT02
000002                          BIT1==  BIT01
000001                          BIT0==  BIT00
                                ;
                                ; EVENT FLAG DEFINITIONS
                                ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                                ;
                                ;                                       ;  BIT POSITION IN SECOND STATUS WORD
000040                          EF.START==      32.                     ; (100000) START COMMAND WAS ISSUED
000037                          EF.RESTART==    31.                     ; (040000) RESTART COMMAND WAS ISSUED
000036                          EF.CONTINUE==   30.                     ; (020000) CONTINUE COMMAND WAS ISSUED
000035                          EF.NEW==        29.                     ; (010000) A NEW PASS HAS BEEN STARTED
```

```
                000034                  EF.PWR==        28.             ; (004000) A POWER-FAIL/POWER UP OCCURRED
                                        ;
                                        ;
                                        ; PRIORITY LEVEL DEFINITIONS
                                        ;
                000340                  PRI07== 340
                000300                  PRI06== 300
                000240                  PRI05== 240
                000200                  PRI04== 200
                000140                  PRI03== 140
                000100                  PRI02== 100
                000040                  PRI01== 40
                000000                  PRI00== 0
                                        ;
                                        ;OPERATOR FLAG BITS
                                        ;
                000004                  EVL==     4
                000010                  LOT==    10
                000020                  ADR==    20
                C00040                  IDU==    40
                000100                  ISR==   100
                000200                  UAM==   200
                000400                  BOE==   400
                001000                  PNT==  1000
                002000                  PRI==  2000
                004000                  IXE==  4000
                010000                  IBE== 10000
                020000                  IER== 20000
                040000                  LOE== 40000
                100000                  HOE== 100000
1407
```

```
1409
1410
1411                            ;/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\
1412                            ; 2.0   TMSCP COMMAND LITERALS
1413                            ;/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\
1414
1415                            ; 2.1   COMMAND PACKET OPCODES
1416            000001          OP.ABO  ==      01              ;ABORT COMMAND
1417            000020          OP.ACC  ==      20              ;ACCESS COMMAND
1418            000010          OP.AVL  ==      10              ;AVAILABLE COMMAND
1419            000040          OP.CMP  ==      40              ;COMPARE HOST DATA COMMAND
1420            000013          OP.DAP  ==      13              ;DETERMINE ACCESS PATH COMMAND
1421            000022          OP.ERS  ==      22              ;ERASE COMMAND
1422            000026          OP.ERG  ==      26              ;ERASE GAP COMMAND
1423            000002          OP.GCS  ==      02              ;GET COMMAND STATUS COMMAND
1424            000003          OP.GUS  ==      03              ;GET UNIT STATUS COMMAND
1425            000011          OP.ONL  ==      11              ;ONLINE COMMAND
1426            000041          OP.RD   ==      41              ;READ COMMAND
1427            000045          OP.REP  ==      45              ;REPOSITION COMMAND
1428            000004          OP.SCC  ==      04              ;SET CONTROLLER CHARACTERISTICS COMMAND
1429            000012          OP.SUC  ==      12              ;SET UNIT CHARACTERISTICS COMMAND
1430            000042          OP.WR   ==      42              ;WRITE COMMAND
1431            000044          OP.WTM  ==      44              ;WRITE TAPE MARK COMMAND
1432            000200          OP.END  ==      200             ;END MESSAGE FLAG
1433            000100          OP.AVA  ==      100             ;AVAILABLE ATTENTION MESSAGE
1434            000102          OP.ACP  ==      102             ;ACCESS PATH ATTENTION MESSAGE
1435
1436                            ; 2.2   COMMAND MODIFIERS
1437            040000          MD.CMP  ==      040000          ;COMPARE
1438            020000          MD.CSE  ==      020000          ;CLEAR SERIOUS EXCEPTION
1439            001000          MD.SEC  ==      001000          ;SUPPRESS ERROR CORRECTION
1440            000400          MD.SER  ==      000400          ;SUPPRESS ERROR RECOVERY
1441            000200          MD.DLE  ==      000200          ;DETECT LEOT
1442            000100          MD.IMM  ==      000100          ;IMMEDIATE
1443            000040          MD.EXC  ==      000040          ;EXCLUSIVE ACCESS
1444            000020          MD.UNL  ==      000020          ;UNLOAD
1445            000010          MD.REV  ==      000010          ;REVERSE
1446            000004          MD.OBC  ==      000004          ;OBJECT COUNT
1447            000004          MD.SWP  ==      000004          ;SET WRITE PROTECT
1448            000002          MD.RWD  ==      000002          ;REWIND
1449            000002          MD.ALL  ==      000002          ;ALL CLASS DRIVERS
1450            000001          MD.SPD  ==      000001          ;SPEED
1451            000001          MD.NXU  ==      000001          ;NEXT UNIT
1452
1453                            ; 2.3   GENERIC COMMAND PACKET OFFSETS
1454            000000          P.CRF   ==       0              ;COMMAND REFERENCE NUMBER
1455            000004          P.UNIT  ==       4              ;UNIT NUMBER
1456            000010          P.OPCD  ==       10             ;OPCODE
1457            000012          P.MOD   ==       12             ;MODIFIERS
1458            000014          P.BCNT  ==       14             ;BYTE COUNT
1459            000020          P.BUFF  ==       20             ;BUFFER DESCRIPTOR
1460
1461                            ; 2.4   ABORT AND GET COMMAND STATUS OFFSETS PACKET OFFSETS
1462            000014          P.OTRF  ==       14             ;OUTSTANDING COMMAND REFERENCE NUMBER
1463
1464                            ; 2.5   ONLINE AND SET UNIT CHARACTERISTICS PACKET OFFSETS
1465            000014          P.UNFL  ==       14             ;UNIT FLAGS
```

GLOBAL EQUATES SECTION

```
1466          000034                  P.DVPM  ==      34              ;DEVICE DEPENDENT PARAMETERS
1467          000040                  P.FORM  ==      40              ;FORMAT
1468          000042                  P.SPED  ==      42              ;SPEED
1469
1470                                  ; 2.6   REPOSITION COMMAND PACKET OFFSETS
1471          000014                  P.REDD  ==      14              ;RECORD/OBJECT COUNT
1472          000020                  P.TMGC  ==      20              ;TAPE MARK COUNT
1473
1474                                  ; 2.7   SET CONTROLLER CHARACTERISTICS  PACKET OFFSETS
1475          000014                  P.VRSN  ==      14              ;MSCP VERSION
1476          000016                  P.CNTF  ==      16              ;CONTROLLER FLAGS
1477          000020                  P.HTMO  ==      20              ;HOST TIMEOUT
1478          000024                  P.TIME  ==      24              ;QUAD-WORD TIME AND DATE
1479          000034                  P.CTPM  ==      34              ;CONTROLLER DEPENDENT PARAMETERS
1480
```

```
1482                                ;/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\
1483                                ; 3.0   TMSCP END PACKET LITERALS
1484                                ;/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\
1485
1486                                ; 3.1   END PACKET FLAGS
1487    000040                      EF.LOG  ==      000040          ;ERROR LOG GENERATED
1488    000020                      EF.SEX  ==      000020          ;SERIOUS EXCEPTION
1489    000010                      EF.EOT  ==      000010          ;EOT ENCOUNTERED
1490
1491                                ; 3.2   CONTROLLER FLAGS
1492    000200                      CF.ATN  ==      000200          ;ENABLE ATTENTION INTERUPTS
1493    000100                      CF.MSC  ==      000100          ;ENABLE MISCELLANEOUS ERROR LOG MESSAGES
1494    000040                      CF.OTH  ==      000040          ;ENABLE OTHER HOSTS ERROR LOG MESSAGES
1495    000020                      CF.THS  ==      000020          ;ENABLE THIS HOSTS ERROR LOG MESSAGES
1496
1497                                ; 3.3   UNIT FLAGS
1498    020000                      UF.WPH  ==      020000          ;WRITE PROTECT (HARDWARE)
1499    020000                      UF.VSU  ==      020000          ;VARIABLE SPEED UNIT
1500    010000                      UF.WPS  ==      010000          ;SOFTWARE WRITE PROTECT
1501    C00200                      UF.RMV  ==      000200          ;REMOVABLE MEDIA
1502    000040                      UF.VSS  ==      000040          ;VARIABLE SPEED MODE SUPPRESSION
1503    000002                      UF.CMW  ==      000002          ;COMPARE WRITES
1504    000001                      UF.CMR  ==      000001          ;COMPARE READS
1505
1506                                ; 3.4   GENERIC END MESSAGE OFFSETS
1507    000000                      P.CRF   ==      0               ;COMMAND REFERENCE NUMBER
1508    000004                      P.UNIT  ==      4               ;UNIT NUMBER
1509    000010                      P.OPCD  ==      10              ;ENDCODE
1510    000011                      P.FLGS  ==      11              ;END MESSAGE FLAGS
1511    000012                      P.STS   ==      12              ;STATUS
1512    000014                      P.BCNT  ==      14              ;BYTE COUNT
1513
1514                                ; 3.5   ACCESS,COMPARE HOST DATA,READ,REPOSITION,WRITE,-
1515                                ;       AND WRITE TAPE MARK END MESSAGE OFFSETS
1516    000034                      P.POS   ==      34              ;POSITION (OBJECT COUNT)
1517    000040                      P.TRBC  ==      40              ;TAPE RECORD BYTE COUNT
1518
1519                                ; 3.6   GET COMMAND STATUS END PACKET OFFSETS
1520    000014                      P.OTRF  ==      14              ;OUTSTANDING COMMAND REFERENCE NUMBER
1521    000020                      P.CMST  ==      20              ;COMMAND STATUS
1522
1523                                ; 3.7   GET UNIT STATUS END MESSAGE OFFSETS
1524    000014                      P.MLUN  ==      14              ;MULTI-UNIT CODE
1525    000016                      P.UNFL  ==      16              ;UNIT FLAGS
1526    000024                      P.UNTI  ==      24              ;UNIT IDENTIFIER
1527    000034                      P.MEDI  ==      34              ;MEDIA IDENTIFIER
1528    000040                      P.FORM  ==      40              ;TAPE FORMAT
1529    000042                      P.SPED  ==      42              ;SPEED
1530    000044                      P.FMEM  ==      44              ;FORMAT MENU
1531    000050                      P.CSVR  ==      50              ;CONTROLLER SOFTWARE VERSION
1532    000051                      P.CHVR  ==      51              ;CONTROLLER HARDWARE VERSION
1533    000052                      P.USVR  ==      52              ;UNIT SOFTWARE VERSION
1534    000053                      P.UHVR  ==      53              ;UNIT HARDWARE VERSION
1535
1536                                ; 3.8   OP.ONL, OP.AVL AND OP.SUC MESSAGE OFFSETS
1537    000044                      P.MXWR  ==      44              ;MAXIMUM WRITE RECORD SIZE
1538    000050                      P.NREC  ==      50              ;NOISE RECORD
```

```
1539
1540                                    ; 3.9   REPOSITION MESSAGE OFFSETS
1541            000014                  P.RCSK  ==      14              ;RECORDS SKIPPED
1542            000020                  P.TMSK  ==      20              ;TAPE MARKS SKIPPED
1543
1544                                    ; 3.10  SET CONTROLLER CHARACTERISTICS MESSAGE OFFSETS
1545            000014                  P.VRSN  ==      14              ;MSCP VERSION
1546            000016                  P.CNTF  ==      16              ;CONTROLLER FLAGS
1547            000020                  P.HTMO  ==      20              ;HOST TIMEOUT
1548            000024                  P.TIME  ==      24              ;QUAD-WORD TIME AND DATE
```

```
1550                            ;/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\
1551                            ; 4.0   ERROR LOG LITERALS
1552                            ;/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\
1553
1554                            ; 4.1   ERROR LOG MESSAGE FORMAT CODES
1555  000000                    FM.CNT  ==      000000          ;CONTROLLER ERRORS
1556  000001                    FM.BAD  ==      000001          ;HOST MEMORY ACCESS ERRORS WITH BUS ADDRESS
1557  000005                    FM.TPE  ==      000005          ;TAPE TRANSFER ERRORS
1558
1559                            ; 4.2   ERROR LOG MESSAGE FLAGS
1560  000200                    LF.SUC  ==      000200          ;OPERATION SUCCESSFUL
1561  000100                    LF.CON  ==      000100          ;OPERATION CONTINUING
1562  000001                    LF.SNR  ==      000001          ;SEQUENCE NUMBER REQUEST
1563
1564                            ; 4.3   TAPE FORMAT FLAG VALUES
1565  000001                    TF.800  ==      000001          ;NRZI 800 BPI
1566  000002                    TF.PE   ==      000002          ;PHASE ENCODED 1600 BPI
1567  000004                    TF.GCR  ==      000004          ;GROUP CODED RECORDING 6250 BPI
1568  000010                    TF.BLK  ==      000010          ;6667 BPI
1569
1570                            ; 4.4   ERROR LOG MESSAGE OFFSETS
1571  000000                    L.CRF   ==      0               ;COMMAND REFERENCE NUMBER
1572  000004                    L.UNIT  ==      4               ;UNIT NUMBER
1573  000006                    L.SEQN  ==      6               ;SEQUENCE NUMBER
1574  000010                    L.FMT   ==      10              ;FORMAT
1575  000011                    L.FLGS  ==      11              ;ERROR LOG MESSAGE FLAGS
1576  000012                    L.EVNT  ==      12              ;EVENT CODES
1577  000014                    L.CNTI  ==      14              ;CONTROLLER ID
1578  000024                    L.CSVR  ==      24              ;CONTROLLER SOFTWARE VERSION
1579  000025                    L.CHVR  ==      25              ;CONTROLLER HARDWARE VERSION
1580  000026                    L.MLUN  ==      26              ;MULTI UNIT CODE
1581  000030                    L.UNTI  ==      30              ;UNIT ID
1582  000030                    L.BADR  ==      30              ;BUS ADDRESS
1583  000040                    L.USVR  ==      40              ;UNIT SOFTWARE VERSION
1584  000041                    L.UHVR  ==      41              ;UNIT HARDWARE VERSION
1585  000042                    L.LVL   ==      42              ;RETRY LEVEL
1586  000042                    L.FMTD  ==      42              ;FORMAT DEPENDENT
1587  000043                    L.RTRY  ==      43              ;RETRY COUNT FOR THE CURRENT LEVEL
1588  000044                    L.GPCT  ==      44              ;GAP COUNT
1589  000044                    L.VSER  ==      44              ;VOLUME SERIAL NUMBER
1590  000044                    L.PSTN  ==      44              ;TAPE OBJECT COUNT
1591  000050                    L.STI   ==      50              ;STI INFORMATION
1592  000050                    L.FHVR  ==      50              ;FORMATTER HARDWARE VERSION
1593  000051                    L.FSVR  ==      51              ;FORMATTER SOFTWARE VERSION
1594  000052                    L.STS   ==      52              ;CONTROLLER INTERNAL STATUS
1595  000053                    L.DRVC  ==      53              ;DRIVE ERROR CODE
1596  000054                    L.DFLG  ==      54              ;DRIVE STATE FLAGS
1597  000055                    L.TRK   ==      55              ;LOGICAL TRACK NUMBER
1598  000056                    L.PBLK  ==      56              ;PHYSICAL BLOCK NUMBER
1599  000060                    L.LBLK  ==      60              ;LOGICAL BLOCK NUMBER
1600  000061                    L.CNT0  ==      61              ;TAPE COUNT 0
1601  000062                    L.CNT1  ==      62              ;TAPE COUNT 1
1602  000063                    L.CNT2  ==      63              ;TAPE COUNT 2
1603  000064                    L.DRVS  ==      64              ;DRIVE STATE
1604  000066                    L.RWST  ==      66              ;READ/WRITE STATE
1605  000070                    L.OPFL  ==      70              ;OPERATION FLAGS
1606
```

```
1607                                    ; 4.5   STATUS AND EVENT CODES
1608            000037                  ST.MSK  ==      37              ;STATUS/EVENT CODE MASK
1609            000040                  ST.SUB  ==      40              ;SUB-CODE MULTIPLIER
1610            000000                  ST.SUC  ==      0               ;SUCCESS
1611            000001                  ST.CMD  ==      1               ;INVALID COMMAND
1612            000002                  ST.ABO  ==      2               ;COMMAND ABORTED
1613            000003                  ST.OFL  ==      3               ;UNIT-OFFLINE
1614            000004                  ST.AVL  ==      4               ;UNIT-AVAILABLE
1615            000005                  ST.MFE  ==      5               ;MEDIA FORMAT ERROR
1616            000006                  ST.WPR  ==      6               ;WRITE PROTECTED
1617            000007                  ST.CMP  ==      7               ;COMPARE ERROR
1618            000010                  ST.DAT  ==      10              ;DATA ERROR
1619            000011                  ST.HST  ==      11              ;HOST BUFFER ACCESS ERROR
1620            000012                  ST.CNT  ==      12              ;CONTROLLER ERROR
1621            000013                  ST.DRV  ==      13              ;DRIVE ERROR
1622            000014                  ST.FNT  ==      14              ;FORMATTER ERROR
1623            000015                  ST.BOT  ==      15              ;BOT ENCOUNTERED
1624            000016                  ST.TM   ==      16              ;TAPE MARK ENCOUNTERED
1625            000020                  ST.RDT  ==      20              ;RECORD DATA TRUNCATED
1626            000021                  ST.POL  ==      21              ;POSITION LOST
1627            000022                  ST.SEX  ==      22              ;SERIOUS EXCEPTION
1628            000023                  ST.LED  ==      23              ;LEOT DETECTED
1629            000037                  ST.DIA  ==      37              ;INTERNAL DIAGNOSTIC MESSAGE
1630            000400                  ST.ONL  ==      400             ;UNIT ALREADY ONLINE
1631
1632            000010                  EV.LGP  ==      10              ;LONG GAP ENCOUNTERED
1633            000050                  EV.DST  ==      50              ;DATA SYNC TIMEOUT
1634            000052                  EV.CTO  ==      52              ;COMM CHANNEL TIMEOUT
1635            000053                  EV.SRT  ==      53              ;DRIVE COMMAND TIMEOUT
1636            000113                  EV.SRI  ==      113             ;CONTROLLER DECTECTED TRANSMISION ERROR
1637            000150                  EV.COR  ==      150             ;CORRECTABLE ERROR
1638            000152                  EV.IDS  ==      152             ;INTERNAL INCONSISTENCY ERROR
1639            000153                  EV.SER  ==      153             ;SOFT ERROR
1640            000213                  EV.HER  ==      213             ;HARD ERROR
1641            000350                  EV.URE  ==      350             ;UNRECORVERABLE DATA ERROR
```

```
1643                                    ;/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\
1644                                    ; 5.0   PROGRAM DEFINED LITERALS
1645                                    ;/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\
1646
1647                                    ; 5.1   TKSA BIT DEFINITIONS
1648    100000                          ERR     ==      100000          ;ERROR
1649    004000                          S1      ==      004000          ;STEP 1
1650    000001                          GO      ==      000001          ;GO
1651    111400                          TKINIT  ==      111400          ;TU81 STEP 1 RESPONSE
1652
1653                                    ; 5.2   DRIVE IN USE TABLE BIT DEFINITIONS
1654    000001                          AVB     ==      000001          ;DRIVE AVAILABLE
1655    000002                          NRDY    ==      000002          ;DRIVE NOT READY
1656    000004                          EOT     ==      000004          ;DRIVE AT EOT
1657    000010                          DROP    ==      000010          ;DRIVE DROPPED
1658    000020                          FAIL    ==      000020          ;DRIVE FAILED
1659
1660                                    ; 5.3   I/O STATUS MESSAGES
1661    000000                          IONORM  ==      0               ;SUCCESSFUL COMMAND TRANSMISSION
1662    100000                          NURESP  ==      BIT15           ;NEW RESPONSE RECEIVED
1663    040000                          ERRLOG  ==      BIT14           ;ERROR LOG PACKET RECEIVED
1664    020000                          IOICRD  ==      BIT13           ;INSUFFICIENT CREDIT TO POST COMMAND
1665    000001                          CMDLST  ==      1.              ;GCS RESPONSE NEVER CAME BACK
1666    000002                          IOHUNG  ==      2.              ;CONTROLLER HUNG
1667    000003                          IOPDRE  ==      3.              ;PORT DETECTED ERROR
1668    000004                          IOTIME  ==      4.              ;CONTROLLER TIME OUT
1669    000005                          MISSEQ  ==      5.              ;COMMAND RETURNED IN WRONG SEQUENCE
1670    000006                          INTERR  ==      6.              ;ERROR DURING INITIALIZATION
1671    000007                          ILLCMD  ==      7.              ;ILLEGAL COMMAND
1672
1673                                    ; 5.4   RESPONSE CONDITION CODES
1674    000001                          SUCCES  ==      000001          ;RESPONSE HANDLED SUCCESSFULLY
1675    000002                          SEREXC  ==      000002          ;SERIOUS EXCEPTION CONDITION
1676    000004                          ABORT   ==      000004          ;SYSTEM FATAL ERROR ABORT PROGRAM
1677
1678                                    ; 5.5   Q/Q PORT LITERALS
1679    100000                          OWN     ==      BIT15           ;DESCRIPTOR OWNERSHIP BIT
1680    040000                          FLAG    ==      BIT14           ;DESCRIPTOR INTERUPT FLAG BIT
1681    000200                          IMM     ==      BIT07           ;IMMEDIATE COMMAND FLAG
1682
1683                                    ; 5.6   PROGRAM LITERALS
1684    177546                          KWCSR   ==      177546          ;LINE CLOCK REGISTER ADDRESS
1685    001233                          RS1     ==      1233            ;RANDOM GENERATOR SEED
1686    007622                          RS2     ==      7622            ;RANDOM GENERATOR SEED
1687    000000                          RS3     ==      0               ;RANDOM GENERATOR SEED
1688    000000                          NULPAT  ==      0               ;NO DATA PATTERN NEEDED
1689    000000                          RNDBYT  ==      0               ;RANDOM BYTE COUNT
1690    020000                          MAXBUF  ==      8192.           ;MAXIMUM BUFFER SIZE
1691    000024                          MINBUF  ==      20.             ;MINIMUM BUFFER SIZE
1692    000000                          RNDITR  ==      0               ;RANDOM ITERATION COUNT
1693    003720                          MAXITR  ==      2000.           ;MAXIMUM ITERATION SET
1694    000144                          MINITR  ==      100.            ;MINIMUM ITERATION SET
1695    000001                          PAT1    ==      1               ;ALL 1'S DATA PATTERN
1696    000002                          PAT2    ==      2               ;ALL 0'S DATA PATTERN
1697    000003                          PAT3    ==      3               ;WORST CASE MFM DATA PATTERN
1698    000004                          PAT4    ==      4               ;ALTERNATING 1'S AND 0'S
1699    000005                          PAT5    ==      5               ;RANDOM DATA PATTERN
```

```
1700            000006                  PAT6    ==      6               ;1110 REPEATING PATTERN
1701            000007                  PAT7    ==      7               ;COMBINATION PATTERN 3 AND 5
1702            000010                  ENDPAT  ==      8.              ;RANDOM PATTERN VALUE
1703            000200                  ALLPAT  ==      200             ;CYCLE THROUGH ALL PATTERNS
1704            000002                  UNITSP  ==      2               ;STEP THROUGH UNITS
1705            000000                  HSTIMO  ==      0               ;HOST TIMEOUT VALUE
1706            000000                  MSCPVR  ==      0               ;MSCP VERSION NUMBER
1707            177776                  LOBYTE  ==      -2              ;LOW BYTE OFFSET FOR COMPARE DATA
1708            177777                  HIBYTE  ==      -1              ;HIGH BYTE OFFSET FOR COMPARE DATA
1709            004716                  T2END   ==      2510.           ;RECORDS TO FILL 2 TRACKS
1710            000004                  N       ==      4               ;VALUE USED IN SUBITR
1711            000001                  ONE     ==      1               ;BYTE OFFSET
1712
1713                                    ; 5.7   ERROR MASKING LITERALS
1714            000001                  LEDB    ==      000001          ;DETECT LOGICAL END OF TAPE
1715            000002                  RDTB    ==      000002          ;RECORD DATA TRUNCATED
1716            000004                  SEXB    ==      000004          ;SERIOUS EXCEPTION
1717            000010                  TMB     ==      000010          ;ENCOUNTERED TAPE MARK
1718            000020                  WPRB    ==      000020          ;DRIVE WRITE PROTECTED
1719            C00040                  AVLB    ==      000040          ;UNIT AVAILABLE
1720            000100                  ONLB    ==      000100          ;UNIT ONLINE
1721
1722                                    ; 5.8   ERROR TYPE LITERALS
1723            000000                  SYSFAT  ==      0               ;SYSTEM FATAL ERROR
1724            000001                  DEVFAT  ==      1               ;DEVICE FATAL ERROR
1725            000002                  HARD    ==      2               ;HARD DEVICE ERROR
1726            000003                  SOFT    ==      3               ;SOFT DEVICE ERROR
1727            000004                  STATUS  ==      4               ;STATUS MESSAGE
1728
1729                                    ; 5.9 BIT VALUES FOR LUN FLAG
1730            000001                  INITDON ==      000001          ;INITIALIZATION HAS BEEN DONE ON THIS UNIT
1731            000002                  SEREXC  ==      000002          ;A SERIOUS EXCEPTION CONDITION EXITS
1732            000004                  NOTALY  ==      000004          ;DON'T TALLY BYTES FOR THIS COMMAND
1733            000010                  EOTPR   ==      000010          ;EOT PRINTED FOR THIS UNIT
1734            000020                  ODDFLG  ==      000020          ;ODD BYTE COUNT FLAG
1735            000040                  MTBLOV  ==      000040          ;MEDIA STATS OVERFLOW FLAG
1736            000100                  ECCFLG  ==      000100          ;DON'T DECREMENT ECC COUNT FLAG
1737            000200                  RETFLG  ==      000200          ;RETRY FLAG
1738
1739                                    ;PROGRAM CONTROL FLAG BIT VALUES
1740            000001                  T7BRFL  ==      000001          ;BRANCH FLAG FOR TEST 7
1741            000002                  NCLKFL  ==      000002          ;NO CLOCK PRESENT FLAG
1742            000004                  TCNTFL  ==      000004          ;COUNT RECORDS AND TAPE MARKS FLAG
1743            000010                  DRERFL  ==      000010          ;DRIVE ERROR FLAG
1744            000020                  GCSCFL  ==      000020          ;GET COMMAND STATUS COMMAND FLAG
1745            000040                  GCSRFL  ==      000040          ;GET COMMAND STATUS RESPONSE FLAG
1746            000100                  CMDONE  ==      000100          ;ALL COMMANDS ISSUED FLAG
1747            000200                  DROPIT  ==      000200          ;DRIVE BEING DROPPED
1748            000400                  TPASS1  ==      000400          ;FIRST PASS THROUGH TEST
```

```
1750                            ;/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\
1751                            ; 6.0   PROGRAM OFFSETS
1752                            ;/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\
1753
1754                            ; 6.1   TEST TABLE OFFSETS
1755            000000          CMD     ==      0.              ;COMMAND PRIMITIVE OFFSET
1756            000001          DATPAT  ==      1.              ;DATA PATTERN
1757            000002          ITMCNT  ==      2.              ;ITEM (BYTE) COUNT OFFSET
1758            000004          ITRCNT  ==      4.              ;ITERATION COUNT OFFSET
1759            000006          TSTSTP  ==      6.              ;USED TO STEP THROUGH TEST TABLES
1760
1761                            ; 6.2   PROGRAM COMMAND BUFFER OFFSETS
1762            000000          CMD     ==      0               ;COMMAND PRIMITIVE OFFSET
1763            000002          ITMOFF  ==      2.              ;ITEM (BYTE) COUNT OFFSET
1764            000004          OBOFFL  ==      4.              ;LOW ORDER OBJECT COUNT
1765            000006          OBOFFH  ==      6.              ;HIGH ORDER OBJECT COUNT
1766            000010          XFERST  ==      8.              ;TRANSFER STATUS
1767            000012          BUFOFF  ==      10.             ;COMMAND BUFFER ADDRESS
1768            000014          PCBSTP  ==      12.             ;STEP TO NEXT SLOT
1769            000044          PCB3SP  ==      36.             ;3 STEPS
1770
1771                            ; 6.3   TMSCP DRIVER BUFFER OFFSETS
1772            000002          HIADDR  ==       2.             ;DESCRIPTOR ADDRESS OFFSET
1773            177777          CONID   ==       1.             ;COMMAND/RESPONSE CONNECTION TYPE I.D.
1774            177776          CRD     ==      -2.             ;COMMAND/RESPONSE CREDIT LIMIT OFFSET
1775            177774          MSGLEN  ==      -4.             ;COMMAND/RESPONSE MESSAGE LENGTH
1776            000104          DRBSTP  ==      68.             ;RESPONSE PACKET SIZE (WORDS)
1777            000050          DCBSTP  ==      40.             ;COMMAND PACKET SIZE (WORDS)
1778            000170          DCB3SP  ==      120.            ;3 COMMAND PACKET SIZE
1779            000004          DSPSTP  ==      4.              ;PORT DESCRIPTOR SIZE
1780
1781                            ; 6.4   DRIVE STATUS CODES
1782
1783            000026          ECCDC   ==      26      ;ECC CORRECTION ON DATA BLOCK
1784            000030          ECCTC   ==      30      ;ECC CORRECTION ON A TAPE MARK
1785            000036          ECCBC   ==      36      ;CRC ERROR ON ECC BLOCK
1786
```

```
1788
1789                                    ; 6.5   LUN TABLE OFFSETS
1790    000000                          TKIP    ==      0       ;IP REGISTER ADDRESS
1791    000002                          TKSA    ==      2       ;SA REGISTER ADDRESS
1792    000004                          TKUNIT  ==      4       ;TMSCP DEVICE UNIT NUMBER
1793
1794    000006                          CMDSEQ  ==      6       ;COMMAND REFERENCE NUMBER
1795    000010                          SLTUSE  ==     10       ;BIT MAP OF RESPONSES RECEIVED
1796    000012                          CMDSSV  ==     12       ;COMMAND DESCRIPTOR
1797    000014                          CNUSAV  ==     14       ;NEW COMMAND BUFFER POINTER
1798    000016                          COLSAV  ==     16       ;OLD COMMAND BUFFER POINTER
1799    000020                          RNUSAV  ==     20       ;NEW RESPONSE BUFFER POINTER
1800    000022                          ROLSAV  ==     22       ;OLD RESPONSE BUFFER POINTER
1801
1802    000024                          PATSAV  ==     24       ;DATA PATTERN
1803    000026                          LUNFLG  ==     26       ;INITIALIZATION FLAG
1804    000030                          LEOTFL  ==     30       ;UNIT LOGICAL END OF TAPE FLAG
1805    000032                          UNDROP  ==     32       ;UNIT DROP COUNT
1806    000034                          OBJFDL  ==     34       ;OBJECT COUNT LOW ORDER
1807    000036                          OBJFDH  ==     36       ;OBJECT COUNT HIGH ORDER
1808
1809    000040                          GSTEWR  ==     40       ;WRITE STATUS ERROR IN GCR
1810    000042                          GSTERD  ==     42       ;READ STATUS ERROR IN GCR
1811    000044                          GSTEUA  ==     44       ;UNIT ACCESS STATUS ERROR IN GCR
1812    000046                          GSFTWR  ==     46       ;SOFT WRITE ERROR IN GCR
1813    000050                          GSFTRD  ==     50       ;SOFT READ ERROR IN GCR
1814    000052                          GHRDWR  ==     52       ;HARD WRITE ERROR IN GCR
1815    000054                          GHRDRD  ==     54       ;HARD READ ERROR IN GCR
1816    000056                          GHRDUA  ==     56       ;HARD UNIT ACCESS IN GCR
1817    000060                          GMEDER  ==     60       ;WRITE MEDIA ERROR IN GCR
1818    000062                          GDCERR  ==     62       ;DATA COMPARE ERROR IN GCR
1819    000064                          GOTHWR  ==     64       ;OTHER WRITE ERRORS IN GCR
1820    000066                          GOTHRD  ==     66       ;OTHER READ ERRORS IN GCR
1821    000070                          GOTHUA  ==     70       ;OTHER UNIT ACCESS ERRORS IN GCR
1822    000072                          GCRDRP  ==     72       ;TIMES UNIT WAS DROPPED IN GCR
1823    000074                          GNOERR  ==     74       ;NO ERROR
1824
1825    000076                          PSTEWR  ==     76       ;WRITE STATUS ERROR IN PE
1826    000100                          PSTERD  ==    100       ;READ STATUS ERROR IN PE
1827    000102                          PSTEUA  ==    102       ;UNIT ACCESS STATUS ERROR IN PE
1828    000104                          PSFTWR  ==    104       ;SOFT WRITE ERROR IN PE
1829    000106                          PSFTRD  ==    106       ;SOFT READ ERROR IN PE
1830    000110                          PHRDWR  ==    110       ;HARD WRITE ERROR IN PE
1831    000112                          PHRDRD  ==    112       ;HARD READ ERROR IN PE
1832    000114                          PHRDUA  ==    114       ;HARD UNIT ACCESS IN PE
1833    000116                          PMEDER  ==    116       ;WRITE MEDIA ERROR IN PE
1834    000120                          PDCERR  ==    120       ;DATA COMPARE ERROR IN PE
1835    000122                          POTHWR  ==    122       ;OTHER WRITE ERRORS IN PE
1836    000124                          POTHRD  ==    124       ;OTHER READ ERRORS IN PE
1837    000126                          POTHUA  ==    126       ;OTHER UNIT ACCESS ERRORS IN PE
1838    000130                          PEDRP   ==    130       ;TIMES UNIT WAS DROPPED IN PE
1839    000132                          PNOERR  ==    132       ;NO ERROR
1840
1841    000134                          GWRBY1  ==    134       ;HUNDREDS BYTES WRITTEN IN GCR
1842    000136                          GWRBY2  ==    136       ;THOUSANDS BYTES WRITTEN IN GCR
1843    000140                          GWRBY3  ==    140       ;MILLIONS BYTES WRITTEN IN GCR
1844    000142                          GWRBY4  ==    142       ;BILLIONS BYTES WRITTEN IN GCR
```

```
1845          000144                  GRDBY1  ==      144             ;HUNDREDS BYTES READ IN GCR
1846          000146                  GRDBY2  ==      146             ;THOUSANDS BYTES READ IN GCR
1847          000150                  GRDBY3  ==      150             ;MILLIONS BYTES READ IN GCR
1848          000152                  GRDBY4  ==      152             ;BILLIONS BYTES READ IN GCR
1849
1850          000154                  PWRBY1  ==      154             ;HUNDREDS BYTES WRITTEN IN PE
1851          000156                  PWRBY2  ==      156             ;THOUSANDS BYTES WRITTEN IN PE
1852          000160                  PWRBY3  ==      160             ;MILLIONS BYTES WRITTEN IN PE
1853          000162                  PWRBY4  ==      162             ;BILLIONS BYTES WRITTEN IN PE
1854          000164                  PRDBY1  ==      164             ;HUNDREDS BYTES READ IN PE
1855          000166                  PRDBY2  ==      166             ;THOUSANDS BYTES READ IN PE
1856          000170                  PRDBY3  ==      170             ;MILLIONS BYTES READ IN PE
1857          000172                  PRDBY4  ==      172             ;BILLIONS BYTES READ IN PE
1858
1859          000174                  SED1    ==      174             ;PRIME RANDOM GENERATOR SEED
1860          000176                  SED2    ==      176             ;PRIME RANDOM GENERATOR SEED
1861          000200                  SED3    ==      200             ;PRIME RANDOM GENERATOR SEED
1862          000202                  SEED1   ==      202             ;RANDOM GENERATOR SEED
1863          000204                  SEED2   ==      204             ;RANDOM GENERATOR SEED
1864          000206                  SEED3   ==      206             ;RANDOM GENERATOR SEED
1865
1866          000210                  URSPBF  ==      210             ;START OF THIS UNITS RESPONSE BUFFER
1867          000212                  URBEND  ==      212             ;END OF THIS UNITS RESPONSE BUFFER
1868          000214                  URDSRG  ==      214             ;START OF THIS UNITS RESPONSE DESCRIPTOR RING
1869          000216                  URDEND  ==      216             ;END OF THIS UNITS RESPONSE DESCRIPTOR RING
1870          000220                  UCDSRG  ==      220             ;START OF THIS UNITS COMMAND DESCRIPTOR RING
1871          000222                  UCDEND  ==      222             ;END OF THIS UNITS COMMAND DESCRIPTOR RING
1872
1873          000224                  LUNSTP  ==      224             ;OFFSET TO NEXT LUN BLOCK
```

```
1875                                    ;/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\
1876                                    ; 7.0   PROGRAM PRIMITIVES
1877                                    ;/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\
1878
1879                                    ; 7.1   COMMAND PRIMITIVE LITERALS
1880            000000                  NUL     ==      000             ;NULL
1881            000010                  RD      ==      010             ;READ
1882            000011                  RDR     ==      011             ;READ REVERSE
1883            000020                  WR      ==      020             ;WRITE
1884            000030                  CMP     ==      030             ;COMPARE HOST DATA
1885            000031                  CMR     ==      031             ;COMPARE HOST DATA REVERSE
1886            000040                  ACC     ==      040             ;ACCESS
1887            000041                  ACR     ==      041             ;ACCESS REVERSE
1888            000050                  SPC     ==      050             ;SPACE RECORDS
1889            000051                  SCR     ==      051             ;SPACE RECORDS REVERSE
1890            000052                  SCD     ==      052             ;SPACE TO LEOT
1891            000060                  SKP     ==      060             ;SKIP TAPE MARKS
1892            000061                  SKR     ==      061             ;SKIP TAPE MARKS REVERSE
1893            000062                  SKD     ==      062             ;SKIP TO LEOT
1894            000070                  SPO     ==      070             ;SPACE OBJECTS
1895            000071                  SPR     ==      071             ;SPACE OBJECTS REVERSE
1896            000100                  WTM     ==      100             ;WRITE TAPE MARKS
1897            000110                  ERS     ==      110             ;ERASE
1898            000113                  ERI     ==      113             ;ERASE IMMEDIATE
1899            000120                  ERG     ==      120             ;ERASE GAPS
1900            000130                  AVL     ==      130             ;AVAILABLE
1901            000134                  AVU     ==      134             ;AVAILABLE UNLOAD
1902            000140                  ONL     ==      140             ;ONLINE
1903            000150                  SUC     ==      150             ;SET UNIT CHARACTERISTICS
1904            000155                  SUW     ==      155             ;SET UNIT CHARA. W/WRITE PROTECT
1905            000160                  REW     ==      160             ;REWIND
1906            000163                  RWI     ==      163             ;REWIND IMMEDIATE
1907            000170                  INT     ==      170             ;INITIALIZATION
1908            000200                  ABO     ==      200             ;ABORT
1909            000210                  GCS     ==      210             ;GET COMMAND STATUS
1910            000220                  GUS     ==      220             ;GET UNIT STATUS
1911            000230                  SCC     ==      230             ;SET CONTROLLER CHARACTERISTICS
1912
1913                                    ; 7.2   COMMAND PRIMITIVE MODIFIER LITERALS
1914            000001                  REVBIT  ==      1               ;REVERSE MODIFIER
1915            000002                  EOTBIT  ==      2               ;DETECT LEOT MODIFIER
1916            000003                  IMMBIT  ==      3               ;IMMEDIATE MODIFIER
1917            000004                  UNLBIT  ==      4               ;UNLOAD MOIFIER
1918            000005                  WPRBIT  ==      5               ;WRITE PROTECT MODIFIER
1919
```

```
                                .SBTTL  GLOBAL DATA SECTION

                                ;/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\
                                ; 8.0 PROGRAM AND DRIVER DATA STRUCTURES
                                ;/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\

                                ; 8.1 UNIT LUN BLOCKS
002322  000000  000000  000000  LUN0::           .WORD   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
002400  000000  000000  000000                            0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
002456  000000  000000  000000                            0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
002532  004216                                            DRSPB0
002534  005256                                            DRBEN0
002536  010416                                            RDSRG0
002540  010456                                            RDREN0
002542  010456                                            CDSRG0
002544  010476                                            CDREN0

002546  000000  000000  000000  LUN1::           .WORD   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
002624  000000  000000  000000                            0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
002702  000000  000000  000000                            0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
002756  005256                                            DRSPB1
002760  006316                                            DRBEN1
002762  010502                                            RDSRG1
002764  010542                                            RDREN1
002766  010542                                            CDSRG1
002770  010562                                            CDREN1

002772  000000  000000  000000  LUN2::           .WORD   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
003050  000000  000000  000000                            0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
003126  000000  000000  000000                            0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
003202  006316                                            DRSPB2
003204  007356                                            DRBEN2
003206  010566                                            RDSRG2
003210  010626                                            RDREN2
003212  010626                                            CDSRG2
003214  010646                                            CDREN2

003216  000000  000000  000000  LUN3::           .WORD   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
003274  000000  000000  000000                            0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
003352  000000  000000  000000                            0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
003426  007356                                            DRSPB3
003430  010416                                            DRBEN3
003432  010652                                            RDSRG3
003434  010712                                            RDREN3
003436  010712                                            CDSRG3
003440  010732                                            CDREN3


                                ; 8.2 PROGRAM COMMAND BUFFERS
003442                          PCMDBF::        .BLKW   24.     ;PROGRAM COMMAND RING
        003522                  PCBEND==.                       ;PROGRAM COMMAND RING END
003522                          DUMPKT::        .BLKW   2.      ;PROGRAM DUMMY COMMAND PACKET
003526                          DRINUS::        .BLKW   4.      ;DRIVE IN USE TABLE

                                ; 8.3 PROGRAM VARIABLES
003536  000000                  CMDCNT::        .WORD   0       ;COMMAND COUNT
003540  000000                  RSPCNT::        .WORD   0       ;RESPONSE COUNT
```

GLOBAL DATA SECTION

```
1978 003542  000000  CCTSAV::    .WORD  0    ;COMMAND COUNT SAVE
1979 003544  000000  SEXCNT::    .WORD  0    ;SERIOUS EXCEPTION COUNT
1980 003546  000000  COUNT::     .WORD  0    ;COMMAND LOOP COUNTER
1981 003550  000000  TEMP::      .WORD  0    ;TEMPORARY STORE
1982 003552  000000  RESPON::    .WORD  0    ;RESPONSE STATUS
1983 003554  000000  BRCNT::     .WORD  0    ;BRANCH COUNTER
1984 003556  000000  HNDLRP::    .WORD  0    ;NUMBER OF RESPONSES
1985 003560  000000  MRETRY::    .WORD  0    ;MANUAL RETRY COUNTER
1986 003562  000000  MANCNT::    .WORD  0    ;NUMBER OF ACTUAL WRITE/READ MANUAL RETRIES
1987 003564  000000  ARETRY::    .WORD  0    ;AUTO RETRY COUNTER
1988 003566  000000  AUTCNT::    .WORD  0    ;NUMBER OF ACTUAL WRITE/READ AUTO RETRIES
1989 003570  000000  SOFTER::    .WORD  0    ;SOFT ERROR COUNT
1990
1991 003572  000000  RESP::      .WORD  0    ;DRIVER RESPONSE COUNT
1992 003574  000000  BYTES::     .WORD  0    ;BYTE COUNT
1993 003576  000000  ITERS::     .WORD  0    ;ITERATION COUNT
1994 003600  000000  BUFADR::    .WORD  0    ;COMMAND BUFFER ADDRESS
1995 003602  000000  SUBCNT::    .WORD  0    ;SUB-ITERATION COUNT FOR DATA COMPARES
1996
1997 003604  000000  RANWRD::    .WORD  0    ;USED BY RANGEN
1998 003606  000000  RAN1::      .WORD  0    ;SEED WORK LOCATION
1999 003610  000000  RAN2::      .WORD  0    ;SEED WORK LOCATION
2000 003612  000000  RAN3::      .WORD  0    ;SEED WORK LOCATION
2001
2002 003614  000000  SAVDIF::    .WORD  0    ;COMMAND AND RESPONSE COUNT DIFFERENCE
2003 003616  000000  TSTMSK::    .WORD  0    ;TEST LOAD WITH ACCEPTABLE ERROR CODES
2004 003620  000000  WRKMSK::    .WORD  0    ;USED BY ERROR DECODE
2005
2006 003622  000000  CMPERR::    .WORD  0    ;NUMBER OF BYTES IN ERROR
2007 003624          BYTADD::    .BLKW  10.  ;SAVE TABLE FOR BYTE IN ERROR ADDRESS
2008         003650  TBLEND == .             ;END OF BYTE ADDRESS TABLE
2009 003650          DATBL::     .BLKW  10.  ;SAVE TABLE FOR BYTE IN ERROR DATA
2010 003674  000000  PCFLAG::    .WORD  0    ;PROGRAM CONTROL FLAGS
2011
2012 003676  000000  OBJECT::    .WORD  0    ;OBJECT COUNTER FOR TEST 2
2013 003700  000000  PASCNT::    .WORD  0    ;PASS COUNTER
2014 003702  000000  PASS1::     .WORD  0    ;"1ST PASS OF TEST" FLAG
2015 003704  000000  UDROP::     .WORD  0    ;NUMBER OF DROPPED UNITS
2016 003706  000000  UEOT::      .WORD  0    ;COUNT OF UNITS AT EOT
2017
2018 003710  000000  R8::        .WORD  0    ;USED FOR TEMP STORAGE
2019 003712  000000  R9::        .WORD  0    ;USED FOR TEMP STORAGE
2020 003714  000000  R10::       .WORD  0    ;USED FOR TEMP STORAGE
2021 003716  000000  R11::       .WORD  0    ;USED FOR TEMP STORAGE
2022 003720  000000  R12::       .WORD  0    ;USED FOR TEMP STORAGE
2023 003722  000000  R13::       .WORD  0    ;USED FOR TEMP STORAGE
2024
2025 003724  000000  SECRNS::    .WORD  0    ;SERIOUS EXCEPTION CMD REF #
2026 003726  000000  RECCNT::    .WORD  0    ;NUMBER OF RECORDS
2027 003730  000000  TMCNT::     .WORD  0    .NUMBER OF TAPE MARKS
2028
2029 003732  000004  FORMAT::    .WORD  4    ;DEFAULT TAPE FORMAT CODE (GCR)
2030 003734  000000  INFORM::    .WORD  0    ;INITIAL TEST FORMAT
2031 003736  000000  EVENT::     .WORD  0    ;EVENT CODE STORAGE
2032 003740  000000  R3SAVE::    .WORD  0    ;SAVE LOCATION FOR R3
2033 003742  000000  R4SAVE::    .WORD  0    ;SAVE LOCATION FOR R4
2034 003744  000000  CMDSAV::    .WORD  0    ;SAVE LOCATION FOR CURRENT COMMAND
```

GLOBAL DATA SECTION

```
 2035 003746  000000                BYTSAV::        .WORD   0       ;SAVE LOCATION FOR ORIGINAL BYTE COUNT
 2036 003750     000                DAYS::          .BYTE   0       ;NUMBER OF DAYS IN RUN
 2037                               .EVEN
```

```
 2039                                   ;/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\
 2040                                   ; 9.0 TMSCP CLASS AND PORT DRIVER DATA STRUCTURES
 2041                                   ;/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\/*\
 2042
 2043                                   ; 9.1 CLASS DRIVER BUFFERS
 2044 003752                            CMDBF1::        .BLKW   2.      ;DRIVE COMMAND BUFFER 1
 2045 003756                            DCMDBF::        .BLKW   18.     ;DRIVER COMMAND BUFFER
 2046 004022                            CMDBF2::        .BLKW   20.     ;DRIVE COMMAND BUFFER 2
 2047 004072                            CMDBF3::        .BLKW   20.     ;DRIVE COMMAND BUFFER 3
 2048 004142                            CMDBF4::        .BLKW   20.     ;DRIVE COMMAND BUFFER 4
 2049
 2050 004212                            RSPBF0::        .BLKW   2.      ;TOP 4 LOCATIONS OF RESPONSE BUFFER UNIT 0
 2051 004216                            DCBEND::                        ;END OF COMMAND BUFFER
 2052 004216                            DRSPB0::        .BLKW   270.    ;DRIVER RESPONSE BUFFER UNIT 0
 2053
 2054 005252                            RSPBF1::        .BLKW   2.      ;TOP 4 LOCATIONS OF RESPONSE BUFFER UNIT 1
 2055 005256                            DRBEN0::                        ;END OF RESPONSE BUFFER UNIT 0
 2056 005256                            DRSPB1::        .BLKW   270.    ;DRIVER RESPONSE BUFFER UNIT 1
 2057
 2058 006312                            RSPBF2::        .BLKW   2.      ;TOP 4 LOCATIONS OF RESPONSE BUFFER UNIT 2
 2059 006316                            DRBEN1::                        ;END OF RESPONSE BUFFER UNIT 1
 2060 006316                            DRSPB2::        .BLKW   270.    ;DRIVER RESPONSE BUFFER UNIT 2
 2061
 2062 007352                            RSPBF3::        .BLKW   2.      ;TOP 4 LOCATIONS OF RESPONSE BUFFER UNIT 3
 2063 007356                            DRBEN2::                        ;END OF RESPONSE BUFFER UNIT 2
 2064 007356                            DRSPB3::        .BLKW   270.    ;DRIVER RESPONSE BUFFER UNIT 3
 2065
```

```
 2067                                   ; 9.2 U/Q PORT DESCRIPTOR RINGS
 2068 010412                            DSRNG0::        .BLKW   2.      ;DESCRIPTOR RING UNIT 0
 2069 010416                            DRBEN3::                        ;END OF RESPONSE BUFFER UNIT 3
 2070 010416                            RDSRG0::        .BLKW   16.     ;RESPONSE DESCRIPTOR RING UNIT 0
 2071 010456                            RDREN0::                        ;END OF RESPONCE DESCRIPTOR RING UNIT 0
 2072 010456                            CDSRG0::        .BLKW   8.      ;COMMAND DESCRIPTOR RING UNIT 0
 2073 010476                            CDREN0::                        ;END OF COMMAND DESCRIPTOR RING UNIT 0
 2074
 2075 010476                            DSRNG1::        .BLKW   2.      ;DESCRIPTOR RING UNIT 1
 2076 010502                            RDSRG1::        .BLKW   16.     ;RESPONSE DESCRIPTOR RING UNIT 1
 2077 010542                            RDREN1::                        ;END OF RESPONCE DESCRIPTOR RING UNIT 1
 2078 010542                            CDSRG1::        .BLKW   8.      ;COMMAND DESCRIPTOR RING UNIT 1
 2079 010562                            CDPEN1::                        ;END OF COMMAND DESCRIPTOR RING UNIT 1
 2080
 2081 010562                            DSRNG2::        .BLKW   2.      ;DESCRIPTOR RING UNIT 2
 2082 010566                            RDSRG2::        .BLKW   16.     ;RESPONSE DESCRIPTOR RING UNIT 2
 2083 010626                            RDREN2::                        ;END OF RESPONCE DESCRIPTOR RING UNIT 2
 2084 010626                            CDSRG2::        .BLKW   8.      ;COMMAND DESCRIPTOR RING UNIT 2
 2085 010646                            CDREN2::                        ;END OF COMMAND DESCRIPTOR RING UNIT 2
 2086
 2087 010646                            DSRNG3::        .BLKW   2.      ;DESCRIPTOR RING UNIT 3
 2088 010652                            RDSRG3::        .BLKW   16.     ;RESPONSE DESCRIPTOR RING UNIT 3
 2089 010712                            RDREN3::                        ;END OF RESPONCE DESCRIPTOR RING UNIT 3
 2090 010712                            CDSRG3::        .BLKW   8.      ;COMMAND DESCRIPTOR RING UNIT 3
 2091 010732                            CDREN3::                        ;END OF COMMAND DESCRIPTOR RING UNIT 3
 2092
 2093                                   ; 9.3 CLASS AND PORT DRIVER VARIABLES
 2094 010732  000000                    IOSTAT::        .WORD   0       ;I/O STATUS
 2095 010734  177777                    CMSTSV::        .WORD   -1      ;COMMAND STATUS FROM GCS MODE
 2096 010736  000000                    GCSREF::        .WORD   0       ;GCS COMMAND REFERENCE NUMBER
 2097 010740  000000                    CNTHI::         .WORD   0       ;VALUE OF THE HIGH TIMEOUT
 2098 010742  000000                    TIMER::         .WORD   0       ;TIMER VALUE
 2099 010744  000000                    LOOPS::         .WORD   0       ;
 2100 010746  000120                    CNTFLG::        .WORD   CF.THS!CF.MSC   ;CONTROLLER FLAGS(ENABLE THIS HOSTS
 2101                                                                   ;AND MISCELLANEOUS ERROR LOG MESSAGES)
 2102 010750  000000                    PCKSIZ::        .WORD   0       ;PACKET SIZE IN BYTES
 2103 010752  000000                    SAERR::         .WORD   0       ;SA REGISTER SAVE ON ERROR
 2104 010754     000                    MINLIM::        .BYTE   0       ;MINIMUM REQUIRED CREDIT LIMIT
 2105 010755     004                    CRDLIM::        .BYTE   4       ;DRIVER CREDIT LIMIT
 2106                                   .EVEN
 2107
```

```
2109                                    ; I/O STATUS ERROR INFORMATION TABLE
2110
2111 010756     001                    IOERTB::.BYTE   DEVFAT          ; GET COMMAND STATUS FAILED
2112 010757     064                             .BYTE   GOTHWR
2113 010760  000001                             .WORD   1
2114 010762  012244                             .WORD   CMLSER
2115 010764  013176                             .WORD   DEVERR
2116
2117 010766     001                             .BYTE   DEVFAT          ; CONTROLLER HUNG
2118 010767     064                             .BYTE   GOTHWR
2119 010770  000002                             .WORD   2
2120 010772  012277                             .WORD   HUNGER
2121 010774  013176                             .WORD   DEVERR
2122
2123 010776     001                             .BYTE   DEVFAT          ; PORT DETECTED ERROR
2124 010777     064                             .BYTE   GOTHWR
2125 011000  000003                             .WORD   3
2126 011002  012317                             .WORD   PORTER
2127 011004  013176                             .WORD   DEVERR
2128
2129 011006     001                             .BYTE   DEVFAT          ; PROGRAM DETECTED COMMAND TIMEOUT
2130 011007     064                             .BYTE   GOTHWR
2131 011010  000004                             .WORD   4
2132 011012  012343                             .WORD   TIMERR
2133 011014  013176                             .WORD   DEVERR
2134
2135 011016     001                             .BYTE   DEVFAT          ; COMMAND SEQUENCE ERROR
2136 011017     064                             .BYTE   GOTHWR
2137 011020  000005                              WORD   5
2138 011022  012400                             .WORD   SEQER
2139 011024  013176                             .WORD   DEVERR
2140
2141 011026     001                             .BYTE   DEVFAT          ; ERROR DETECTED DURING INIT
2142 011027     064                             .BYTE   GOTHWR
2143 011030  000006                             .WORD   6
2144 011032  012431                             .WORD   INITER
2145 011034  013176                             .WORD   DEVERR
2146
2147                                    ; PROGRAM DETECTED ERROR INFORMATION TABLE
2148
2149 011036     001                    CMDT::   .BYTE   DEVFAT          ;INVALID COMMAND ISSUED
2150 011037     040                             .BYTE   GSTEWR
2151 011040  000007                             .WORD   7.
2152 011042  011506                             .WORD   CMDER
2153 011044  013176                             .WORD   DEVERR
2154
2155 011046     001                    ABOT::   .BYTE   DEVFAT          ;COMMAND ABORTED
2156 011047     040                             .BYTE   GSTEWR
2157 011050  000010                             .WORD   8.
2158 011052  011531                             .WORD   ABOER
2159 011054  013176                             .WORD   DEVERR
2160
2161 011056     001                    OFLT::   .BYTE   DEVFAT          ;UNIT OFFLINE
2162 011057     040                             .BYTE   GSTEWR
2163 011060  000011                             .WORD   9.
2164 011062  011545                             .WORD   OFLER
2165 011064  013176                             .WORD   DEVERR
```

GLOBAL DATA SECTION

```
2166
2167 011066     001                AVLT::  .BYTE   DEVFAT          ;UNIT AVAILABLE ERROR
2168 011067     040                        .BYTE   GSTEWR
2169 011070  000012                        .WORD   10.
2170 011072  011562                        .WORD   AVLER
2171 011074  013176                        .WORD   DEVERR
2172
2173 011076     001                IVST1:: .BYTE   DEVFAT          ;INVALID STATUS RETURNED
2174 011077     040                        .BYTE   GSTEWR
2175 011100  000013                        .WORD   11.
2176 011102  012166                        .WORD   IVSER
2177 011104  013176                        .WORD   DEVERR
2178
2179 011106     001                WPRT::  .BYTE   DEVFAT          ;UNIT WRITE PROTECTED
2180 011107     040                        .BYTE   GSTEWR
2181 011110  000014                        .WORD   12.
2182 011112  011607                        .WORD   WPRER
2183 011114  013176                        .WORD   DEVERR
2184
2185 011116     002                CMPT::  .BYTE   HARD            ;DATA COMPARE ERROR
2186 011117     062                        .BYTE   GDCERR
2187 011120  000015                        .WORD   13.
2188 011122  012515                        .WORD   CMPER
2189 011124  013176                        .WORD   DEVERR
2190
2191 011126     001                HDATT:: .BYTE   DEVFAT          ;HARD DATA ERROR
2192 011127     052                        .BYTE   GHRDWR
2193 011130  000016                        .WORD   14.
2194 011132  011634                        .WORD   HDATER
2195 011134  013176                        .WORD   DEVERR
2196
2197 011136     001                HSTT::  .BYTE   DEVFAT          ;HOST DETECTED TIMEOUT
2198 011137     064                        .BYTE   GOTHWR
2199 011140  000017                        .WORD   15.
2200 011142  012216                        .WORD   HSTER
2201 011144  013176                        .WORD   DEVERR
2202
2203 011146     001                CNTT::  .BYTE   DEVFAT          ;CONTROLLER ERROR
2204 011147     064                        .BYTE   GOTHWR
2205 011150  000020                        .WORD   16.
2206 011152  011725                        .WORD   CNTER
2207 011154  013176                        .WORD   DEVERR
2208
2209 011156     001                DRVT::  .BYTE   DEVFAT          ;DRIVE ERROR
2210 011157     064                        .BYTE   GOTHWR
2211 011160  000021                        .WORD   17.
2212 011162  011746                        .WORD   DRVER
2213 011164  013176                        .WORD   DEVERR
2214
2215 011166     001                FMTT::  .BYTE   DEVFAT          ;FORMATTER ERROR
2216 011167     064                        .BYTE   GOTHWR
2217 011170  000022                        .WORD   18.
2218 011172  011762                        .WORD   FMTER
2219 011174  013176                        .WORD   DEVERR
2220
2221 011176     001                BOTT::  .BYTE   DEVFAT          ;UNEXPECTED BOT ENCOUNTERED
2222 011177     040                        .BYTE   GSTEWR
```

GLOBAL DATA SECTION

```
2223 011200  000023                    .WORD   19.
2224 011202  012002                    .WORD   BOTER
2225 011204  013176                    .WORD   DEVERR
2226
2227 011206     001            TMT::   .BYTE   DEVFAT          ;UNEXPECTED TAPE MARK ENCOUNTERED
2228 011207     040                    .BYTE   GSTEWR
2229 011210  000024                    .WORD   20.
2230 011212  012022                    .WORD   TMER
2231 011214  013176                    .WORD   DEVERR
2232
2233 011216     001            IVST2:: .BYTE   DEVFAT          ;INVALID STATS RECEIVED
2234 011217     040                    .BYTE   GSTEWR
2235 011220  000025                    .WORD   21.
2236 011222  012166                    .WORD   IVSER
2237 011224  013176                    .WORD   DEVERR
2238
2239 011226     001            RDTT::  .BYTE   DEVFAT          ;DATA RECORD TRUNCATED
2240 011227     040                    .BYTE   GSTEWR
2241 011230  000026                    .WORD   22.
2242 011232  012050                    .WORD   RDTER
2243 011234  013176                     WORD   DEVERR
2244
2245 011236     001            POLT::  .BYTE   DEVFAT          ;TAPE POSITION LOST
2246 011237     040                    .BYTE   GSTEWR
2247 011240  000027                    .WORD   23.
2248 011242  012076                    .WORD   POLER
2249 011244  013176                    .WORD   DEVERR
2250
2251 011246     001            SEXT::  .BYTE   DEVFAT          ;SERIOUS EXCEPTION
2252 011247     040                    .BYTE   GSTEWR
2253 011250  000030                    .WORD   24.
2254 011252  012114                    .WORD   SEXER
2255 011254  013176                    .WORD   DEVERR
2256
2257 011256     001            LEDT::  .BYTE   DEVFAT          ;LEOT ENCOUNTERED
2258 011257     040                    .BYTE   GSTEWR
2259 011260  000031                    .WORD   25.
2260 011262  012136                    .WORD   LEDER
2261 011264  013176                    .WORD   DEVERR
2262
2263 011266     001            IVST3:: .BYTE   DEVFAT          ;INVALID STATUS RETURNED
2264 011267     040                    .BYTE   GSTEWR
2265 011270  000032                    .WORD   26.
2266 011272  012166                    .WORD   IVSER
2267 011274  013176                    .WORD   DEVERR
2268
2269 011276     002            DCMPT:: .BYTE   HARD            ;DATA COMPARE ERROR
2270 011277     062                    .BYTE   GDCERR
2271 011300  000033                    .WORD   27.
2272 011302  012540                    .WORD   DCMPER
2273 011304  013176                    .WORD   DEVERR
2274
2275 011306     002            RLST::  .BYTE   HARD            ;RECORD LENGTH SHORT ERROR
2276 011307     040                    .BYTE   GSTEWR
2277 011310  000034                    .WORD   28.
2278 011312  012452                    .WORD   RLSER
2279 011314  013176                    .WORD   DEVERR
```

```
2280
2281 011316  003                SDATT:: .BYTE   SOFT            ;SOFT DATA ERROR
2282 011317  046                        .BYTE   GSFTWR
2283 011320  000035                     .WORD   29.
2284 011322  011654                     .WORD   SDATER
2285 011324  013176                     .WORD   DEVERR
2286
2287                            ;ERROR LOG ERROR TABLES
2288
2289 011326                     CNTERL:
2290 011326  002                        .BYTE   HARD            ;CONTROLLER ERROR LOG
2291 011327  040                        .BYTE   GSTEWR
2292 011330  000036                     .WORD   30.
2293 011332  012622                     .WORD   CNTEL
2294 011334  014424                     .WORD   ERLGER
2295
2296 011336                     BADERL:
2297 011336  002                        .BYTE   HARD            ;HOST MEMORY ACCSESS ERROR LOG
2298 011337  040                        .BYTE   GSTEWR
2299 011340  000037                     .WORD   31.
2300 011342  012647                     .WORD   BADEL
2301 011344  014424                     .WORD   ERLGER
2302
2303 011346                     TPEERL:
2304 011346  003                        .BYTE   SOFT            ;TAPE TRANSFER ERROR LOG
2305 011347  074                        .BYTE   GNOERR
2306 011350  000040                     .WORD   32.
2307 011352  012572                     .WORD   TPEEL
2308 011354  014424                     .WORD   ERLGER
2309
2310 011356                     UNKERL:
2311 011356  002                        .BYTE   HARD            ;UNKNOWN ERROR LOG
2312 011357  074                        .BYTE   GNOERR
2313 011360  000041                     .WORD   33.
2314 011362  012675                     .WORD   UNKEL
2315 011364  014424                     .WORD   ERLGER
2316
2317
```

```
                                        .SBTTL  GLOBAL TEXT SECTION

                                        ; COMMAND PRIMITIVE ASCII

011366                                  CMDASC::
011366  116  125  114                   .ASCIZ  ?NUL?           ;NULL
011372  122  104  040                   .ASCIZ  ?RD ?           ;READ
011376  127  122  124                   .ASCIZ  ?WRT?           ;WRITE
011402  103  115  120                   .ASCIZ  ?CMP?           ;COMPARE HOST DATA
011406  101  103  103                   .ASCIZ  ?ACC?           ;ACCESS
011412  123  120  103  SPCASC:          .ASCIZ  ?SPC?           ;SPACE RECORDS
011416  123  113  120                   .ASCIZ  ?SKP?           ;SKIP TAPE MARKS
011422  123  120  117                   .ASCIZ  ?SPO?           ;SPACE OBJECTS
011426  127  124  115                   .ASCIZ  ?WTM?           ;WRITE TAPE MARK
011432  105  122  123                   .ASCIZ  ?ERS?           ;ERASE
011436  105  122  107                   .ASCIZ  ?ERG?           ;ERASE GAP
011442  101  126  114                   .ASCIZ  ?AVL?           ;AVAILABLE
011446  117  116  114                   .ASCIZ  ?ONL?           ;ONLINE
011452  123  125  103                   .ASCIZ  ?SUC?           ;SET UNIT CHARACTERISTICS
011456  122  105  127                   .ASCIZ  ?REW?           ;REWIND
011462  111  116  124                   .ASCIZ  ?INT?           ;INITIALIZE
011466  101  102  117                   .ASCIZ  ?ABO?           ;ABORT
011472  107  103  123                   .ASCIZ  ?GCS?           ;GET COMMAND STATUS
011476  107  125  123                   .ASCIZ  ?GUS?           ;GET UNIT STATUS
011502  123  103  103                   .ASCIZ  ?SCC?           ;SET CONTROLLER CHARACTERISTICS
                                        .EVEN
```

```
2347                                    ;
2348                                    ; FORMAT STATEMENTS USED IN PRINT CALLS
2349                                    ;
2350
2351 011506     111     116     126    CMDER:   .ASCIZ  /INVALID CMD ISSUED/
2352 011531     103     115     104    ABOER:   .ASCIZ  /CMD ABORTED/
2353 011545     125     116     111    OFLER:   .ASCIZ  /UNIT OFFLINE/
2354 011562     125     116     111    AVLER:   .ASCIZ  /UNIT AVAILABLE ERROR/
2355 011607     125     116     111    WPRER:   .ASCIZ  /UNIT WRITE PROTECTED/
2356 011634     110     101     122    HDATER:  .ASCIZ  /HARD DATA ERROR/
2357 011654     123     117     106    SDATER:  .ASCIZ  /SOFT DATA ERROR/
2358 011674     110     117     123    BADER:   .ASCIZ  /HOST BUFFER ACCESS ERROR/
2359 011725     103     117     116    CNTER:   .ASCIZ  /CONTROLLER ERROR/
2360 011746     104     122     111    DRVER:   .ASCIZ  /DRIVE ERROR/
2361 011762     106     117     122    FMTER:   .ASCIZ  /FORMATTER ERROR/
2362 012002     102     117     124    BOTER:   .ASCIZ  /BOT ENCOUNTERED/
2363 012022     124     101     120    TMER:    .ASCIZ  /TAPE MARK ENCOUNTERED/
2364 012050     104     101     124    RDTER:   .ASCIZ  /DATA RECORD TRUNCATED/
2365 012076     120     117     123    POLER:   .ASCIZ  /POSITION LOST/
2366 012114     123     105     122    SEXER:   .ASCIZ  /SERIOUS EXCEPTION/
2367 012136     114     117     107    LEDER:   .ASCIZ  /LOGICAL EOT ENCOUNTERED/
2368 012166     111     116     126    IVSER:   .ASCIZ  /INVALID STATUS RECEIVED/
2369 012216     110     117     123    HSTER:   .ASCIZ  /HOST DETECTED TIMEOUT/
2370 012244     116     117     040    CMLSER:  .ASCIZ  /NO RESPONSE TO GCS COMMAND/
2371 012277     103     117     116    HUNGER:  .ASCIZ  /CONTROLLER HUNG/
2372 012317     120     117     122    PORTER:  .ASCIZ  /PORT-DETECTED ERROR/
2373 012343     120     122     117    TIMERR:  .ASCIZ  /PROGRAM DETECTED CMD TIMEOUT/
2374 012400     122     105     123    SEQER::  .ASCIZ  /RESPONSE OUT OF SEQUENCE/
2375 012431     120     117     122    INITER:  .ASCIZ  /PORT INIT FAILED/
2376 012452     122     105     103    RLSER:   .ASCIZ  /RECORD LENGTH SHORT/
2377 012476     123     124     101    STATER:  .ASCIZ  /STATUS MESSAGE/
2378 012515     104     101     124    CMPER:   .ASCIZ  /DATA COMPARE ERROR/
2379 012540     123     057     127    DCMPER:  .ASCIZ  ?S/W DETECTED DATA COMPARE?
2380
2381 012572     124     101     120    TPEEL:   .ASCIZ  /TAPE TRANSFER ERROR LOG/
2382 012622     103     117     116    CNTEL:   .ASCIZ  /CONTROLLER ERROR LOG/
2383 012647     110     117     123    BADEL:   .ASCIZ  /HOST MEMORY ERROR LOG/
2384 012675     125     116     113    UNKEL:   .ASCIZ  /UNKNOWN ERROR LOG FORMAT CODE/
2385
2386 012733     122     105     124    RTYEL:   .ASCIZ  /RETRY RECOVERED READ ERROR/
2387 012766     122     105     124    COREL:   .ASCIZ  /RETRY RECOVERED WRITE ERROR/
2388 013022     110     101     122    UREEL:   .ASCIZ  /HARD READ ERROR LOG/
2389 013046     110     101     122    UWEEL:   .ASCIZ  /HARD WRITE ERROR LOG/
2390 013073     104     101     124    CMPEL:   .ASCIZ  /DATA COMPARE ERROR LOG/
2391 013122     114     117     116    LGPEL:   .ASCIZ  /LONG GAP ENCOUNTERD/
2392 013146     104     122     111    DRVEL:   .ASCIZ  /DRIVE ERROR LOG/
2393                                    .EVEN
2400
2401
```

```
                                        .SBTTL  GLOBAL ERROR REPORT SECTION

                                        ;++
                                        ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX CALLS
                                        ; THAT ARE USED IN MORE THAN ONE TEST.  IT ALSO INCLUDES THE ASCII MESSAGES
                                        ; THAT ARE USED BY THE PRINTB AND PRINTX CALLS..
                                        ;--

013166                                          ERRTBL                          ;GENERIC ERROR TABLE
013166                                  L$ERRTBL::
013166  000000                          ERRTYP::       .WORD   0
013170  000000                          ERRNBR::       .WORD   0
013172  000000                          ERRMSG::       .WORD   0
013174  000000                          ERRBLK::       .WORD   0

013176                                          BGNMSG DEVERR
013176                                  DEVERR::
013176                                          PUSH    <R1,R5>                 ;SAVE R1 AND R5
013176  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
013200  010546                                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
013202  013703  003740                          MOV     R3SAVE,R3               ;RESTORE R3
013206  013704  003742                          MOV     R4SAVE,R4               ;RESTORE R4
013212  116205  000000                          MOVB    CMD(R2),R5              ;GET THE COMMAND PRIMITIVE
013216  042705  177407                          BIC     #177407,R5              ;CLEAR MODIFIERS
013222  006205                                  ASR     R5                      ;THE PRIMITIVE
013224  062705  011366                          ADD     #CMDASC,R5              ;PUT ADDRESS IN R5
013230  016237  000000  003710                  MOV     CMD(R2),R8              ;GET THE PRIMITIVE AGAIN
013236  042737  177770  003710                  BIC     #177770,R8              ;SAVE THE LAST 3 BITS
013244  001014                                  BNE     10$                     ;BRANCH IF NOT ZERO
013246                                          PRINTB  #ERR00,R5,TKUNIT(R4)
013246  016446  000004                          MOV     TKUNIT(R4),-(SP)
013252  010546                                  MOV     R5,-(SP)
013254  012746  015730                          MOV     #ERR00,-(SP)
013260  012746  000003                          MOV     #3,-(SP)
013264  010600                                  MOV     SP,R0
013266  104414                                  TRAP    C$PNTB
013270  062706  000010                          ADD     #10,SP
013274  000513                                  BR      60$                     ;GO PRINT THE REST OF THE MESSAGE
013276  022737  000001  C03710  10$:     CMP     #REVBIT,R8              ;IS IT A REVERSE ?
013304  001014                                  BNE     20$                     ;BRANCH IF NOT
013306                                          PRINTB  #ERR01,R5,TKUNIT(R4)
013306  016446  000004                          MOV     TKUNIT(R4),-(SP)
013312  010546                                  MOV     R5,-(SP)
013314  012746  016000                          MOV     #ERR01,-(SP)
013320  012746  000003                          MOV     #3,-(SP)
013324  010600                                  MOV     SP,R0
013326  104414                                  TRAP    C$PNTB
013330  062706  000010                          ADD     #10,SP
013334  000473                                  BR      60$                     ;GO PRINT THE REST OF THE MESSAGE
013336  032737  000002  003710  20$:     BIT     #EOTBIT,R8              ;IS IT A DETECT EOT ?
013344  001414                                  BEQ     30$                     ;BRANCH IF NOT
013346                                          PRINTB  #ERR02,R5,TKUNIT(R4)
013346  016446  000004                          MOV     TKUNIT(R4), (SP)
013352  010546                                  MOV     R5,-(SP)
013354  012746  016056                          MOV     #ERR02,-(SP)
013360  012746  000003                          MOV     #3,-(SP)
013364  010600                                  MOV     SP,R0
```

GLOBAL ERROR REPORT SECTION

```
        013366  104414                          TRAP    C$PNTB
        013370  062706  000010                  ADD     #10,SP
   2440 013374  000453                          BR      60$                     ;GO PRINT THE REST OF THE MESSAGE
   2441 013376  022737  000003  003710  30$:    CMP     #IMMBIT,R8              ;IS IT A IMMEDIATE ?
   2442 013404  001014                          BNE     40$                     ;BRANCH IF NOT
   2443 013406                                  PRINTB  #ERR03,R5,TKUNIT(R4)
        013406  016446  000004                  MOV     TKUNIT(R4), (SP)
        013412  010546                          MOV     R5,-(SP)
        013414  012746  016135                  MOV     #ERR03,-(SP)
        013420  012746  000003                  MOV     #3,-(SP)
        013424  010600                          MOV     SP,R0
        013426  104414                          TRAP    C$PNTB
        013430  062706  000010                  ADD     #10,SP
   2444 013434  000433                          BR      60$                     ;GO PRINT THE REST OF THE MESSAGE
   2445 013436  022737  000004  003710  40$:    CMP     #UNLBIT,R8              ;IS IT A UNLOAD ?
   2446 013444  001014                          BNE     50$                     ;BRANCH IF NOT
   2447 013446                                  PRINTB  #ERR04,R5,TKUNIT(R4)
        013446  016446  000004                  MOV     TKUNIT(R4),-(SP)
        013452  010546                          MOV     R5,-(SP)
        013454  012746  016213                  MOV     #ERR04,-(SP)
        013460  012746  000003                  MOV     #3,-(SP)
        013464  010600                          MOV     SP,R0
        013466  104414                          TRAP    C$PNTB
        013470  062706  000010                  ADD     #10,SP
   2448 013474  000413                          BR      60$                     ;GO PRINT THE REST OF THE MESSAGE
   2449 013476                          50$:    PRINTB  #ERR05,R5,TKUNIT(R4)
        013476  016446  000004                  MOV     TKUNIT(R4),-(SP)
        013502  010546                          MOV     R5,-(SP)
        013504  012746  016272                  MOV     #ERR05,-(SP)
        013510  012746  000003                  MOV     #3,-(SP)
        013514  010600                          MOV     SP,R0
        013516  104414                          TRAP    C$PNTB
        013520  062706  000010                  ADD     #10,SP
   2450 013524                          60$:    PRINTB  #ERR06,PASCNT,PATSAV(R4)
        013524  016446  000024                  MOV     PATSAV(R4),-(SP)
        013530  013746  003700                  MOV     PASCNT,-(SP)
        013534  012746  016351                  MOV     #ERR06,-(SP)
        013540  012746  000003                  MOV     #3,-(SP)
        013544  010600                          MOV     SP,R0
        013546  104414                          TRAP    C$PNTB
        013550  062706  000010                  ADD     #10,SP
   2451 013554  022705  011412                  CMP     #SPCASC,R5              ;IS IT A DATA TRANSFER ERROR ?
   2452 013560  101412                          BLOS    70$                     ;NO, DON'T PRINT THE BYTE COUNT
   2453 013562                                  PRINTB  #ERR07,BYTES            ;PRINT THE BYTE COUNT
        013562  013746  003574                  MOV     BYTES,-(SP)
        013566  012746  016421                  MOV     #ERR07,-(SP)
        013572  012746  000002                  MOV     #2,-(SP)
        013576  010600                          MOV     SP,R0
        013600  104414                          TRAP    C$PNTB
        013602  062706  000006                  ADD     #6,SP
   2454 013606                          70$:    PRINTB  #ERR08,OBOFFH(R2),OBOFFL(R2)
        013606  016246  000004                  MOV     OBOFFL(R2),-(SP)
        013612  016246  000006                  MOV     OBOFFH(R2),-(SP)
        013616  012746  016461                  MOV     #ERR08,-(SP)
        013622  012746  000003                  MOV     #3,-(SP)
        013626  010600                          MOV     SP,R0
        013630  104414                          TRAP    C$PNTB
```

GLOBAL ERROR REPORT SECTION

```
      013632  062706  000010                  ADD     #10,SP
 2455 013636  032764  000200  000026          BIT     #RETFLG,LUNFLG(R4)      ;ARE WE DOING RETRIES ?
 2456 013644  001412                          BEQ     80$                     ;NO, DON'T PRINT RETRY COUNT
 2457 013646                                  PRINTB  #ERR18,MANCNT           ;PRINT THE RETRY COUNT
      013646  013746  003562                  MOV     MANCNT,-(SP)
      013652  012746  017110                  MOV     #ERR18,-(SP)
      013656  012746  000002                  MOV     #2,-(SP)
      013662  010600                          MOV     SP,R0
      013664  104414                          TRAP    C$PNTB
      013666  062706  000006                  ADD     #6,SP
 2458 013672  122737  000006  010732  80$:    CMPB    #INTERR,IOSTAT          ;IS IT A PORT INIT FAILURE ?
 2459 013700  001001                          BNE     90$                     ;KEEP GOING IF IT ISN'T
 2460 013702  000404                          BR      100$                    ;GO PRINT SA CONTENTS
 2461 013704  122737  000003  010732  90$:    CMPB    #IOPDRE,IOSTAT          ;IS IT A PORT DETECTED FAILURE ?
 2462 013712  001014                          BNE     110$                    ;KEEP GOING IF IT ISN'T
 2463 013714                          100$:   PRINTB  #ERR10,SAERR            ;PRINT THE SA CONTENTS IF IT IS
      013714  013746  010752                  MOV     SAERR,-(SP)
      013720  012746  016557                  MOV     #ERR10,-(SP)
      013724  012746  000002                  MOV     #2,-(SP)
      013730  010600                          MOV     SP,R0
      013732  104414                          TRAP    C$PNTB
      013734  062706  000006                  ADD     #6,SP
 2464 013740  005037  010752                  CLR     SAERR                   ;CLEAR THE ERROR OUT OF THE LOCATION
 2465 013744  105737  010732          110$:   TSTB    IOSTAT                  ;WAS IT AN I/O ERROR ?
 2466 013750  001154                          BNE     DEVEXT                  ;GET OUT IF IT WAS
 2467 013752  032764  000200  000026          BIT     #RETFLG,LUNFLG(R4)      ;ARE WE DOING RETRIES ?
 2468 013760  001150                          BNE     DEVEXT                  ;DON'T PRINT PACKET
 2469 013762  005737  003622                  TST     CMPERR                  ;WAS IT A COMPARE ERROR ?
 2470 013766  001051                          BNE     CMPPRI                  ;GO PRINT THE ERROR DATA
 2471 013770                                  PRINTX  #ERR11
      013770  012746  016611                  MOV     #ERR11,-(SP)
      013774  012746  000001                  MOV     #1,-(SP)
      014000  010600                          MOV     SP,R0
      014002  104415                          TRAP    C$PNTX
      014004  062706  000004                  ADD     #4,SP
 2472 014010                                  PRINTX  #ERR12
      014010  012746  016637                  MOV     #ERR12,-(SP)
      014014  012746  000001                  MOV     #1,-(SP)
      014020  010600                          MOV     SP,R0
      014022  104415                          TRAP    C$PNTX
      014024  062706  000004                  ADD     #4,SP
 2473 014030  010305                          MOV     R3,R5                   ;GET POINTER TO RESPONSE PACKET
 2474 014032  010301                          MOV     R3,R1                   ;AND A SECOND COPY
 2475 014034  062701  000002                  ADD     #2,R1                   ;R1 POINT TO SECOND WORD OF PACKET
 2476 014040  005763  177774          PRIPCK: TST     MSGLEN(R3)              ;CHECK THE MESSAGE LENGTH
 2477 014044  100422                          BMI     CMPPRI                  ;GET OUT IF IT WENT NEGATIVE
 2478 014046                                  PRINTX  #ERR13,(R1),(R5)
      014046  011546                          MOV     (R5),-(SP)
      014050  011146                          MOV     (R1),-(SP)
      014052  012746  016675                  MOV     #ERR13,-(SP)
      014056  012746  000003                  MOV     #3,-(SP)
      014062  010600                          MOV     SP,R0
      014064  104415                          TRAP    C$PNTX
      014066  062706  000010                  ADD     #10,SP
 2479 014072  062701  000004                  ADD     #4,R1                   ;GET THE NEXT WORD
 2480 014076  062705  000004                  ADD     #4,R5                   ;AND AGAIN
 2481 014102  162763  000004  177774          SUB     #4,MSGLEN(R3)           ;ADJUST MESSAGE LENGTH DOWN 2 WORDS
```

GLOBAL ERROR REPORT SECTION

```
2482 014110  001353                    BNE     PRIPCK                  ;KEEP PRINTING TILL ALL DONE
2483 014112  005737  003622  CMPPRI:   TST     CMPERR                  ;WAS THIS A COMPARE ERROR ?
2484 014116  001471                    BEQ     DEVEXT                  ;GET OUT IF IT WASN'T
2485 014120                            PUSH    <R2>                    ;SAVE R2
2486 014122  012701  003624            MOV     #BYTADD,R1              ;POINT R1 TO THE BYTE ADDRESS TABLE
2487 014126  012702  003650            MOV     #DATBL,R2               ;POINT R2 TO THE WRITE DATA TABLE
2488 014132  013705  003622            MOV     CMPERR,R5               ;LET R5 = THE NUMBER OF BYTES IN ERROR
2489 014136                            PRINTX  #ERR14
     014136  012746  016726            MOV     #ERR14,-(SP)
     014142  012746  000001            MOV     #1,-(SP)
     014146  010600                    MOV     SP,R0
     014150  104415                    TRAP    C$PNTX
     014152  062706  000004            ADD     #4,SP
2490 014156                            PRINTX  #ERR15
     014156  012746  016754            MOV     #ERR15,-(SP)
     014162  012746  000001            MOV     #1,-(SP)
     014166  010600                    MOV     SP,R0
     014170  104415                    TRAP    C$PNTX
     014172  062706  000004            ADD     #4,SP
2491 014176            1$:     PRINTX  #ERR16,(R1),<B,ONE(R2)>,<B,(R2)>
     014176  005046                    CLR     -(SP)
     014200  151216                    BISB    (R2),(SP)
     014202  005046                    CLR     -(SP)
     014204  156216  000001            BISB    ONE(R2),(SP)
     014210  011146                    MOV     (R1),-(SP)
     014212  012746  017024            MOV     #ERR16,-(SP)
     014216  012746  000004            MOV     #4,-(SP)
     014222  010600                    MOV     SP,R0
     014224  104415                    TRAP    C$PNTX
     014226  062706  000012            ADD     #12,SP
2492 014232  005337  003622            DEC     CMPERR                  ;SUBTRACT 1 FROM NUMBER OF ERRORS
2493 014236  001405                    BEQ     CPRIEX                  ;GO PRINT TOTAL NUMBER IN ERROR
2494 014240  005721                    TST     (R1)+                   ;POINT R1 TO THE NEXT ADDRESS
2495 014242  005722                    TST     (R2)+                   ;POINT R2 TO THE NEXT DATA
2496 014244  022701  003650            CMP     #TBLEND,R1              ;HAVE WE PRINTED THE WHOLE TABLE ?
2497 014250  001352                    BNE     1$                      ;NO CONTINUE
2498 014252            CPRIEX: PRINTX  #ERR17,R5
     014252  010546                    MOV     R5,-(SP)
     014254  012746  017051            MOV     #ERR17,-(SP)
     014260  012746  000002            MOV     #2,-(SP)
     014264  010600                    MOV     SP,R0
     014266  104415                    TRAP    C$PNTX
     014270  062706  000006            ADD     #6,SP
2499 014274  005037  003622            CLR     CMPERR                  ;CLEAR THE ERROR COUNTER
2500 014300                            POP     <R2>
2501 014302            DEVEXT: PRINTF  #LINE
     014302  012746  020524            MOV     #LINE,-(SP)
     014306  012746  000001            MOV     #1,-(SP)
     014312  010600                    MOV     SP,R0
     014314  104417                    TRAP    C$PNTF
     014316  062706  000004            ADD     #4,SP
2502 014322  105737  002216            TSTB    CLOCK                   ;IS THE CLOCK ENABLED
2503 014326  001431                    BEQ     1$                      ;NO, THEN CAN'T PRINT TIME
2504 014330                            PRINTF  #TIME,<B,HOURS>,<B,MINUTE>,<B,SECOND>
     014330  005046                    CLR     -(SP)
     014332  153716  002221            BISB    SECOND,(SP)
     014336  005046                    CLR     -(SP)
```

GLOBAL ERROR REPORT SECTION

```
        014340  153716  002220                  BISB    MINUTE,(SP)
        014344  005046                          CLR     -(SP)
        014346  153716  002217                  BISB    HOURS,(SP)
        014352  012746  020037                  MOV     #TIME,-(SP)
        014356  012746  000004                  MOV     #4,-(SP)
        014362  010600                          MOV     SP,R0
        014364  104417                          TRAP    C$PNTF
        014366  062706  000012                  ADD     #12,SP
  2505  014372                                  PRINTF  #LINE
        014372  012746  020524                  MOV     #LINE,-(SP)
        014376  012746  000001                  MOV     #1,-(SP)
        014402  010600                          MOV     SP,R0
        014404  104417                          TRAP    C$PNTF
        014406  062706  000004                  ADD     #4,SP
  2506  014412                          1$:     POP     <R5,R1>
  2507  014416                                  EXIT    MSG
        014416  000167                          .WORD   J$JMP
        014420  000000                          .WORD   L10002-2-.
  2508  014422                                  ENDMSG
        014422                          L10002:
        014422  104423                          TRAP    C$MSG
```

GLOBAL ERROR REPORT SECTION

```
2510 014424                                        BGNMSG  ERLGER
     014424                                ERLGER::
2511 014424                                        PUSH    <R1,R5>                 ;SAVE R1 AND R5
     014424  010146                               MOV     R1,-(SP)        ;;PUSH R1 ON STACK
     014426  010546                               MOV     R5,-(SP)        ;;PUSH R5 ON STACK
2512 014430  013703  003740                       MOV     R3SAVE,R3               ;RESTORE R3
2513 014434  013704  003742                       MOV     R4SAVE,R4               ;RESTORE R4
2514 014440  113705  003716                       MOVB    R11,R5                  ;GET THE COMMAND PRIMITIVE
2515 014444  042705  177407                       BIC     #177407,R5              ;CLEAR MODIFIERS
2516 014450  006205                              ASR     R5                      ;THE PRIMITIVE
2517 014452  062705  011366                       ADD     #CMDASC,R5              ;PUT ADDRESS IN R5
2518 014456  013737  003716  003710               MOV     R11,R8                  ;GET THE PRIMITIVE AGAIN
2519 014464  042737  177770  003710               BIC     #177770,R8              ;SAVE THE LAST 3 BITS
2520 014472  001014                               BNE     5$                      ;BRANCH IF NOT ZERO
2521 014474                                        PRINTB  #ERR00,R5,TKUNIT(R4)
     014474  016446  000004                       MOV     TKUNIT(R4),-(SP)
     014500  010546                               MOV     R5,-(SP)
     014502  012746  015730                       MOV     #ERR00,-(SP)
     014506  012746  000003                       MOV     #3,-(SP)
     014512  010600                               MOV     SP,R0
     014514  104414                               TRAP    C$PNTB
     014516  062706  000010                       ADD     #10,SP
2522 014522  000512                               BR      30$                     ;GO PRINT THE REST OF THE MESSAGE
2523 014524  022737  000001  003710  5$:          CMP     #REVBIT,R8              ;IS IT A REVERSE ?
2524 014532  001013                               BNE     10$                     ;BRANCH IF NOT
2525 014534                                        PRINTB  #ERR01,R5,TKUNIT(R4)
     014534  016446  000004                       MOV     TKUNIT(R4),-(SP)
     014540  010546                               MOV     R5,-(SP)
     014542  012746  016000                       MOV     #ERR01,-(SP)
     014546  012746  000003                       MOV     #3,-(SP)
     014552  010600                               MOV     SP,R0
     014554  104414                               TRAP    C$PNTB
     014556  062706  000010                       ADD     #10,SP
2526 014562  032737  000002  003710  10$:         BIT     #EOTBIT,R8              ;IS IT A DETECT LEOT ?
2527 014570  001414                               BEQ     15$                     ;BRANCH IF NOT
2528 014572                                        PRINTB  #ERR02,R5,TKUNIT(R4)
     014572  016446  000004                       MOV     TKUNIT(R4),-(SP)
     014576  010546                               MOV     R5,-(SP)
     014600  012746  016056                       MOV     #ERR02,-(SP)
     014604  012746  000003                       MOV     #3,-(SP)
     014610  010600                               MOV     SP,R0
     014612  104414                               TRAP    C$PNTB
     014614  062706  000010                       ADD     #10,SP
2529 014620  000453                               BR      30$                     ;GO PRINT THE REST OF THE MESSAGE
2530 014622  022737  000003  003710  15$:         CMP     #IMMBIT,R8              ;IS IT A IMMEDIATE ?
2531 014630  001014                               BNE     20$                     ;BRANCH IF NOT
2532 014632                                        PRINTB  #ERR03,R5,TKUNIT(R4)
     014632  016446  000004                       MOV     TKUNIT(R4),-(SP)
     014636  010546                               MOV     R5,-(SP)
     014640  012746  016135                       MOV     #ERR03,-(SP)
     014644  012746  000003                       MOV     #3,-(SP)
     014650  010600                               MOV     SP,R0
     014652  104414                               TRAP    C$PNTB
     014654  062706  000010                       ADD     #10,SP
2533 014660  000433                               BR      30$                     ;GO PRINT THE REST OF THE MESSAGE
2534 014662  022737  000004  003710  20$:         CMP     #UNLBIT,R8              ;IS IT A UNLOAD ?
2535 014670  001014                               BNE     25$                     ;BRANCH IF NOT
```

GLOBAL ERROR REPORT SECTION

```
2536 014672                                    PRINTB  @ERR04,R5,TKUNIT(R4)
     014672  016446  000004                    MOV     TKUNIT(R4),-(SP)
     014676  010546                            MOV     R5, (SP)
     014700  012746  016213                    MOV     @ERR04,-(SP)
     014704  012746  000003                    MOV     #3,-(SP)
     014710  010600                            MOV     SP,R0
     014712  104414                            TRAP    C$PNTB
     014714  062706  000010                    ADD     #10,SP
2537 014720  000413                            BR      30$                     ;GO PRINT THE REST OF THE MESSAGE
2538 014722                            25$:    PRINTB  @ERR05 R5,TKUNIT(R4)
     014722  016446  000004                    MOV     TKUNIT(R4),-(SP)
     014726  010546                            MOV     R5,-(SP)
     014730  012746  016272                    MOV     @ERR05,-(SP)
     014734  012746  000003                    MOV     #3,-(SP)
     014740  010600                            MOV     SP,R0
     014742  104414                            TRAP    C$PNTB
     014744  062706  000010                    ADD     #10,SP
2539 014750                            30$:    PRINTB  @ERR06,PASCNT,PATSAV(R4)
     014750  016446  000024                    MOV     PATSAV(R4),-(SP)
     014754  013746  003700                    MOV     PASCNT,-(SP)
     014760  012746  016351                    MOV     @ERR06,-(SP)
     014764  012746  000003                    MOV     #3,-(SP)
     014770  010600                            MOV     SP,R0
     014772  104414                            TRAP    C$PNTB
     014774  062706  000010                    ADD     #10,SP
2540 015000  122763  000005  000010            CMPB    @FM.TPE,L.FMT(R3)       ;IS IT A TAPE TRANSFER ERROR LOG ?
2541                                           BEQ     35$                     ;YES, GO PRINT IT
2542 015006  000137  015416                    JMP     PKPRNT                  ;NO, PRINT THE ERROR LOG PACKET
2543 015012                            35$:    PRINTB  @ERR07,BYTES            ;PRINT THE BYTE COUNT
     015012  013746  003574                    MOV     BYTES,-(SP)
     015016  012746  016421                    MOV     @ERR07,-(SP)
     015022  012746  000002                    MOV     #2,-(SP)
     015026  010600                            MOV     SP,R0
     015030  104414                            TRAP    C$PNTB
     015032  062706  000006                    ADD     #6,SP
2544 015036                                    PRINTB  @ERR08,OBOFFH(R2),OBOFFL(R2)
     015036  016246  000004                    MOV     OBOFFL(R2),-(SP)
     015042  016246  000006                    MOV     OBOFFH(R2),-(SP)
     015046  012746  016461                    MOV     @ERR08,-(SP)
     015052  012746  000003                    MOV     #3,-(SP)
     015056  010600                            MOV     SP,R0
     015060  104414                            TRAP    C$PNTB
     015062  062706  000010                    ADD     #10,SP
2545 015066                                    PRINTX  @ERL00,L.PSTN+2(R3),L.PSTN(R3)
     015066  016346  000044                    MOV     L.PSTN(R3),-(SP)
     015072  016346  000046                    MOV     L.PSTN+2(R3),-(SP)
     015076  012746  017137                    MOV     @ERL00,-(SP)
     015102  012746  000003                    MOV     #3,-(SP)
     015106  010600                            MOV     SP,R0
     015110  104415                            TRAP    C$PNTX
     015112  062706  000010                    ADD     #10,SP
2546 015116                                    PRINTX  @ERL01,<B,L.TRK(R3)>,<B,L.LVL(R3)>,<B,L.RTRY(R3)>
     015116  005046                            CLR     -(SP)
     015120  156316  000043                    BISB    L.RTRY(R3),(SP)
     015124  005046                            CLR     -(SP)
     015126  156316  000042                    BISB    L.LVL(R3),(SP)
     015132  005046                            CLR     -(SP)
```

```
        015134  156316  000055                  BISB    L.TRK(R3),(SP)
        015140  012746  017174                  MOV     @ERL01,-(SP)
        015144  012746  000004                  MOV     #4,-(SP)
        015150  010600                          MOV     SP,R0
        015152  104415                          TRAP    C$PNTX
        015154  062706  000012                  ADD     #12,SP
  2547  015160                                  PRINTX  @ERL02,<B,L.LBLK(R3)>,L.PBLK(R3)
        015160  016346  000056                  MOV     L.PBLK(R3),-(SP)
        015164  005046                          CLR     -(SP)
        015166  156316  000060                  BISB    L.LBLK(R3),(SP)
        015172  012746  017272                  MOV     @ERL02,-(SP)
        015176  012746  000003                  MOV     #3,-(SP)
        015202  010600                          MOV     SP,R0
        015204  104415                          TRAP    C$PNTX
        015206  062706  000010                  ADD     #10,SP
  2548  015212                                  PRINTX  @ERL03,<B,L.DRVC(R3)>,<B,L.DFLG(R3)>
        015212  005046                          CLR     -(SP)
        015214  156316  000054                  BISB    L.DFLG(R3),(SP)
        015220  005046                          CLR     -(SP)
        015222  156316  000053                  BISB    L.DRVC(R3),(SP)
        015226  012746  017356                  MOV     @ERL03,-(SP)
        015232  012746  000003                  MOV     #3,-(SP)
        015236  010600                          MOV     SP,R0
        015240  104415                          TRAP    C$PNTX
        015242  062706  000010                  ADD     #10,SP
  2549  015246                                  PRINTX  @ERL04,L.DRVS(R3),<B,L.STS(R3)>
        015246  005046                          CLR     -(SP)
        015250  156316  000052                  BISB    L.STS(R3),(SP)
        015254  016346  000064                  MOV     L.DRVS(R3),-(SP)
        015260  012746  017445                  MOV     @ERL04,-(SP)
        015264  012746  000003                  MOV     #3,-(SP)
        015270  010600                          MOV     SP,R0
        015272  104415                          TRAP    C$PNTX
        015274  062706  000010                  ADD     #10,SP
  2550  015300                                  PRINTX  @ERL05,<B,L.CNT0(R3)>,<B,L.CNT1(R3)>
        015300  005046                          CLR     -(SP)
        015302  156316  000062                  BISB    L.CNT1(R3),(SP)
        015306  005046                          CLR     -(SP)
        015310  156316  000061                  BISB    L.CNT0(R3),(SP)
        015314  012746  017531                  MOV     @ERL05,-(SP)
        015320  012746  000003                  MOV     #3,-(SP)
        015324  010600                          MOV     SP,R0
        015326  104415                          TRAP    C$PNTX
        015330  062706  000010                  ADD     #10,SP
  2551  015334                                  PRINTX  @ERL06,<B,L.CNT2(R3)>,L.RWST(R3)
        015334  016346  000066                  MOV     L.RWST(R3),-(SP)
        015340  005046                          CLR     -(SP)
        015342  156316  000063                  BISB    L.CNT2(R3),(SP)
        015346  012746  017620                  MOV     @ERL06,-(SP)
        015352  012746  000003                  MOV     #3,-(SP)
        015356  010600                          MOV     SP,R0
        015360  104415                          TRAP    C$PNTX
        015362  062706  000010                  ADD     #10,SP
  2552  015366                                  PRINTX  @ERL07,L.OPFL(R3)
        015366  016346  000070                  MOV     L.OPFL(R3),-(SP)
        015372  012746  017704                  MOV     @ERL07,-(SP)
        015376  012746  000002                  MOV     #2,-(SP)
```

```
        015402  010600                          MOV     SP,R0
        015404  104415                          TRAP    C$PNTX
        015406  062706  000006                  ADD     #6,SP
 2553   015412  000137  015610                  JMP     MSGEXT                          ;GET OUT
 2554   015416                          PKPRNT: PRINTB   #ERR08,@#OBOFFH(R2),@#OBOFFL(R2)
        015416  016246  000004                  MOV     @#OBOFFL(R2),-(SP)
        015422  016246  000006                  MOV     @#OBOFFH(R2),-(SP)
        015426  012746  016461                  MOV     #ERR08,-(SP)
        015432  012746  000003                  MOV     #3,-(SP)
        015436  010600                          MOV     SP,R0
        015440  104414                          TRAP    C$PNTB
        015442  062706  000010                  ADD     #10,SP
 2555   015446                                  PRINTF  #LINE
        015446  012746  020524                  MOV     #LINE,-(SP)
        015452  012746  000001                  MOV     #1,-(SP)
        015456  010600                          MOV     SP,R0
        015460  104417                          TRAP    C$PNTF
        015462  062706  000004                  ADD     #4,SP
 2556   015466                                  PRINTX  #ERL08
        015466  012746  017734                  MOV     #ERL08,-(SP)
        015472  012746  000001                  MOV     #1,-(SP)
        015476  010600                          MOV     SP,R0
        015500  104415                          TRAP    C$PNTX
        015502  062706  000004                  ADD     #4,SP
 2557   015506                                  PRINTX  #ERR12
        015506  012746  016637                  MOV     #ERR12,-(SP)
        015512  012746  000001                  MOV     #1,-(SP)
        015516  010600                          MOV     SP,R0
        015520  104415                          TRAP    C$PNTX
        015522  062706  000004                  ADD     #4,SP
 2558   015526  010305                          MOV     R3,R5                           ;GET POINTER TO RESPONSE PACKET
 2559   015530  010301                          MOV     R3,R1                           ;AND A SECOND COPY
 2560   015532  062701  000002                  ADD     #2,R1                           ;R1 POINT TO SECOND WORD OF PACKET
 2561   015536  005763  177774          1$:     TST     MSGLEN(R3)                      ;ARE WE STILL POSITIVE ?
 2562   015542  100422                          BMI     MSGEXT                          ;NO, GET OUT
 2563   015544                                  PRINTX  #ERR13,(R1),(R5)
        015544  011546                          MOV     (R5),-(SP)
        015546  011146                          MOV     (R1),-(SP)
        015550  012746  016675                  MOV     #ERR13,-(SP)
        015554  012746  000003                  MOV     #3,-(SP)
        015560  010600                          MOV     SP,R0
        015562  104415                          TRAP    C$PNTX
        015564  062706  000010                  ADD     #10,SP
 2564   015570  062701  000004                  ADD     #4,R1                           ;GET THE NEXT WORD
 2565   015574  062705  000004                  ADD     #4,R5                           ;AND AGAIN
 2566   015600  162763  000004  177774          SUB     #4,MSGLEN(R3)                   ;ADJUST MESSAGE LENGTH DOWN 2 WORDS
 2567   015606  001353                          BNE     1$                              ;KEEP PRINTING TILL ALL DONE
 2568   015610                          MSGEXT: PRINTF  #LINE
        015610  012746  020524                  MOV     #LINE,-(SP)
        015614  012746  000001                  MOV     #1,-(SP)
        015620  010600                          MOV     SP,R0
        015622  104417                          TRAP    C$PNTF
        015624  062706  000004                  ADD     #4,SP
 2569   015630  105737  002216                  TSTB    CLOCK                           ;IS THE CLOCK ENABLED
 2570   015634  001431                          BEQ     1$                              ;NO, THEN CAN'T PRINT TIME
 2571   015636                                  PRINTF  #TIME,<@B,HOURS>,<@B,MINUTE>,<@B,SECOND>
        015636  005046                          CLR     -(SP)
```

```
        015640  153716  002221                  BISB    SECOND,(SP)
        015644  005046                          CLR     -(SP)
        015646  153716  002220                  BISB    MINUTE,(SP)
        015652  005046                          CLR     -(SP)
        015654  153716  002217                  BISB    HOURS,(SP)
        015660  012746  020037                  MOV     #TIME,-(SP)
        015664  012746  000004                  MOV     #4,-(SP)
        015670  010600                          MOV     SP,R0
        015672  104417                          TRAP    C$PNTF
        015674  062706  000012                  ADD     #12,SP
  2572  015700                                  PRINTF  #LINE
        015700  012746  020524                  MOV     #LINE,-(SP)
        015704  012746  000001                  MOV     #1,-(SP)
        015710  010600                          MOV     SP,R0
        015712  104417                          TRAP    C$PNTF
        015714  062706  000004                  ADD     #4,SP
  2573  015720                          1$:     POP     <R5,R1>
  2574  015724                                  EXIT    MSG
        015724  000167                          .WORD   J$JMP
        015726  C02706                          .WORD   L10003-2-.
  2575
  2576  015730     045     101     103  ERR00:: .ASCIZ  ?%ACOMMAND: %T%S3%AT/MSCP UNIT: %03%A(0)?
  2577  016000     045     101     103  ERR01:: .ASCIZ  ?%ACOMMAND: %T%A-REV%S3%AT/MSCP UNIT: %03%A(0)?
  2578  016056     045     101     103  ERR02:: .ASCIZ  ?%ACOMMAND: %T%A-LEOT%S3%AT/MSCP UNIT: %03%A(0)?
  2579  016135     045     101     103  ERR03:: .ASCIZ  ?%ACOMMAND: %T%A-IMM%S3%AT/MSCP UNIT: %03%A(0)?
  2580  016213     045     101     103  ERR04:: .ASCIZ  ?%ACOMMAND: %T%A-UNLD%S3%AT/MSCP UNIT: %03%A(0)?
  2581  016272     045     101     103  ERR05:: .ASCIZ  ?%ACOMMAND: %T%A-WRPR%S3%AT/MSCP UNIT: %03%A(0)?
  2582  016351     045     116     045  ERR06:: .ASCIZ  ?%N%APASS: %03%A(D)   DATA PAT: %02%A(0)?
  2583  016421     045     116     045  ERR07:: .ASCIZ  ?%N%ARECORD BYTE COUNT: %D6%A(D)?
  2584  016461     045     116     045  ERR08:  .ASCIZ  ?%N%AOBJECT CNT : %06%06%A(0)?
  2585  016516     045     116     045  ERR09:: .ASCIZ  ?%N%ADRIVE ERROR CODE : %T%T%A(H)?
  2586  016557     045     116     045  ERR10:: .ASCIZ  ?%N%ASA CONTENTS: %06%A(0)?
  2587  016611     045     116     045  ERR11:: .ASCIZ  ?%N%N%ARESPONSE PACKET?
  2588  016637     045     116     045  ERR12:: .ASCIZ  ?%N%S5%AHIGH WORD%S6%ALOW WORD?
  2589  016675     045     116     045  ERR13:: .ASCIZ  ?%N%S5%06%A(0)%S6%06%A(0)?
  2590  016726     045     116     045  ERR14:: .ASCIZ  ?%N%S2%ABYTE%S10%ADATA?
  2591  016754     045     116     045  ERR15:: .ASCIZ  ?%N%S2%AADDR%S3%AEXPECTED%S3%ARECEIVED%N?
  2592  017024     045     123     061  ERR16:: .ASCIZ  ?%S1%06%S5%03%S8%03%N?
  2593  017051     045     101     124  ERR17:: .ASCIZ  ?%ATOTAL BYTES IN ERROR : %D4%N?
  2594  017110     045     116     045  ERR18:: .ASCIZ  ?%N%ARETRIES:  %D2%A(D)?
  2595  017137     045     116     045  ERL00:: .ASCIZ  ?%N%ATAP OBJ CNT: %06%06%A(0)?
  2596  017174     045     116     045  ERL01:: .ASCIZ  ?%N%ATRK NUM:  %D2%A(D)  LEVEL:  %D2%A(D)   RETRIES:  %D2%A(D)?
  2597  017272     045     116     045  ERL02:: .ASCIZ  ?%N%ALOG BLK NUM:  %D3%A(D)  PHYS BLK NUM:  %D6%A(D)?
  2598  017356     045     116     045  ERL03:: .ASCIZ  ?%N%ADRV CODE:     %03%A(0)  DRV FLGS:     %03%A(0)?
  2599  017445     045     116     045  ERL04:: .ASCIZ  ?%N%ADRV STATE: %06%A(0)  INTERN STATUS:    %03%A(0)?
  2600  017531     045     116     045  ERL05:: .ASCIZ  ?%N%ATAP CNT 0:    %03%A(0)  TAP CNT 1:     %03%A(0)?
  2601  017620     045     116     045  ERL06:: .ASCIZ  ?%N%ATAP CNT 2:    %03%A(0)  RD/WR STATE:   %06%A(0)?
  2602  017704     045     116     045  ERL07:: .ASCIZ  ?%N%AOPER FLGS: %06%A(0)?
  2603  017734     045     116     045  ERL08:: .ASCIZ  ?%N%N%AERROR LOG PACKET?
  2604  017763     045     116     045  NCLK::  .ASCIZ  ?%N%A*\*/*\*/* CLOCK NOT PRESENT *\*/*\*/*%N?
  2605  020037     045     116     045  TIME::  .ASCIZ  ?%N%A*\*/*\*/* TIME %Z2%A:%Z2%A:%Z2%A *\*/*\*/*%N%N?
  2606  020122     045     116     045  DAY::   .ASCIZ  ?%N%A*\*/*\*/* START OF DAY %02%A *\*/*\*/*%N?
  2607  020177     045     101     102  COUNTS::.ASCIZ  ?%ABYTE COUNT: %D%A(D)  FILE SIZE: %D%A(D)%N?
  2608  020253     045     116     045  UNTEOT::.ASCIZ  ?%N%AUNIT: %01%A IS AT EOT%N?
  2609  020307     045     116     045  UNTLOT::.ASCIZ  ?%N%AUNIT: %01%A IS AT LEOT%N?
  2610  020344     045     116     045  DUMP::  .ASCIZ  ?%N%AR1:%06%A R2:%06%A R3:%06%A R4:%06%A R5:%06%N?
  2611  020425     045     116     045  DUMP1:: .ASCIZ  ?%N%N%ABYTES X-FERRED: %D5%A(D)%N?
```

GLOBAL ERROR REPORT SECTION

```
2612 020466      045      117      066  DUMP2:: .ASCIZ  ?%O6%S3%O6%S3%O6%S3%O6%S3%O6%N?
2613 020524      045      116      000  LINE::  .ASCIZ  ?%N?
2614 020527      045      116      045  BYPASS::.ASCIZ  /%N%A TEST %Z3%A BYPASSED%N/
2615 020562      045      116      045  TSTGCR::.ASCIZ  /%N%A TESTING IN GCR%N/
2616 020610      045      116      045  TSTPE:: .ASCIZ  /%N%A TESTING IN PE%N/
2617                                    .EVEN
2618 020634                             ENDMSG
     020636                         L10003:
     020636  104423                  TRAP    C$MSG
```

```
                                              ;PROT  TION TABLE

     020640                                           B 'PROT
     020640                                   L$  ^T::
     020640  000000                                   .WORD   0
     020642  177777                                   .WORD   -1
     020644  177777                                   .WORD    1
     020646                                           ENDPROT
```

```
                                        .SBTTL  CLOCK HANDLER
                                        BGNSRV  NOCLK
020646                          NOCLK::
                                        CLRB    CLOCK           ;CLEAR THE CLOCK ENABLED BIT
020646  105037  002216                  BIS     #NCLKFL,PCFLAG  ;SET UP NO CLOCK PRESENT FLAG
020652  052737  000002  003674          PRINTF  #NCLK           ;PRINT MESSAGE
020660                                  MOV     #NCLK, (SP)
020660  012746  017763                  MOV     #1,-(SP)
020664  012746  000001                  MOV     SP,R0
020670  010600                          TRAP    C$PNTF
020672  104417                          ADD     #4,SP
020674  062706  000004
                                        ENDSRV
020700                          L10005:
020700                                  RTI
020700  000002
                                        BGNSRV  KWHDL
020702                          KWHDL::
020702
                                        INCB    SUBSEC          ;INCREMENT THE SUB SECOND COUNTER
020702  105237  002222                  CMPB    #60.,SUBSEC     ;IS IT A SECOND YET ?
020706  122737  000074  002222          BNE     HDLEXT          ;NO, GET OUT
020714  001051                          CLRB    SUBSEC          ;CLEAR THE SUBSEC COUNTER
020716  105037  002222                  INCB    SECOND          ;INCREMNET THE SECONDS COUNTER
020722  105237  002221                  INC     TIMER           ;INCREMENT THE COMMAND TIMER
020726  005237  010742                  CMPB    #60.,SECOND     ;IS IT A MINUTE YET ?
020732  122737  000074  002221          BNE     HDLEXT          ;NO, GET OUT
020740  001037                          CLRB    SECOND          ;CLEAR THE SECOND COUNTER
020742  105037  002221                  INCB    MINUTE          ;INCREMENT THE MINUTE COUNTER
020746  105237  002220                  CMPB    #60.,MINUTE     ;IS IT AN HOUR YET ?
020752  122737  000074  002220          BNE     HDLEXT          ;NO, GET OUT
020760  001027                          CLRB    MINUTE          ;CLEAR THE MINUTE COUNTER
020762  105037  002220                  INCB    HOURS           ;INCREMENT THE HOUR COUNTER
020766  105237  002217                  CMPB    #24.,HOURS      ;IS IT A DAY YET ?
020772  122737  000030  002217          BNE     HDLEXT          ;NO, GET OUT
021000  001017                          CLRB    HOURS           ;CLEAR THE HOURS COUNTER
021002  105037  002217                  INCB    DAYS            ;INCREMENT THE DAY COUNT
021006  105237  003750                  PRINTF  #DAY,<B,DAYS>   ;PRINT END OF DAY STATMENT
021012                                  CLR     -(SP)
021012  005046                          BISB    DAYS,(SP)
021014  153716  003750                  MOV     #DAY,-(SP)
021020  012746  020122                  MOV     #2,-(SP)
021024  012746  000002                  MOV     SP,R0
021030  010600                          TRAP    C$PNTF
021032  104417                          ADD     #6,SP
021034  062706  000006
021040                          HDLEXT:
021040                                  ENDSRV
021040                          L10006:
021040                                  RTI
021040  000002
```

```
                                        .SBTTL  SCHEDULER
                                        ;*************************
                                        ;
                                        ; SCHEDULER
                                        ;
                                        ;Called by       : Test N
                                        ;Calls to        : CMMDSQ
                                        ;Outputs         : EOT Flag, Dropped Flag
                                        ;Register Inputs: R5 (Pointer to command active in table - not used here)
                                        ;Registers Used : R4 (Pointer to LUN Block for use by called subs)
                                        ;

021042                                  SCHED::
021042  010537  003744                          MOV     R5,CMDSAV               ;SAVE THE CURRENT COMMAND
021046  005001                                  CLR     R1                      ;SET R1 TO FIST UNIT
021050  005037  002074                          CLR     L$LUN                   ;SET L$LUN TO FIRST UNIT
021054  012704  002322                          MOV     #LUNO,R4                ;SET R4 TO THE FIRST LUN BLOCK
021060  022737  000003  002114                  CMP     #3,L$TEST               ;ARE WE IN TEST 3 ?
021066  001014                                  BNE     1$                      ;YES, PRINT LINEFEED
021070  122765  000020  000000                  CMPB    #WR,CMD(R5)             ;IS IT A WRITE COMMAND ?
021076  001010                                  BNE     1$                      ;NO, GET OUT
021100                                          PRINTF  #LINE                   ;PRINT A LINE FEED
021100  012746  020524                          MOV     #LINE,-(SP)
021104  012746  000001                          MOV     #1,-(SP)
021110  010600                                  MOV     SP,R0
021112  104417                                  TRAP    C$PNTF
021114  062706  000004                          ADD     #4,SP
021120  032761  000001  003526          1$:     BIT     #AVB,DRINUS(R1)         ;SEE IF DRIVE IS PRESENT AND AVAILABLE
021126  001424                                  BEQ     2$                      ;GET THE NEXT DRIVE IF IT ISN'T
021130  032761  000004  003526                  BIT     #EOT,DRINUS(R1)         ;CHECK IF THE DRIVE IS AT EOT
021136  001020                                  BNE     2$                      ;GET NEXT DRIVE IF IT IS

021140  012764  000377  000010                  MOV     #377,SLTUSE(R4)         ;SET ALL RESPONSE SLOTS TO THE PORT
021146  004737  026476                          JSR     PC,PRTCLR               ;GO DO IT
021152  112737  000004  010755                  MOVB    #4,CRDLIM               ;CREDITS START AT 4 FOR NEW LUN
021160  004737  021366                          JSR     PC,CMMDSQ               ;GO DO THE TEST ON THIS DRIVE
021164  004737  021232                          JSR     PC,RETRY                ;CHECK IF WE'RE DOING RETRIES
021170  032764  000200  000026                  BIT     #RETFLG,LUNFLG(R4)      ;ARE WE DOING RETRIES ?
021176  001350                                  BNE     1$                      ;BRANCH IF SO

021200  022701  000006                  2$:     CMP     #6.,R1                  ;HAVE WE DONE ALL DRIVES ?
021204  001410                                  BEQ     3$                      ;GET OUT
021206  062701  000002                          ADD     #UNTSTP,R1              ;GET NEXT UNIT
021212  062704  000224                          ADD     #LUNSTP,R4              ;SET UP THE NEXT LUN BLOCK
021216  005237  002074                          INC     L$LUN                   ;GET NEXT UNIT
021222                                          BREAK
021222  104422                                  TRAP    C$BRK
021224  000735                                  BR      1$                      ;GO DO THE NEXT ONE
021226  000240                          3$:     NOP     ;TEMP
021230  000207                                  RTS     PC                      ;RETURN
```

```
                                        .SBTTL  RETRY
                                        ;**************************
                                        ;
                                        ;RETRY
                                        ;
                                        ;Called by      :SCHEDULER
                                        ;Inputs         :MRETRY, ARETRY, CMDSAV
                                        ;Outputs        :Retry flag
                                        ;Register output:R5 (Pointer to command active in table)
                                        ;Registers Used :R4 (Pointer to LUN block)
                                        ;

021232                                  RETRY::
021232  005737  003560                        TST     MRETRY                  ;ARE WE DOING MANUAL RETRIES
021236  001427                              BEQ     20$                     ;BRANCH IF NOT
021240  052764  000200  000026              BIS     #RETFLG,LUNFLG(R4)      ;SET RETRY FLAG
021246  032737  000001  003560              BIT     #1,MRETRY               ;TIME FOR SPACE RECORD REVERSE ?
021254  001014                              BNE     10$                     ;BRANCH IF NOT
021256  022764  000001  000034              CMP     #1,OBJFDL(R4)           ;IS THIS THE FIRST OBJECT ON TAPE ?
021264  001004                              BNE     5$                      ;NO, DO REPOSITION REVERSE
021266  012705  042340                      MOV     #T1REW,R5               ;YES, SET UP TO DO A REWIND
021272  000240                              NOP     ;TEMP
021274  000433                              BR      100$
                                        ;
021276  012705  042616              5$:     MOV     #RTSPR1,R5              ;SET UP TO BACK UP ONE RECORD
021302  000240                              NOP     ;TEMP
021304  000427                              BR      100$
                                        ;
021306  013705  003744              10$:    MOV     CMDSAV,R5               ;RESTORE FAILING COMMAND
021312  000240                              NOP     ;TEMP
021314  000423                              BR      100$
                                        ;
021316  005737  003564              20$:    TST     ARETRY                  ;ARE WE DOING AUTO RETRIES
021322  001420                              BEQ     100$                    ;BRANCH IF NOT
021324  032737  000001  003564              BIT     #1,ARETRY               ;TIME FOR SPACE RECORD REVERSE ?
021332  001012                              BNE     30$                     ;BRANCH IF NOT
021334  022764  000001  000034              CMP     #1,OBJFDL(R4)           ;IS THIS THE FIRST OBJECT ON TAPE ?
021342  001003                              BNE     25$                     ;NO, DO REPOSITION REVERSE
021344  012705  042340                      MOV     #T1REW,R5               ;YES, SET UP TO DO A REWIND
021350  000405                              BR      100$
                                        ;
021352  012705  042616              25$:    MOV     #RTSPR1,R5              ;SET UP TO BACK UP ONE RECORD
021356  000402                              BR      100$
                                        ;
021360  013705  003744              30$:    MOV     CMDSAV,R5               ;RESTORE FAILING COMMAND
021364  000207                      100$:   RTS     PC                      ;RETURN
```

```
                                .SBTTL  COMMAND SEQUENCER
                                ;**************************
                                ;
                                ;COMMAND SEQUENCER
                                ;
                                ;Called by      : SCHED
                                ;Calls to       : CMDBLD, QCMD, CLSDRV, RSPHDR, UNJAM
                                ;Register Inputs: R5 - POINTER TO COMMAND ACTIVE IN TABLE
                                ;                 R4   POINTER TO LUN BLOCK
                                ;
                                ;

021366                          CMMDSQ::
021366                                  PUSH    <R1,R5>                 ;SAVE R1 AND R5
021366  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
021370  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
021372  004737  021764                  JSR     PC,CMDBLD               ;GO BUILD THE COMMAND
021376  042737  000100  003674          BIC     #CMDONE,PCFLAG          ;GET SET TO START ISSUING COMMANDS
021404  013737  003576  003536          MOV     ITERS,CMDCNT            ;GET THE COMMAND COUNT
021412  013737  003576  003540          MOV     ITERS,RSPCNT            ;GET THE RESPONSE COUNT
021420  012737  177777  010734          MOV     #-1,CMSTSV              ;RESET THE GCS PROGRESS COUNT

021426  032737  000040  003674  5$:     BIT     #GCSRFL,PCFLAG          ;STILL LOOKING FOR A GCS RESPONSE ?
021434  001023                          BNE     15$                     ;DON'T QUEUE UP THE NEXT COMMAND
021436  005737  003536                  TST     CMDCNT                  ;DO WE STILL HAVE COMMANDS TO ISSUE ?
021442  001004                          BNE     10$                     ;YES , KEEP GOING.
021444  052737  000100  003674          BIS     #CMDONE,PCFLAG          ;SET THE ALL COMMANDS ISSUED FLAG
021452  000414                          BR      15$

021454  004737  023144  10$:     JSR     PC,QCMD                 ;GO QUEUE UP THE NEXT COMMAND
021460  032737  000002  003674          BIT     #NCLKFL,PCFLAG          ;IS A CLOCK PRESENT ?
021466  001003                          BNE     13$                     ;NO CLOCK, START REGULAR TIMER
021470  005037  010742                  CLR     TIMER                   ;SET TIMER TO 0
021474  000403                          BR      15$                     ;GO ISSUE COMMAND
021476  012737  010000  010742  13$:     MOV     #10000,TIMER            ;SET UP THE TIMER

021504                          15$:    BREAK
021504  104422                          TRAP    C$BRK
021506  004737  023616                  JSR     PC,CLSDRV               ;CALL THE CLASS DRIVER
021512  032737  000002  003674          BIT     #NCLKFL,PCFLAG          ;IS A CLOCK PRESENT ?
021520  001007                          BNE     18$                     ;NO CLOCK, START REGULAR TIMER
021522  022737  000171  010742          CMP     #121.,TIMER             ;HAVE WE TIMED OUT ?
021530  101012                          BHI     20$                     ;NO, KEEP GOING
021532  005037  010742                  CLR     TIMER                   ;SET TIMER TO 0
021536  000414                          BR      25$                     ;YES,SET UP FOR TIME OUT

021540  005337  010742          18$:    DEC     TIMER                   ;DECREMENT THE TIMER
021544  001004                          BNE     20$                     ;BRANCH IF NOT 0
021546  012737  010000  010742          MOV     #10000,TIMER            ;RESET THE TIMER
021554  000405                          BR      25$                     ;SET TIME OUT ERROR

021556  022737  020000  010732  20$:     CMP     #IOICRD,IOSTAT          ;INSUFFICIENT CREDITS ?
021564  001747                          BEQ     15$                     ;YES, TRY AGAIN
021566  000425                          BR      35$                     ;YES, CHECK IT OUT

021570  032737  000040  003674  25$:     BIT     #GCSRFL,PCFLAG          ;WAITING FOR A GCS RESPONSE ?
021576  001412                          BEQ     30$                     ;NO, SET UP TO DO A GCS
021600  042737  000040  003674          BIC     #GCSRFL,PCFLAG          ;CLEAR THE GCS RESPONSE FLAG
```

```
 2809 021606  012737  000004  010732          MOV     #IOTIME,IOSTAT          ;SET UP TIME OUT ERROR
 2810 021614  004737  034022                  JSR     PC,CORDMP               ;DO A VARIABLES DUMP
 2811 021620  000240                          NOP
 2812 021622  000407                          BR      35$                     ;GO REPORT ERROR AND DROP UNIT
 2813
 2814 021624  052737  000020  003674  30$:    BIS     #GCSCFL,PCFLAG          ;SET THE GCS COMMAND FLAG
 2815 021632  052737  000040  003674          BIS     #GCSRFL,PCFLAG          ;SET THE GCS RESPONSE FLAG
 2816 021640  000721                          BR      15$                     ;GO ISSUE THE GCS COMMAND
 2817
 2818 021642  022737  060000  010732  35$:    CMP     #ERRLOG!IOICRD,IOSTAT   ;DID WE GET ERROR LOG PACKET ONLY?
 2819 021650  001406                          BEQ     40$                     ;YES - SO BRANCH AROUND NEXT INSTRUCTION
 2820 021652  032737  000140  003674          BIT     #CMDONE!GCSRFL,PCFLAG   ;HAVE ALL COMMANDS BEEN ISSUED ?
 2821 021660  001002                          BNE     40$                     ;YES, DON'T LET CMDCNT GO NEGATIVE
 2822 021662  005337  003536                  DEC     CMDCNT                  ;DECREMENT THE COMMAND COUNT
 2823 021666  022737  000000  010732  40$:    CMP     #IONORM,IOSTAT          ;WAS IT A NORMAL COMPLETION ?
 2824 021674  001414                          BEQ     45$                     ;YES , GET OUT
 2825 021676  004737  027706                  JSR     PC,RSPHDL               ;NO, LETS SEE WHAT IT WAS
 2826
 2827 021702  032737  020000  010732          BIT     #IOICRD,IOSTAT          ;WERE WE IN RSPHDL FOR ERROR LOG ONLY?
 2828 021710  C01275                          BNE     15$                     ;YES - GO TRY TO POST SAME COMMAND.
 2829 021712  022737  000002  003552          CMP     #SEREXC,RESPON          ;WAS IT A SERIOUS EXCEPTION ?
 2830 021720  001002                          BNE     45$                     ;NO, CONTINUE
 2831 021722  004737  033420                  JSR     PC,UNJAM                ;YES, GO UNJAM THE QUEUES
 2832
 2833 021726  032761  000001  003526  45$:    BIT     #AVB,DRINUS(R1)         ;HAS THE DRIVE BEEN DROPPED ?
 2834 021734  001407                          BEQ     50$                     ;YES, EXIT
 2835 021736  005737  003540                  TST     RSPCNT                  ;HAVE WE GOTTEN ALL THE RESPONSE BACK ?
 2836 021742  001231                          BNE     5$                      ;NO, GO BACK TO THE TOP
 2837 021744  032737  000040  003674          BIT     #GCSRFL,PCFLAG          ;STILL LOOKING FOR A GCS RESPONSE ?
 2838 021752  001225                          BNE     5$                      ;YES, DON'T GET OUT YET
 2839
 2840 021754                          50$:    POP     <R5,R1>                 ;RESTORE REGISTERS R1 AND R5
      021754  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
      021756  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 2841 021760  000240                          NOP     ;TEMP
 2842 021762  000207                          RTS     PC                      ;RETURN
```

```
                                        .SBTTL  COMMAND BUILDER
                                        ;******************************
                                        ;
                                        ;COMMAND BUILDER
                                        ;
                                        ;Called by      : CMMDSQ
                                        ;Calls to       : BYTCNT, SELDAT, SELREC
                                        ;Register Inputs: R5 - pointer to test's command table
                                        ;                 R4 - pointer to LUN BLOCK
                                        ;Register Output: R3 - new pointer for command ring (set to start of ring)
                                        ;                 R2   old pointer for command ring (set to start of ring)
                                        ;Registers Used : R3 Pointer to dummy packet before setting to command ring
                                        ;
021764                                  CMDBLD::
021764  032764  000200  000026                BIT     @RETFLG,LUNFLG(R4)      ;ARE WE IN RETRY MODE ?
021772  001003                              BNE     1$                      ;YES DON'T INIT SUB COUNT
021774  012737  000004  003602              MOV     @N,SUBCNT               ;INITIALIZE THE SUB-ITERATION COUNTER
022002  012703  003522              1$:     MOV     @DUMPKT,R3              ;PUT THE DUMMY PACKET ADDRESS IN R3
022006  116563  000000  000000              MOVB    CMD(R5),CMD(R3)         ;MOVE THE COMMAND PRIMITIVE TO THE PACKET
                                        ;
022014  004737  022130                      JSR     PC,BYTCNT               ;GO GET THE BYTE COUNT
022020  013763  003574  000002              MOV     BYTES,ITMOFF(R3)        ;PUT THE BYTE COUNT IN THE DUMMY PACKET
022026  004737  022246                      JSR     PC,SELDAT               ;GO GET THE DATA
022032  004737  022712                      JSR     PC,SELREC               ;GO GET THE RECORD COUNT
                                        ;
022036  022737  000003  002114              CMP     @3,L$TEST               ;ARE WE IN TEST 3 ?
022044  001024                              BNE     5$                      ;YES, PRINT COUNTS
022046  122765  000020  000000              CMPB    @WR,CMD(R5)             ;IS IT A WRITE COMMAND ?
022054  001020                              BNE     5$                      ;NO, GET OUT
022056  032764  000200  000026              BIT     @RETFLG,LUNFLG(R4)      ;ARE WE IN RETRY MODE ?
022064  001014                              BNE     5$                      ;YES DON'T PRINT COUNTS
022066                                      PRINTF  @COUNTS,BYTES,ITERS     ;YES, PRINT BYTE AND ITERATION COUNTS
022066  013746  003576                      MOV     ITERS,-(SP)
022072  013746  003574                      MOV     BYTES,-(SP)
022076  012746  020177                      MOV     @COUNTS,-(SP)
022102  012746  000003                      MOV     @3,-(SP)
022106  010600                              MOV     SP,R0
022110  104417                              TRAP    C$PNTF
022112  062706  000010                      ADD     @10,SP
                                        ;
022116  012703  003442              5$:     MOV     @PCMDBF,R3              ;PUT THE PROGRAM COMMAND RING ADDRESS IN
022122  010302                              MOV     R3,R2                   ;R3 AND R2
022124  000240                              NOP     ;TEMP
022126  000207                              RTS     PC                      ;RETURN
                                        ;
                                        ;
```

```
                                .SBTTL  BYTE COUNT
                                ;******************************
                                ;
                                ; BYTE COUNT
                                ;
                                ;Called by       : CMDBLD
                                ;Calls to        : RANGEN
                                ;Outputs         : BYTES (contains byte or item count to be used for this iteration set)
                                ;Register Inputs: R5 - pointer to test command table
                                ;                 R4 - pointer to LUN BLOCK
                                ;Register Output: None
                                ;Registers Used : None
                                ;

022130                          BYTCNT::
022130  005037  003574                  CLR     BYTES                   ;CLEAR BYTES
022134  032764  000200  000026          BIT     #RETFLG,LUNFLG(R4)      ;ARE WE IN RETRY MODE ?
022142  001404                          BEQ     5$                      ;NO, CONTINUE
022144  013737  003746  003574          MOV     BYTSAV,BYTES            ;RESTORE OLD BYTE COUNT
022152  000430                          BR      20$                     ;EXIT
022154  005765  000002          5$:     TST     ITMCNT(R5)              ;CHECK ITMCNT FOR 0
022160  001404                          BEQ     10$                     ;CONTINUE IF IT IS 0
022162  016537  000002  003574          MOV     ITMCNT(R5),BYTES        ;PUT ITMCNT INTO BYTES
022170  000421                          BR      20$                     ;EXIT

022172  122765  000020  000000  10$:    CMPB    #WR,CMD(R5)             ;IS IT A READ OR WRITE
022200  103415                          BLO     20$                     ;GET OUT IF IT ISN'T
022202  004737  023010          15$:    JSR     PC,RANGEN               ;GO TO THE RANDOM GENERATOR
022206  023727  003604  020000          CMP     RANWRD,#MAXBUF          ;IS THE RESULT WITHIN THE LIMITS ?
022214  101372                          BHI     15$                     ;BRANCH IF TOO HIGH
022216  023727  003604  000024          CMP     RANWRD,#MINBUF          ;IS IT TOO SMALL ?
022224  103766                          BLO     15$                     ;BRANCH IF TOO SMALL
022226  013737  003604  003574          MOV     RANWRD,BYTES            ;PUT RANWRD INTO BYTES
022234  013737  003574  003746  20$:    MOV     BYTES,BYTSAV            ;SAVE THE CURRENT BYTE COUNT
022242  000240                          NOP     ;TEMP
022244  000207                          RTS     PC                      ;RETURN
```

```
                        .SBTTL  SELECT DATA PATTERN
                        ;******************************
                        ;
                        ; SELECT DATA PATTERN
                        ;
                        ;Called by      : CMDBLD
                        ;Calls to       : RANGEN
                        ;Inputs         : Data Pattern in test command table
                        ;                 PATSAV in LUN BLOCK if rotating pattern in use
                        ;Outputs        : Write Buffer filled with appropriate data pattern
                        ;                 PATSAV in LUN BLOCK updated to next pattern
                        ;Register Inputs:  R5 - pointer to test command table
                        ;                  R4   pointer to LUN BLOCK
                        ;Registers Used :  R3 - pointer to WRTBUF
                        ;                  R2   pointer to data pattern
                        ;
                        ;
022246                  SELDAT::
022246                          PUSH    <R1,R5>                 ;SAVE R1 AND R5
022252  C32764  000200  000026        BIT     @RETFLG,LUNFLG(R4)      ;ARE WE IN RETRY MODE ?
022260  001050                        BNE     20$                     ;YES DON'T CHANGE DATA
022262  105765  000001                TSTB    DATPAT(R5)              ;TEST DATPAT FOR A TEST PATTERN
022266  001445                        BEQ     20$                     ;BRANCH IF WE DON'T NEED ONE
022270  105737  002235                TSTB    PATERN                  ;PATTERN SPECIFIED IN SOFTWARE P-TABLE ?
022274  001404                        BEQ     1$                      ;NO, KEEP GOING
022276  113764  002235  000024        MOVB    PATERN,PATSAV(R4)       ;PUT THE PATTERN IN THE SAVE LOCATION
022304  000420                        BR      10$
                        ;
022306  105765  000001        1$:     TSTB    DATPAT(R5)              ;DO WE WANT ROTATING DATA PATTERNS ?
022312  100404                        BMI     5$                      ;IF NEGATIVE GO TO 5$
022314  116564  000001  000024        MOVB    DATPAT(R5),PATSAV(R4)   ;LET PATSAV EQUAL DATPAT
022322  000411                        BR      10$                     ;BRANCH
                        ;
022324  005264  000024        5$:     INC     PATSAV(R4)              ;ADD 1 TO PATSAV
022330  026427  000024  000010        CMP     PATSAV(R4),@ENDPAT      ;ARE WE AT THE END OF THE PATTERN TABLE ?
022336  001003                        BNE     10$                     ;NO, KEEP GOING
022340  012764  000001  000024        MOV     @1.,PATSAV(R4)          ;AT THE END, LET PATSAV EQUAL 1
                        ;
022346  013705  003574        10$:    MOV     BYTES,R5                ;PUT THE BYTE COUNT IN R5
022352  032705  000001                BIT     @BIT0,R5                ;IS THE BYTE COUNT ODD ?
022356  001401                        BEQ     15$                     ;BRANCH IF NOT
022360  005205                        INC     R5                      ;MAKE BYTE COUNT EVEN FOR PATGEN
                        ;
022362  012703  070614        15$:    MOV     @WRTBUF,R3              ;POINT R3 TO THE WRITE BUFFER
022366  116401  000024                MOVB    PATSAV(R4),R1           ;SAVE PATSAV IN R1
022372  005301                        DEC     R1                      ;ADJUST FOR TABLE STEP
022374  006301                        ASL     R1                      ;MAKE IT MOD 2 OFFSET
                        ;
022376  004771  022412                JSR     PC,@PATTBL(R1)          ;GO FILL THE BUFFER
022402                  20$:    POP     <R5,R1>                 ;RESTORE R5 AND R1
022406  000240                        NOP     ;TEMP
022410  000207                        RTS     PC                      ;RETURN
                        ;
                        ;
022412                  PATTBL::
022412  022430                        .WORD   PATGN1                  ;ALL 1'S
022414  022444                        .WORD   PATGN2                  ;ALL 0'S
```

SELECT DATA PATTERN

```
2979 022416  022456                        .WORD   PATGN3          ;WORST CASE MFM
2980 022420  022516                        .WORD   PATGN4          ;ALTERNATE 1'S AND 0'S
2981 022422  022532                        .WORD   PATGN5          ;RANDOM DATA
2982 022424  022552                        .WORD   PATGN6          ;1110 REPEATING PATTERN
2983 022426  022566                        .WORD   PATGN7          ;COMBINATION PAT 3 AND 5
2984
2985 022430                        PATGN1:
2986 022430  012723  177777                MOV     #-1,(R3)+       ;PUT ALL 1'S INTO THE BUFFER
2987 022434  162705  000002                SUB     #2,R5           ;SUBTRACT TWO FROM R5
2988 022440  001373                        BNE     PATGN1          ;KEEP GOING IF WE AREN'T AT 0
2989 022442  000207                        RTS     PC              ;RETURN
2990
2991 022444                        PATGN2:
2992 022444  005023                        CLR     (R3)+           ;PUT ALL 0'S INTO THE BUFFER
2993 022446  162705  000002                SUB     #2,R5           ;SUBTRACT TWO FROM R5
2994 022452  001374                        BNE     PATGN2          ;KEEP GOING IF WE AREN'T AT 0
2995 022454  000207                        RTS     PC              ;RETURN
2996
2997 022456                        PATGN3:
2998 022456  012723  133333                MOV     #133333,(R3)+   ;PUT THE NUMBER INTO THE BUFFER
2999 022462  162705  000002                SUB     #2,R5           ;SUBTRACT TWO FROM R5
3000 022466  001412                        BEQ     1$              ;KEEP GOING IF WE AREN'T AT 0
3001 022470  012723  155555                MOV     #155555,(R3)+   ;PUT THE NUMBER INTO THE BUFFER
3002 022474  162705  000002                SUB     #2,R5           ;SUBTRACT TWO FROM R5
3003 022500  001405                        BEQ     1$              ;KEEP GOING IF WE AREN'T AT 0
3004 022502  012723  066666                MOV     #066666,(R3)+   ;PUT THE NUMBER INTO THE BUFFER
3005 022506  162705  000002                SUB     #2,R5           ;SUBTRACT TWO FROM R5
3006 022512  001361                        BNE     PATGN3          ;KEEP GOING IF WE AREN'T AT 0
3007 022514  000207                1$:     RTS     PC              ;RETURN
3008
3009 022516                        PATGN4:
3010 022516  012723  125252                MOV     #125252,(R3)+   ;PUT ALTERNATING 1 AND 0 INTO THE BUFFER
3011 022522  162705  000002                SUB     #2,R5           ;SUBTRACT TWO FROM R5
3012 022526  001373                        BNE     PATGN4          ;KEEP GOING IF WE AREN'T AT 0
3013 022530  000207                        RTS     PC              ;RETURN
3014
3015 022532                        PATGN5:
3016 022532  004737  023010                JSR     PC,RANGEN       ;GO GENERATE RANDOM PATTERN
3017 022536  013723  003604                MOV     RANWRD,(R3)+    ;PUT THE NUMBER INTO THE BUFFER
3018 022542  162705  000002                SUB     #2,R5           ;SUBTRACT TWO FROM R5
3019 022546  001371                        BNE     PATGN5          ;KEEP GOING IF WE AREN'T AT 0
3020 022550  000207                        RTS     PC              ;RETURN
3021
3022 022552                        PATGN6:
3023 022552  012723  167356                MOV     #167356,(R3)+   ;PUT 1110 REPEATING IN BUFFER
3024 022556  162705  000002                SUB     #2,R5           ;SUBTRACT TWO FROM R5
3025 022562  001373                        BNE     PATGN6          ;KEEP GOING IF WE AREN'T AT 0
3026 022564  000207                        RTS     PC              ;RETURN
3027
3028 022566                        PATGN7:
3029 022566                                PUSH    <R2>
3030 022570  012702  001000        1$:     MOV     #512.,R2
3031 022574  012723  133333        3$:     MOV     #133333,(R3)+   ;PUT THE NUMBER INTO THE BUFFER
3032 022600  162705  000002                SUB     #2,R5           ;SUBTRACT TWO FROM R5
3033 022604  001440                        BEQ     10$             ;KEEP GOING IF WE AREN'T AT 0
3034 022606  162702  000002                SUB     #2,R2           ;HAVE WE DONE A FULL BLOCK YET
3035 022612  001420                        BEQ     5$              ;YES DO NEXT BLOCK IN PATTERN 5
```

SELECT DATA PATTERN

```
3036 022614  012723  155555          MK'     #155555,(R3)+       ;PUT THE NUMBER INTO THE BUFFER
3037 022620  162705  000002          SUB     #2,R5               ;SUBTRACT TWO FROM R5
3038 022624  001430                  BEQ     10$                 ;KEEP GOING IF WE AREN'T AT 0
3039 022626  162702  000002          SUB     #2,R2               ;HAVE WE DONE A FULL BLOCK YET
3040 022632  001410                  BEQ     5$                  ;YES DO NEXT BLOCK IN PATTERN 5
3041 022634  012723  066666          MOV     #066666,(R3)+       ;PUT THE NUMBER INTO THE BUFFER
3042 022640  162705  000002          SUB     #2,R5               ;SUBTRACT TWO FROM R5
3043 022644  001420                  BEQ     10$                 ;KEEP GOING IF WE AREN'T AT 0
3044 022646  162702  000002          SUB     #2,R2               ;HAVE WE DONE A FULL BLOCK YET
3045 022652  001350                  BNE     3$                  ;YES DO NEXT BLOCK IN PATTERN 5
3046 022654  012702  001000    5$:   MOV     #512.,R2
3047 022660  004737  023010    6$:   JSR     PC,RANGEN           ;GO GENERATE RANDOM PATTERN
3048 022664  013723  003604          MOV     RANWRD,(R3)+        ;PUT THE NUMBER INTO THE BUFFER
3049 022670  162705  000002          SUB     #2,R5               ;SUBTRACT TWO FROM R5
3050 022674  001404                  BEQ     10$                 ;KEEP GOING IF WE AREN'T AT 0
3051 022676  162702  000002          SUB     #2,R2               ;HAVE WE DONE A FULL BLOCK YET
3052 022702  001366                  BNE     6$                  ;YES DO NEXT BLOCK IN PATTERN 5
3053 022704  000731                  BR      1$
3054 022706                  10$:    POP     <R2>
3055 022710  000207                  RTS     PC                  ;RETURN
3056
3057
```

```
 3059                                   .SBTTL  SELECT RECORD
 3060                                   ;*******************************
 3061                                   ;
 3062                                   ; SELECT RECORD
 3063                                   ;
 3064                                   ;Called by       : CMOBLD
 3065                                   ;Calls to        : RANGEN
 3066                                   ;Outputs         : ITERS (number of iterations for this set)
 3067                                   ;Register Inputs: R5 - pointer to test command table
 3068                                   ;                 R4 - pointer to LUN BLOCK
 3069                                   ;
 3070
 3071 022712                           SELREC::
 3072 022712  032764  000200  000026          BIT     #RETFLG,LUNFLG(R4)      ;ARE WE IN RETRY MODE ?
 3073 022720  001404                          BEQ     5$                      ;NO, KEEP GOING
 3074 022722  012737  000001  003576          MOV     #1,ITERS                ;SET THE ITERATION COUNT TO 1
 3075 022730  000425                          BR      15$                     ;GET OUT
 3076
 3077 022732  005765  000004          5$:     TST     ITRCNT(R5)              ;TEST THE ITERATION COUNT
 3078 022736  001404                          BEQ     10$                     ;IF IT IS 0 THEN BRANCH
 3079 022740  016537  000004  003576          MOV     ITRCNT(R5),ITERS        ;SAVE THE ITERATION COUNT
 3080 022746  000416                          BR      15$                     ;GET OUT
 3081
 3082 022750  004737  023010          10$:    JSR     PC,RANGEN               ;GO TO THE RANDOM GENERATOR
 3083 022754  023727  003604  003720          CMP     RANWRD,#MAXITR          ;IS THE ITERATION COUNT TO HIGH ?
 3084 022762  101372                          BHI     10$                     ;GO TRY AGAIN
 3085 022764  023727  003604  000144          CMP     RANWRD,#MINITR          ;IS THE ITERATION SET TOO SMALL ?
 3086 022772  103766                          BLO     10$                     ;GO TRY AGAIN
 3087 022774  013737  003604  003576          MOV     RANWRD,ITERS            ;SAVE THE RANDOM NUMBER
 3088 023002  000400                          BR      15$                     ;EXIT
 3089
 3090 023004  000240                  15$:    NOP     ;TEMP
 3091 023006  000207                          RTS     PC                      ;RETURN
```

```
                                .SBTTL  RANDOM NUMBER GENERATOR
                                ;*****************************
                                ;
                                ;RANDOM NUMBER GENERATOR
                                ;
                                ;Called by      : BYTCNT, ELDAT, SELREC
                                ;Inputs         : RAN1, RAN2, RAN3
                                ;Outputs        : RANWRD
                                ;Registers Used : R5
                                ;
                                RANGEN::
023010                                  PUSH    <R5>                    ;SAVE R5
023012  016437  000202  003606          MOV     SEED1(R4),RAN1          ;PUT SEED1 INTO RAN1
023020  016437  000204  003610          MOV     SEED2(R4),RAN2          ;PUT SEED2 INTO RAN2
023026  016437  000206  003612          MOV     SEED3(R4),RAN3          ;PUT SEED3 INTO RAN3
023034  013705  003606                  MOV     RAN1,R5                 ;MOVE THE FIRST SEED INTO R5
023040  000241                          CLC                             ;CLEAR THE CARRY FLAG
023042  005337  003612                  DEC     RAN3                    ;DECREMENT THE THIRD SEED
023046  006105                          ROL     R5                      ;
023050  006105                          ROL     R5                      ;
023052  063705  003610                  ADD     RAN2,R5                 ;ADD THE SECOND SEED TO R5
023056  010537  003606                  MOV     R5,RAN1                 ;PUT IT ALL IN THE FIRST SEED
023062  063705  003612                  ADD     RAN3,R5                 ;PUT THE THIRD SEED INTO R5
023066  006105                          ROL     R5                      ;
023070  006105                          ROL     R5                      ;
023072  063705  003610                  ADD     RAN2,R5                 ;ADD THE SECOND SEED TO R5
023076  006105                          ROL     R5                      ;
023100  006105                          ROL     R5                      ;
023102  010537  003610                  MOV     R5,RAN2                 ;PUT IT IN THE SECOND SEED
023106  013737  003606  003604          MOV     RAN1,RANWRD             ;PUT THE FIRST SEED INTO RANWORD
023114  013764  003606  000202          MOV     RAN1,SEED1(R4)          ;PUT RAN1 INTO SEED1
023122  013764  003610  000204          MOV     RAN2,SEED2(R4)          ;PUT RAN2 INTO SEED2
023130  013764  003612  000206          MOV     RAN3,SEED3(R4)          ;PUT RAN3 INTO SEED3
023136                                  POP     <R5>                    ;RESTORE R5
023140  000240                          NOP     ;TEMP
023142  000207                          RTS     PC                      ;EXIT
```

```
                                .SBTTL  QUEUE COMMANDS
                                ;**********************************
                                ;
                                ; QUEUE COMMANDS
                                ;
                                ;Called by      : CMMDSQ
                                ;Calls to       : SUBITR
                                ;Register Inputs: R3   pointer to next slot in ring
                                ;                 R4   pointer to LUN BLOCK
                                ;Register Output: R3   updated to point to next available slot
                                ;Registers Used : R5   Points to dummy packet
                                ;
023144                          QCMD::
023144  022703  003522                  CMP     #PCBEND,R3              ;IS R3 POINTING AT THE END OF THE RING ?
023150  001002                         BNE     1$                      ;NO, THEN KEEP GOING
023152  012703  003442                  MOV     #PCMDBF,R3              ;YES, SET IT TO THE RING BEGINNING
023156  012705  003522          1$:     MOV     #DUMPKT,R5              ;POINT R5 TO THE DUMMY PACKET
023162  116563  000000  000000        MOVB    CMD(R5),CMD(R3)         ;PUT THE COMMAND PRIMITIVE INTO THE RING
023170  016563  000002  000002        MOV     ITMOFF(R5),ITMOFF(R3)   ;PUT THE ITEM OFFSET INTO THE RING

023176  004737  023436                  JSR     PC,OBCTHD               ;GO GET THE OBJECT COUNT
023202  016463  000034  000004        MOV     OBJFDL(R4),OBOFFL(R3)   ;PUT THE LOW FIELD INTO THE RING
023210  016463  000036  000006        MOV     OBJFDH(R4),OBOFFH(R3)   ;PUT THE HIGH FIELD INTO THE RING

023216  005037  003600                  CLR     BUFADR                  ;CLEAR THE BUFFER ADDRESS LOCATION
023222  004737  023246                  JSR     PC,SUBITR               ;GO TO SUB ITERS
023226  000400                         BR      5$
023230  013763  003600  000012  5$:     MOV     BUFADR,BUFOFF(R3)       ;PUT THE BUFFER ADDRESS INTO THE RING
023236  062703  000014                  ADD     #PCBSTP,R3              ;MOVE R3 TO THE NEXT SLOT IN THE RING
023242  000240                         NOP     ;TEMP
023244  000207                         RTS     PC                      ;RETURN
                                .EVEN
```

```
3165                                    .SBTTL  SUB ITERATION
3166                                    ;*****************************
3167                                    ;
3168                                    ; SUB ITERATION
3169                                    ;
3170                                    ;Called by       : QCMD
3171                                    ;Outputs         : BUFADR
3172                                    ;Register Inputs: R3   pointer to command slot
3173                                    ;                 R4   pointer to LUN BLOCK
3174                                    ;
3175
3176 023246                            SUBITR::
3177 023246  105763  000000                      TSTB    CMD(R3)                 ;ARE WE ISSUING NULL COMMANDS ?
3178 023252  001467                              BEQ     35$                     ;BRANCH IF THE NULL COMMAND
3179 023254  122763  000020  000000              CMPB    #WR,CMD(R3)             ;IS IT GREATER THAN A WRITE
3180 023262  103447                              BLO     25$                     ;YES, BRANCH
3181
3182 023264  005337  003602          5$:         DEC     SUBCNT                  ;SUBTRACT 1 FROM SUBCNT
3183 023270  001025                              BNE     15$                     ;BRANCH IF NOT 0
3184 023272  C16337  000004  070612              MOV     OBOFFL(R3),WRTBUF-2     ;PUT LOW ORDER OBJECT COUNT IN WRTBUF
3185 023300  012737  000004  003602              MOV     #N,SUBCNT               ;REINIT SUBCNT
3186 023306  012737  050614  003600  10$:        MOV     #RDBUF,BUFADR           ;PUT THE READ BUFFER ADDRESS IN BUFADR
3187 023314  122763  000020  000000              CMPB    #WR,CMD(R3)             ;IS IT A WRITE COMMAND
3188 023322  001043                              BNE     35$                     ;GET OUT IF IT'S NOT
3189 023324  022737  000003  002114              CMP     #3,L$TEST               ;ARE WE IN TEST 3 ?
3190 023332  001017                              BNE     20$                     ;NO, SET WRITE BUFFER IN BUFADR
3191 023334  012737  070612  003600              MOV     #WRTBUF-2,BUFADR        ;SET MODIFIED WRITE BUFFER IN BUFADR
3192 023342  000433                              BR      35$                     ;EXIT
3193
3194 023344  122763  000020  000000  15$:        CMPB    #WR,CMD(R3)             ;SEE IF ITS A WRITE
3195 023352  001407                              BEQ     20$                     ;YES, BRANCH
3196 023354  022737  000006  002114              CMP     #6,L$TEST               ;ARE WE IN TEST 6 ?
3197 023362  001751                              BEQ     10$                     ;YES, PUT READ BUFFER IN BUFADR
3198 023364  112763  000040  000000              MOVB    #ACC,CMD(R3)            ;SET UP AN ACCESS DATA COMMAND
3199 023372  012737  070614  003600  20$:        MOV     #WRTBUF,BUFADR          ;SET WRTBUF ADDRESS IN BUFADR
3200 023400  000414                              BR      35$
3201
3202 023402  032764  000200  000026  25$:        BIT     #RETFLG,LUNFLG(R4)      ;ARE WE IN RETRY MODE
3203 023410  001410                              BEQ     35$                     ;NO, GET OUT
3204 023412  022737  000004  003602              CMP     #N,SUBCNT               ;JUST DO A TRACE RECORD
3205 023420  001002                              BNE     30$                     ;NO, INCREMENT THE SUB COUNT
3206 023422  005037  003602                      CLR     SUBCNT                  ;YES, CLEAR THE SUB COUNT
3207 023426  005237  003602          30$:        INC     SUBCNT                  ;INCREMENT THE SUB COUNT
3208 023432  000240                  35$:        NOP     ;TEMP
3209 023434  000207                              RTS     PC                      ;RETURN
```

```
                                .SBTTL  OBJET COUNT HANDLER
                                ;******************************
                                ;
                                ; OBJECT COUNT HANDLER
                                ;
                                ;Called by      : QCMD
                                ;Inputs         : Current Object Count in LUN BLOCK
                                ;Outputs        : Updated Object Count in LUN BLOCK
                                ;Register Inputs: R3   pointer to command slot
                                ;                 R4   pointer to LUN BLOCK
                                ;
                                ;

023436                          OBCTHD::
023436                                  PUSH    <R1>                    ;SAVE R1
023440  116301  000000                  MOVB    CMD(R3),R1              ;PUT THE COMMAND PRIMITIVE INTO R1
023444  042701  000007                  BIC     #7,R1                   ;STRIP OFF THE MODIFIERS
023450  005701                          TST     R1                      ;IS IT THE NULL COMMAND ?
023452  001456                          BEQ     6$                      ;EXIT IF IT IS
023454  022701  000160                  CMP     #REW,R1                 ;IS IT A REWIND ?
023460  001005                          BNE     1$                      ;BRANCH IF NOT
023462  005064  000034                  CLR     OBJFDL(R4)              ;CLEAR THE OBJECT
023466  005064  000036                  CLR     OBJFDH(R4)              ;COUNT FIELD AND
023472  000446                          BR      6$                      ;EXIT

023474  032764  000200  000026  1$:     BIT     #RETFLG,LUNFLG(R4)      ;ARE WE IN RETRY MODE
023502  001012                          BNE     2$                      ;YES, ONLY ONE OBJECT AT A TIME
023504  022701  000050                  CMP     #SPC,R1                 ;IS IT A NON DATA TRANSFER COMMAND ?
023510  101007                          BHI     2$                      ;BRANCH IF IT IS
023512  022701  000100                  CMP     #WTM,R1                 ;IS IT A WRITE TAPE MARK ?
023516  001404                          BEQ     2$                      ;BRANCH IF IT IS
023520  016337  000002  003710          MOV     ITMOFF(R3),R8           ;PUT THE ITEM COUNT IN TEMP REGISTER
023526  000403                          BR      3$                      ;CONTINUE

023530  012737  000001  003710  2$:     MOV     #1,R8                   ;PUT A 1 IN THE TEMP REGISTER
023536  032763  000002  000000  3$:     BIT     #EOTBIT,CMD(R3)         ;IS IT AN LEOT COMMAND ?
023544  001021                          BNE     6$                      ;GET OUT IF IT IS
023546  032763  000001  000000          BIT     #REVBIT,CMD(R3)         ;IS THE COMMAND REVERSE ?
023554  001007                          BNE     4$                      ;BRANCH IF REVERSE
023556  063764  003710  000034          ADD     R8,OBJFDL(R4)           ;ADD TEMP TO THE OBJECT COUNT
023564  103011                          BCC     6$                      ;BRANCH IF NO CARRY
023566  005264  000036                  INC     OBJFDH(R4)              ;OTHERWISE ADD 1 TO THE HIGH OBJECT COUNT
023572  000406                          BR      6$                      ;EXIT

023574  163764  003710  000034  4$:     SUB     R8,OBJFDL(R4)           ;IF REVERSE, SUBTRACT TEMP FROM THE
023602  103002                          BCC     6$                      ;OBJECT COUNT AND BRANCH IF NO CARRY
023604  005364  000036                  DEC     OBJFDH(R4)              ;OTHERWISE SUBTRACT 1 FROM OBJECT COUNT HIGH
023610                          6$:     POP     <R1>                    ;RESTORE R1
023612  000240                          NOP     ;TEMP
023614  000207                          RTS     PC                      ;EXIT
```

```
                        .SBTTL  CLASS DRIVER TRANSMIT
                        ;******************************
                        ;
                        ;Class Driver Transmit
                        ;
                        ;Called By      : CMMDSQ
                        ;Calls To       : CDRECV, STFPCK, PRTDRV
                        ;Inputs         : CRDLIM - Command slots open in the drive.
                        ;                 COLSAV - Old driver command pointer.
                        ;Outputs        : IOSTAT - Transfer status.
                        ;                 CMDSEQ   Number appended to each command packet.
                        ;                 GCSREF - Get Command Status reference number.
                        ;Register Inputs: R2 - Old pointer to program command ring.
                        ;                 R3 - New pointer to program command ring.
                        ;                 R4 - Lun block pointer.
                        ;Register Outputs: R5 - Old pointer to driver command ring.
                        ;

023616                  CLSDRV::
023616                          PUSH    <R3,R5>                 ;SAVE R3, R5
023622  016405  000016          MOV     COLSAV(R4),R5           ;POINT R5 TO THE OLD DRIVER COMMAND
023626  032737  000040  003674  BIT     #GCSRFL,PCFLAG          ;IS THIS A GCS COMMAND ?
023634  001010                  BNE     10$                     ;YES, GO SETUP
023636  022703  003442          CMP     #PCMDBF,R3              ;IS R3 AT THE BEGINNING OF THE PROGRAM RING ?
023642  001403                  BEQ     5$                      ;YES, BRANCH
023644  162703  000014          SUB     #PCBSTP,R3              ;NO, MOVE R3 ONE SLOT BACK
023650  000402                  BR      10$                     ;CONTINUE

023652  062703  000044  5$:     ADD     #PCB3SP,R3              ;YES, ADVANCE R3 TWO SLOTS
023656  005037  010732  10$:    CLR     IOSTAT                  ;CLEAR THE I/O STATUS WORD
023662  122763  000170  000000  CMPB    #INT,CMD(R3)            ;IS THIS A INITIALIZATION COMMAND ?
023670  001003                  BNE     15$                     ;CONTINUE IF IT ISN'T

023672  004737  026556          JSR     PC,PRTINT               ;CALL THE PORT INIT ROUTINE
023676  000464                  BR      55$                     ;EXIT
023700  005764  000010  15$:    TST     SLTUSE(R4)              ;DID WE HANDLE ANY RESPONSES LAST TIME ?
023704  001402                  BEQ     20$                     ;BRANCH IF NOT
023706  004737  026476          JSR     PC,PRTCLR               ;GO CLEAR THE OLD RESPONSES

023712  004737  024064  20$:    JSR     PC,CDRECV               ;GO CHECK FOR ANY NEW RESPONSES
023716  105737  010732          TSTB    IOSTAT                  ;IS THE I/O STATUS O.K. ?
023722  001052                  BNE     55$                     ;EXIT IF IT ISN'T

023724  032737  000020  003674  25$:    BIT     #GCSCFL,PCFLAG          ;IS THIS A GCS COMMAND ?
023732  001010                  BNE     30$                     ;YES, GO SETUP MINLIM
023734  032737  000100  003674  BIT     #CMDONE,PCFLAG          ;IS THIS A NULL COMMAND ?
023742  001042                  BNE     55$                     ;EXIT IF IT IS
023744  032763  000200  000000  BIT     #IMM,CMD(R3)            ;IS THIS AN IMMEDIATE COMMAND ?
023752  001404                  BEQ     35$                     ;NO, BRANCH
023754  112737  000001  010754  30$:    MOVB    #1,MINLIM               ;YES, SET MINIMUM LIMIT TO 1
023762  000403                  BR      40$                     ;BRANCH

023764  112737  000002  010754  35$:    MOVB    #2,MINLIM               ;NO, SET MINIMUM LIMIT TO 2
023772  123737  010755  010754  40$:    CMPB    CRDLIM,MINLIM           ;DO WE HAVE ENOUGH CREDITS ?
024000  103004                  BHIS    45$                     ;YES, KEEP GOING
024002  052737  020000  010732  BIS     #IOICRD,IOSTAT          ;SET INSUFFICIENT CREDIT IN I/O STATUS
024010  000417                  BR      55$                     ;GET OUT
```

```
3318
3319 024012  005264  000006          45$:  INC   CMDSEQ(R4)              ;ADD 1 TO THE COMMAND SEQUENCE NUMBER
3320 024016  032737  000020  003674        BIT   #GCSCFL,PCFLAG          ;IS IT A GCS COMMAND ?
3321 024024  001403                        BEQ   50$                     ;NO, BRANCH
3322 024026  016437  000006  010736        MOV   CMDSEQ(R4),GCSREF       ;SAVE THE COMMAND REFERENCE NUMBER
3323
3324 024034  105337  010755          50$:  DECB  CRDLIM                  ;SUBTRACT 1 FROM THE CREDIT LIMIT
3325 024040  004737  024342                JSR   PC,S.FPCK               ;GO FILL THE TMSCP PACKET
3326 024044  004737  026172                JSR   PC,PRTDRV               ;GO SEND THE COMMAND
3327
3328 024050  010564  000016          55$:  MOV   R5,COLSAV(R4)           ;SAVE R5 IN COMMAND OLD POINTER SAVE
3329 024054                                POP   <R5,R3>                 ;RESTORE R3, AND R5
3330 024060  000240                        NOP   ;TEMP
3331 024062  000207                        RTS   PC                      ;RETURN
```

```
                                .SBTTL  CLASS DRIVER RECEIVE
                                ;******************************
                                ;
                                ;Class Driver Receive
                                ;
                                ;Called By      : CLSDVR
                                ;Calls To       : PDRECV, PRTCLR
                                ;Inputs         : RESP   - The number of RESPONSEs found.
                                ;                 GCSREF - Get Command Status reference number.
                                ;                 RNUSAV - New response buffer save
                                ;                 CMSTSV - Command progress count.
                                ;                 ELBSAV - Error log buffer pointer.
                                ;Register Inputs: R2 - Old pointer to program command ring.
                                ;                 R3 - New pointer to program command ring.
                                ;                 R4 - Lun block pointer.
                                ;                 R5 - Old pointer to driver command ring.
                                ;Registers Used : R1 - Old pointer to driver RESPONSE ring.
                                ;
024064                          CDRECV::
024064                                  PUSH    <R1,R2>                 ;SAVE R1,R2
024070  016401  000020                  MOV     RNUSAV(R4),R1           ;LET R1 = NEW RESPONSE BUFFER SAVE
024074  004737  026352                  JSR     PC,PDRECV               ;CALL PORT DRIVER RECEIVE
024100  005737  003572                  TST     RESP                    ;DID WE GET A RESPONSE ?
024104  001506                          BEQ     35$                     ;NO, GET OUT OF HERE
024106  013737  003572  003556          MOV     RESP,HNDLRP             ;SAVE A COPY FOR RSPHDL
                                
024114  022761  000022  000012  5$:     CMP     @ST.SEX,P.STS(R1)       ;IS IT A SERIOUS EXCEPTION ?
024122  001425                          BEQ     10$                     ;YES, CONTINUE
024124  005761  000000                  TST     P.CRF(R1)               ;IS IT AN UNSOLICITED ERROR LOG ?
024130  001422                          BEQ     10$                     ;YES, GO HANDLE ERROR LOG
024132  026165  000000  000000          CMP     P.CRF(R1),P.CRF(R5)     ;IS THIS THE COMMAND THAT IS EXPECTED ?
024140  001416                          BEQ     10$                     ;YES, CONTINUE
024142  023761  010736  000000          CMP     GCSREF,P.CPF(R1)        ;IS IT THE GCS END PACKET
024150  001003                          BNE     7$                      ;NO, GO DO RESPONSE OUT OF SEQUENCE
024152  004737  027006                  JSR     PC,GCSHDL               ;GO TO THE GCS HANDLING ROUTINE
024156  000461                          BR      35$                     ;GET OUT
                                
024160  112737  000005  010732  7$:     MOVB    @MISSEQ,IOSTAT          ;SET MISSING SEQUENCE IN I/O STATUS
024166  004737  034022                  JSR     PC,CORDMP
024172  000240                          NOP
024174  000452                          BR      35$                     ;EXIT
                                
024176  032761  000200  000010  10$:    BIT     @OP.END,P.OPCD(R1)      ;YES, IS IT AN END PACKET ?
024204  001427                          BEQ     20$                     ;NO, GO HANDLE ERROR LOG
024206  052737  100000  010732          BIS     @NURESP,IOSTAT          ;SET A NEW RESPONSE IN THE I/O STATUS
024214  105237  010755                  INCB    CRDLIM                  ;ADD 1 TO THE CREDIT LIMIT
024220  062705  000050                  ADD     @DCBSTP,R5              ;ADJUST THE OLD COMMAND POINTER
024224  022705  004216                  CMP     @DCBEND,R5              ;IS IT AT THE END OF THE RING ?
024230  001002                          BNE     15$                     ;NO, BRANCH
024232  012705  003756                  MOV     @DCMDBF,R5              ;YES, SET IT TO THE TOP OF THE RING
                                
024236  016162  000012  000010  15$:    MOV     P.STS(R1),XFERST(R2)    ;PUT REPSONCE STATUS IN THE HOST PACKET
024244  062702  000014                  ADD     @PCBSTP,R2              ;ADJUST R2 TO POINT AT THE NEXT SLOT
024250  022702  003522                  CMP     @PCBEND,R2              ;IS IT AT THE END OF THE RING ?
024254  001006                          BNE     25$                     ;NO, BRANCH
024256  012702  003442                  MOV     @PCMDBF,R2              ;YES, SET IT BACK TO TOP OF THE RING
```

```
3390 024262  000403                          BR      25$                     ;BRANCH TO THE END
3391
3392 024264  052737  040000  010732  20$:    BIS     @ERRLOG,IOSTAT          ;SET ERROR LOG IN I/O STATUS
3393 024272  005337  003572          25$:    DEC     RESP                    ;SUBTRACT 1 FROM THE RESPONSE COUNT
3394 024276  062701  000104                  ADD     @DRBSTP,R1              ;ADJUST R1
3395 024302  026401  000212                  CMP     URBEND(R4),R1           ;IS IT AT THE END OF THE RING ?
3396 024306  001002                          BNE     30$                     ;NO, KEEP GOING
3397 024310  016401  000210                  MOV     URSPBF(R4),R1           ;YES, SET IT TO BEGINNING OF THE RING
3398
3399 024314  005737  003572           30$:   TST     RESP                    ;HAVE WE DONE ALL THE RESPONSES ?
3400 024320  001275                          BNE     5$                      ;NO, DO IT AGAIN
3401
3402 024322  005037  003572          35$:    CLR     RESP                    ;CLEAR NOW IN CASE WE MADE ERROR EXIT
3403 024326  010164  000020                  MOV     R1,RNUSAV(R4)           ;SAVE THE NEW RESPONSE BUFFER POINTER
3404 024332                                  POP     <R2,R1>                 ;RESTORE R2,R1
3405 024336  000240                          NOP     ;TEMP
3406 024340  000207                          RTS     PC                      ;RETURN
```

```
                                        .SBTTL  COMMAND STUFFER
                                        ;******************************
                                        ;
                                        ; Stuff TMSCP Command Packet
                                        ;
                                        ;Called By      : CLSDRV
                                        ;Inputs         : CNUSAV - Points to next slot in the driver command ring.
                                        ;                 CMDSEQ - Number appended to each command packet.
                                        ;                 GCSREF - Get Command Status reference number.
                                        ;                 SEREXP - Flag set non-zero on occurrence of a serious exception.
                                        ;Outputs        : PCKSIZ - Length in bytes of the command packet.
                                        ;Register Inputs: R3 - New pointer to program command ring.
                                        ;                 R4 - Lun block pointer.
                                        ;                 R5 - Old pointer to driver command ring.
                                        ;Registers Used : R1 - New pointer to driver command ring.

024342                                  STFPCK::
024342                                          PUSH    <R1,R2>                 ;SAVE R1 AND R2
024346  005037  010750                          CLR     PCKSIZ                  ;CLEAR PACKET SIZE
024352  016401  000014                          MOV     CNUSAV(R4),R1           ;LET R1 EQUAL THE NEW COMMAND POINTER
024356  016461  000006  000000                  MOV     CMDSEQ(R4),P.CRF(R1)    ;PUT COMMAND SEQUENCE NUMBER INTO PACKET
024364  005061  000002                          CLR     P.CRF+2(R1)             ;CLEAR THE UPPER WORD
024370  016461  000004  000004                  MOV     TKUNIT(R4),P.UNIT(R1)   ;PUT THE UNIT NUMBER INTO THE PACKET
024376  005061  000006                          CLR     P.UNIT+2(R1)            ;CLEAR THE UPPER WORD
024402  005061  000012                          CLR     P.MOD(R1)               ;CLEAR MODIFIERS FIELD
024406  032737  000020  003674                  BIT     #GCSCFL,PCFLAG          ;ARE WE IN GCS COMMAND MODE ?
024414  001402                                  BEQ     5$                      ;NO, CONTINUE
024416  000137  025654                          JMP     GCMDST                  ;YES, GO DO A GET COMMAND STATUS

024422  016361  000002  000014  5$:             MOV     ITMOFF(R3),P.BCNT(R1)   ;PUT THE BYTE COUNT INTO THE PACKET
024430  005061  000016                          CLR     P.BCNT+2(R1)            ;CLEAR THE UPPER WORD
024434  016337  000000  003710                  MOV     CMD(R3),R8              ;PUT THE PRIMITIVE IN R8
024442  042737  177770  003710                  BIC     #177770,R8              ;GET JUST THE MODIFIERS
024450  022737  000001  003710                  CMP     #REVBIT,R8              ;IS THE COMMAND A REVERSE ?
024456  001003                                  BNE     10$                     ;NO, BRANCH
024460  052761  000010  000012                  BIS     #MD.REV,P.MOD(R1)       ;YES, SET REVERSE IN THE MODIFIER FIELD

024466  032764  000002  000026  10$:            BIT     #SEREXC,LUNFLG(R4)      ;IS IT A SERIOUS EXCEPTION CONDITION ?
024474  001406                                  BEQ     15$                     ;NO, BRANCH
024476  052761  020000  000012                  BIS     #MD.CSE,P.MOD(R1)       ;YES, SET CLEAR SERIOUS EXCEPTION
024504  042764  000002  000026                  BIC     #SEREXC,LUNFLG(R4)      ;CLEAR SERIOUS EXCEPTION FLAG

024512  116302  000000          15$:            MOVB    CMD(R3),R2              ;PUT THE COMMAND PRIMITIVE INTO R1
024516  006202                                  ASR     R2                      ;
024520  006202                                  ASR     R2                      ;
024522  042702  177701                          BIC     #^C76,R2                ;ADJUST FOR THE CASE STATEMENT
024526  022702  000046                          CMP     #46,R2                  ;ARE WE IN THE RANGE ?
024532  103002                                  BHIS    20$                     ;YES, KEEP GOING
024534  000137  026044                          JMP     ILCMD                   ;NO, HANDLE AN ILLEGAL COMMAND
024540  000172  024544          20$:            JMP     @CMDTBL(R2)             ;SELECT

024544  024614                  CMDTBL: .WORD   NULL
024546  024620                          .WORD   READ
024550  024662                          .WORD   WRITE
024552  024724                          .WORD   CHODAT
024554  024744                          .WORD   ACCESS
024556  024764                          .WORD   SPCREC
```

```
3465 024560  025032                          .WORD   SKP,MK
3466 024562  025106                          .WORD   SPCOBJ
3467 024564  025162                          .WORD   WTAPMK
3468 024566  025210                          .WORD   ERASE
3469 024570  025246                          .WORD   ERASGP
3470 024572  025266                          .WORD   AVALAB
3471 024574  025324                          .WORD   ONLINE
3472 024576  025422                          .WORD   SUNCHR
3473 024600  025520                          .WORD   REWIND
3474 024602  025616                          .WORD   INIT
3475 024604  025622                          .WORD   ABOR
3476 024606  025654                          .WORD   GCMDST
3477 024610  025714                          .WORD   GUNSTA
3478 024612  025742                          .WORD   SCNTCH
3479
3480 024614  000137  026140          NULL:   JMP     COMEXI                  ;EXIT
3481
3482 024620  012761  000041  000010  READ:   MOV     #OP.RD,P.OPCD(R1)       ;PUT THE READ OPCODE INTO THE PACKET
3483 024626  012737  000034  010750          MOV     #34,PCKSIZ              ;PUT THE PACKET SIZE INTO THE PACKET
3484 024634  052761  000400  000012          BIS     #MD.SER,P.MOD(R1)       ;DISALLOW AUTO RETRIES
3485 024642  005737  003564                  TST     ARETRY                  ;ARE WE DOING AUTO RETRIES ?
3486 024646  001403                          BEQ     1$                      ;NO, GET OUT
3487 024650  042761  000400  000012          BIC     #MD.SER,P.MOD(R1)       ;ALLOW AUTO RETRIES
3488 024656  000137  026056          1$:     JMP     BUFDSC                  ;GOTO THE BUFFER DESCRIPTOR ROUTINE
3489
3490 024662  012761  000042  000010  WRITE:  MOV     #OP.WR,P.OPCD(R1)       ;PUT THE WRITE OPCODE INTO THE PACKET
3491 024670  012737  000034  010750          MOV     #34,PCKSIZ              ;PUT THE PACKET SIZE INTO THE PACKET
3492 024676  052761  000400  000012          BIS     #MD.SER,P.MOD(R1)       ;DISALLOW AUTO RETRIES
3493 024704  005737  003564                  TST     ARETRY                  ;ARE WE DOING AUTO RETRIES ?
3494 024710  001403                          BEQ     1$                      ;NO, GET OUT
3495 024712  042761  000400  000012          BIC     #MD.SER,P.MOD(R1)       ;ALLOW AUTO RETRIES
3496 024720  000137  026056          1$:     JMP     BUFDSC                  ;GOTO THE BUFFER DESCRIPTOR ROUTINE
3497
3498 024724  012761  000040  000010  CHODAT: MOV     #OP.CMP,P.OPCD(R1)      ;PUT COMPARE HOST DATA OPCODE IN PACKET
3499 024732  012737  000034  010750          MOV     #34,PCKSIZ              ;PUT THE PACKET SIZE INTO THE PACKET
3500 024740  000137  026056                  JMP     BUFDSC                  ;GOTO THE BUFFER DESCRIPTOR ROUTINE
3501
3502 024744  012761  000020  000010  ACCESS: MOV     #OP.ACC,P.OPCD(R1)      ;PUT THE ACCESS OPCODE INTO THE PACKET
3503 024752  012737  000020  010750          MOV     #20,PCKSIZ              ;PUT THE PACKET SIZE INTO THE PACKET
3504 024760  000137  026110                  JMP     SUPRES                  ;GOTO THE SUPPRESS ROUTINE
3505
3506 024764  012761  000045  000010  SPCREC: MOV     #OP.REP,P.OPCD(R1)      ;PUT REPOSITION OPCODE INTO THE PACKET
3507 024772  005061  000020                  CLR     P.TMGC(R1)              ;CLEAR THE TAPE MARK COUNT
3508 024776  005061  000022                  CLR     P.TMGC+2(R1)            ;CLEAR THE UPPER WORD
3509 025002  032737  000002  003710          BIT     #EOTBIT,R8              ;IS THE DETECT LEOT BIT SET ?
3510 025010  001403                          BEQ     70$                     ;NO,CONTINUE
3511 025012  052761  000200  000012          BIS     #MD.DLE,P.MOD(R1)       ;YES, SET DETECT LEOT IN THE MODIFIER
3512 025020  012737  000024  010750  70$:    MOV     #24,PCKSIZ              ;PUT THE PACKET SIZE INTO THE PACKET
3513 025026  000137  026110                  JMP     SUPRES                  ;GOTO THE SUPPRESS ROUTINE
3514
3515 025032  012761  000045  000010  SKPTMK: MOV     #OP.REP,P.OPCD(R1)      ;PUT THE REPOSITION OPCODE IN PACKET
3516 025040  016161  000014  000020          MOV     P.BCNT(R1),P.TMGC(R1)   ;PUT THE TAPE MARK COUNT IN PACKET
3517 025046  005061  000022                  CLR     P.TMGC+2(R1)            ;CLEAR THE TAPE MARK FIELD
3518 025052  005061  000014                  CLR     P.BCNT(R1)              ;CLEAR THE UPPER WORD
3519 025056  032737  000002  003710          BIT     #EOTBIT,R8              ;IS THE DETECT LEOT BIT SET ?
3520 025064  001403                          BEQ     100$                    ;NO,CONTINUE
3521 025066  052761  000200  000012          BIS     #MD.DLE,P.MOD(R1)       ;YES, SET DETECT LEOT IN THE MODIFIER
```

```
3522 025074  012737  000024  010750  100$:   MOV     #24,PCKSIZ              ;PUT THE PACKET SIZE INTO THE PACKET
3523 025102  000137  026110                  JMP     SUPRES                  ;GOTO THE SUPPRESS ROUTINE
3524
3525 025106  012761  000045  000010  SPCOBJ: MOV     #OP.REP,P.OPCD(R1)      ;PUT THE REPOSITION OPCODE IN PACKET
3526 025114  052761  000004  000012          BIS     #MD.OBC,P.MOD(R1)       ;SET THE OBJECT BIT IN THE MODIFIER
3527 025122  005061  000020                  CLR     P.TMGC(R1)              ;CLEAR THE TAPE MARK FIELD
3528 025126  005061  000022                  CLR     P.TMGC+2(R1)            ;CLEAR THE UPPER WORD
3529 025132  032764  000200  000026          BIT     #RETFLG,LUNFLG(R4)      ;ARE WE DOING RETRIES ?
3530 025140  001403                          BEQ     1$                      ;BRANCH IF NOT
3531 025142  012761  000001  000014          MOV     #1,P.BCNT(R1)           ;SET UP TO DO 1 RECORD
3532 025150  012737  000024  010750  1$:     MOV     #24,PCKSIZ              ;PUT THE PACKET SIZE INTO THE PACKET
3533 025156  000137  026110                  JMP     SUPRES                  ;GOTO THE SUPPRESS ROUTINE
3534
3535 025162  012761  000044  000010  WTAPMK: MOV     #OP.WTM,P.OPCD(R1)      ;PUT WRITE TAPE MARK OPCODE IN PACKET
3536 025170  052761  020000  000012          BIS     #MD.CSE,P.MOD(R1)       ;YES, SET CLEAR SERIOUS EXCEPTION
3537 025176  012737  000014  010750          MOV     #14,PCKSIZ              ;PUT THE PACKET SIZE INTO THE PACKET
3538 025204  000137  026140                  JMP     COMEXI                  ;GOTO THE EXIT
3539
3540 025210  012761  000022  000010  ERASE:  MOV     #OP.ERS,P.OPCD(R1)      ;PUT THE ERASE OPCODE INTO THE PACKET
3541 025216  C22737  000003  003710          CMP     #IMMBIT,R8              ;IS THE IMMEDIATE BIT SET ?
3542 025224  001403                          BEQ     20$                     ;NO,CONTINUE
3543 025226  052761  000100  000012          BIS     #MD.IMM,P.MOD(R1)       ;YES, SET IMMEDIATE IN THE MODIFIER
3544 025234  012737  000014  010750  20$:    MOV     #14,PCKSIZ              ;PUT THE PACKET SIZE INTO THE PACKET
3545 025242  000137  026140                  JMP     COMEXI                  ;GOTO THE EXIT
3546
3547 025246  012761  000026  000010  ERASGP: MOV     #OP.ERG,P.OPCD(R1)      ;PUT ERASE GAP OPCODE INTO THE PACKET
3548 025254  012737  000014  010750          MOV     #14,PCKSIZ              ;PUT THE PACKET SIZE INTO THE PACKET
3549 025262  000137  026140                  JMP     COMEXI                  ;GOTO THE EXIT
3550
3551 025266  012761  000010  000010  AVALAB: MOV     #OP.AVL,P.OPCD(R1)      ;PUT AVAILABLE OPCODE INTO THE PACKET
3552 025274  022737  000004  003710          CMP     #UNLBIT,R8              ;IS THE UNLOAD BIT SET ?
3553 025302  001403                          BEQ     10$                     ;NO,CONTINUE
3554 025304  052761  000020  000012          BIS     #MD.UNL,P.MOD(R1)       ;YES, SET UNLOAD IN THE MODIFIER FIELD
3555 025312  012737  000014  010750  10$:    MOV     #14,PCKSIZ              ;PUT THE PACKET SIZE INTO THE PACKET
3556 025320  000137  026140                  JMP     COMEXI                  ;GOTO THE EXIT
3557
3558 025324  012761  000011  000010  ONLINE: MOV     #OP.ONL,P.OPCD(R1)      ;PUT THE ONLINE OPCODE INTO THE PACKET
3559 025332  005061  000014                  CLR     P.UNFL-2(R1)            ;CLEAR THE UNIT FLAG FIELD
3560 025336  005061  000016                  CLR     P.UNFL(R1)              ;
3561 025342  005061  000020                  CLR     P.UNFL+2(R1)            ;
3562 025346  005061  000022                  CLR     P.UNFL+4(R1)            ;
3563 025352  005061  000024                  CLR     P.UNFL+6(R1)            ;
3564 025356  005061  000026                  CLR     P.UNFL+10(R1)           ;
3565 025362  005061  000030                  CLR     P.UNFL+12(R1)           ;
3566 025366  005061  000032                  CLR     P.UNFL+14(R1)           ;
3567 025372  005061  000034                  CLR     P.DVPM(R1)              ;CLEAR THE DEVICE PARAMETER FIELD
3568 025376  013761  003732  000040          MOV     FORMAT,P.FORM(R1)       ;PUT THE TAPE FORMAT INTO THE PACKET
3569 025404  005061  000042                  CLR     P.SPED(R1)              ;CLEAR THE SPEED FIELD
3570 025410  012737  000044  010750          MOV     #44,PCKSIZ              ;PUT THE PACKET SIZE INTO THE PACKET
3571 025416  000137  026140                  JMP     COMEXI                  ;GOTO THE EXIT
3572
3573 025422  012761  000012  000010  SUNCHR: MOV     #OP.SUC,P.OPCD(R1)      ;SET UNIT CHARA. OPCODE INTO THE PACKET
3574 025430  005061  000014                  CLR     P.UNFL 2(R1)            ;CLEAR THE UNIT FLAG FIELD
3575 025434  005061  000016                  CLR     P.UNFL(R1)              ;
3576 025440  005061  000020                  CLR     P.UNFL+2(R1)            ;
3577 025444  005061  000022                  CLR     P.UNFL+4(R1)            ;
3578 025450  005061  000024                  CLR     P.UNFL+6(R1)            ;
```

```
3579 025454  005061  000026                    CLR     P.UNFL+10(R1)           ;
3580 025460  005061  000030                    CLR     P.UNFL+12(R1)           ;
3581 025464  005061  000032                    CLR     P.UNFL+14(R1)           ;
3582 025470  005061  000034                    CLR     P.DVPM(R1)              ;CLEAR THE DEVICE PARAMETERS FIELD
3583 025474  013761  003732  000040           MOV     FORMAT,P.FORM(R1)       ;PUT THE TAPE FORMAT INTO THE PACKET
3584 025502  005061  000042                    CLR     P.SPED(R1)              ;CLEAR THE SPEED FIELD
3585 025506  012737  000044  010750           MOV     #44,PCKSIZ              ;PUT THE PACKET SIZE INTO THE PACKFT
3586 025514  000137  026140                    JMP     COMEXI                  ;GOTO THE EXIT
3587
3588 025520  012761  000045  000010  REWIND:  MOV     #OP.REP,P.OPCD(R1)      ;PUT THE REPOSTITION OPCODE INTO PACKET
3589 025526  052761  020002  000012           BIS     #MD.CSE!MD.RWD,P.MOD(R1) ;SET THE REWIND MODIFIER
3590 025534  022737  000003  003710           CMP     #IMMBIT,R8              ;IS THE IMMEDIATE BIT SET
3591 025542  001003                            BNE     60$                     ;NO,CONTINUE
3592 025544  052761  000100  000012           BIS     #MD.IMM,P.MOD(R1)       ;YES, SET THE IMMEDIATE MODIFIER
3593 025552  005061  000020          60$:     CLR     P.TMGC(R1)              ;CLEAR THE TAPE MARK
3594 025556  005061  000022                    CLR     P.TMGC+2(R1)            ;     COUNT FIELD
3595 025562  005061  000014                    CLR     P.BCNT(R1)              ;CLEAR THE BYTE COUNT
3596 025566  005061  000016                    CLR     P.BCNT+2(R1)            ;      FIELD
3597 025572  012737  000024  010750           MOV     #24,PCKSIZ              ;PUT THE PACKET SIZE INTO THE PACKET
3598 025600  005064  000006                    CLR     CMDSEQ(R4)              ;RESET THE COMMAND SEQUENCE NUMBER
3599 025604  042764  000010  000026           BIC     #EOTPR,LUNFLG(R4)       ;CLEAT THE EOT PRINT FLAG
3600 025612  000137  026140                    JMP     COMEXI                  ;GOTO THE EXIT
3601
3602 025616  000137  026162          INIT:    JMP     EXIT                    ;EXIT
3603
3604 025622  012761  000001  000010  ABOR:    MOV     #OP.ABO,P.OPCD(R1)      ;PUT THE ABORT OPCODE INTO THE PACKET
3605 025630  016561  000000  000014           MOV     P.CRF(R5),P.OTRF(R1)    ;PUT THE OLD CRN INTO THE PACKET
3606 025636  005061  000016                    CLR     P.OTRF+2(R1)            ;CLEAR THE UPPER WORD
3607 025642  012737  000020  010750           MOV     #20,PCKSIZ              ;PUT THE PACKET SIZE INTO THE PACKET
3608 025650  000137  026140                    JMP     COMEXI                  ;GOTO THE EXIT
3609
3610 025654  012761  000002  000010  GCMDST:  MOV     #OP.GCS,P.OPCD(R1)      ;PUT GCS OPCODE INTO THE PACKET
3611 025662  016561  000000  000014           MOV     P.CRF(R5),P.OTRF(R1)    ;PUT THE OLD CRN INTO THE PACKET
3612 025670  005061  000016                    CLR     P.OTRF+2(R1)            ;CLEAR THE UPPER WORD
3613 025674  012737  000020  010750           MOV     #20,PCKSIZ              ;PUT THE PACKET SIZE INTO THE PACKET
3614 025702  042737  000020  003674           BIC     #GCSCFL,PCFLAG          ;CLEAR GCS COMMAND MODE ?
3615 025710  000137  026140                    JMP     COMEXI                  ;GOTO THE EXIT
3616
3617 025714  012761  000003  000010  GUNSTA:  MOV     #OP.GUS,P.OPCD(R1)      ;PUT THE GUS OPCODE INTO THE PACKET
3618 025722  042761  020000  000012           BIC     #MD.CSE,P.MOD(R1)       ;CLEAR CLEAR SERIOUS EXCEPTION MODIFIER
3619 025730  012737  000014  010750           MOV     #14,PCKSIZ              ;PUT THE PACKET SIZE INTO THE PACKET
3620 025736  000137  026140                    JMP     COMEXI                  ;GOTO THE EXIT
3621
3622 025742  012761  000004  000010  SCNTCH:  MOV     #OP.SCC,P.OPCD(R1)      ;PUT THE SCC OPCODE INTO THE PACKET
3623 025750  005061  000004                    CLR     P.UNIT(R1)              ;CLEAR THE UNIT NUMBER
3624 025754  012761  000000  000014           MOV     #MSCPVR,P.VRSN(R1)      ;PUT THE MSCP VERSION INTO THE PACKET
3625 025762  013761  010746  000016           MOV     CNTFLG,P.CNTF(R1)       ;PUT CONTROLLER FLAGS INTO THE PACKET
3626 025770  012761  000000  000020           MOV     #HSTIMO,P.HTMO(R1)      ;PUT THE HOST TIMEOUT INTO THE PACKET
3627 025776  005061  000022                    CLR     P.HTMO+2(R1)            ;CLEAR THE TIME FIELD
3628 026002  005061  000024                    CLR     P.TIME(R1)              ;
3629 026006  005061  000026                    CLR     P.TIME+2(R1)            ;
3630 026012  005061  000030                    CLR     P.TIME+4(R1)            ;
3631 026016  005061  000032                    CLR     P.TIME+6(R1)            ;
3632 026022  005061  000034                    CLR     P.CTPM(R1)              ;CLEAR THE FIRDT WORD
3633 026026  005061  000036                    CLR     P.CTPM+2(R1)            ;CLEAR THE SECOND WORD
3634 026032  012737  000040  010750  5$:      MOV     #40,PCKSIZ              ;PUT THE PACKET SIZE INTO THE PACKET
3635 026040  000137  026140                    JMP     COMEXI                  ;GOTO THE EXIT
```

```
3636
3637 026044  052737  000007  010732  ILCMD:  BIS     #ILLCMD,IOSTAT          ;SET ILLCMD IN THE I/O STATUS
3638 026052  000137  026162                  JMP     EXIT                    ;GOTO THE ERROR EXIT
3639
3640 026056  016361  000012  000020  BUFDSC: MOV     BUFOFF(R3),P.BUFF(R1)   ;PUT THE BUFFER ADDRESS INTO THE PACKET
3641 026064  005061  000022                  CLR     P.BUFF+2(R1)            ;CLEAR THE REST OF THE BUFFER FIELD
3642 026070  005061  000024                  CLR     P.BUFF+4(R1)            ;
3643 026074  005061  000026                  CLR     P.BUFF+6(R1)            ;
3644 026100  005061  000030                  CLR     P.BUFF+10(R1)           ;
3645 026104  005061  000032                  CLR     P.BUFF+12(R1)           ;
3646
3647 026110  105737  002224          SUPRES: TSTB    SERCOR                  ;IS SUPPRESS ERROR CORRECTION ENABLED ?
3648 026114  001003                  105$:   BNE     105$                    ;NO
3649 026116  052761  001000  000012          BIS     #MD.SEC,P.MOD(R1)       ;YES, SET SEC MODIFIER
3650 026124  105737  002225                  TSTB    SERREC                  ;IS SUPPRESS ERROR RECOVERY ENABLED ?
3651 026130  001003                          BNE     COMEXI                  ;NO
3652 026132  052761  000400  000012          BIS     #MD.SER,P.MOD(R1)       ;YES, SET THE SER MODIFIER
3653
3654 026140  062701  000050          COMEXI: ADD     #DCBSTP,R1              ;SET THE POINTER TO THE NEXT SLOT
3655 026144  022701  004216                  CMP     #DCBEND,R1              ;ARE WE AT THE END OF THE RING ?
3656 026150  001002                          BNE     110$                    ;NO, EXIT
3657 026152  012701  003756                  MOV     #DCMDBF,R1              ;YES, SET THE POINTER TO START OF RING
3658 026156  010164  000014          110$:   MOV     R1,CNUSAV(R4)           ;SAVE THE POINTER
3659
3660 026162                          EXIT:   POP     <R2,R1>                 ;RESTORE R1
3661 026166  000240                          NOP     ;TEMP
3662 026170  000207                          RTS     PC                      ;RETURN
```

```
                                        .SBTTL  PORT DRIVER TRANSMIT
                                        ;******************************
                                        ;
                                        ;Port Driver Transmit
                                        ;
                                        ;Called By       : CLSDVR
                                        ;Inputs          : CMDSSV - Command descriptor ring pointer.
                                        ;                  DCDSAV - Driver command ring pointer.
                                        ;                  CRDLIM - Number of open slots in the driver command ring.
                                        ;                  PCKSIZ - Length in bytes of the command packet.
                                        ;Register Inputs: R4 - Lun block pointer.
                                        ;Registers Used : R2 - Pointer to driver command ring.
                                        ;                  R1   Pointer to driver command descriptor ring.
                                        ;
                                        ;
026172                                  PRTDRV::
026172                                          PUSH    <R3,R2,R1>              ;SAVE R3, R2 AND R1
026200  016402  000014                          MOV     CMUSAV(R4),R2           ;POINT R2 AT NEW COMMAND BUFFER SLOT
026204  016401  000012                          MOV     CMDSSV(R4),R1           ;LET R1 POINT TO THE COMMAND DESCRIPTOR
026210  022702  003756                          CMP     #DCMDBF,R2              ;IS R2 AT TOP OF DRIVER COMMAND Ring
026214  001403                                  BEQ     1$                      ;YES, BRANCH
026216  162702  000050                          SUB     #DCBSTP,R2              ;NO, SUBTRACT DCBSTP FROM R2
026222  000402                                  BR      5$                      ;
                                        ;
026224  062702  000170              1$:         ADD     #DCB3SP,R2              ;YES, ADD DCB3SP TO R2
026230  113762  010755  177776      5$:         MOVB    CRDLIM,CRD(R2)          ;PUT THE CREDIT LIMIT INTO THE PACKET
026236  112762  000001  177777                  MOVB    #1,CONID(R2)            ;PUT THE CONNECTION TYPE INTO THE PACKET
026244  013762  010750  177774                  MOV     PCKSIZ,MSGLEN(R2)       ;PUT THE PACKET SIZE INTO THE PACKET
026252  010211                                  MOV     R2,(R1)                 ;PUT THE PACKET ADDRESS INTO THE DESCRIPTOR
026254  052761  100000  000002                  BIS     #OWN,HIADDR(R1)         ;SET THE OWNERSHIP BIT OF THE DESCRIPTOR
026262  042761  040000  000002                  BIC     #FLAG,HIADDR(R1)        ;CLEAR TO DESCRIPTOR FLAG BIT
026270  005774  000000                          TST     @TKIP(R4)               ;READ THE IP REGISTER
026274  017437  000002  010752                  MOV     @TKSA(R4),SAERR         ;SAVE THE SA FOR THE ERROR PRINTOUT
026302  005737  010752                          TST     SAERR                   ;READ THE SA REGISTER
026306  100003                                  BPL     10$                     ;BRANCH IF NO ERRORS
026310  052737  000003  010732                  BIS     #IOPDRE,IOSTAT          ;SET PORT DETECTED ERROR IN I/O STATUS
                                        ;
026316  062701  000004              10$:        ADD     #DSPSTP,R1              ;ADVANCE THE DESCRIPTOR POINTER
026322  026401  000222                          CMP     UCDEND(R4),R1           ;ARE WE AT END OF THE DESCRIPTOR RING
026326  001002                                  BNE     15$                     ;NO, BRANCH
026330  016401  000220                          MOV     UCDSRG(R4),R1           ;YES, SET POINTER TO START OF THE RING
                                        ;
026334  010164  000012              15$:        MOV     R1,CMDSSV(R4)           ;SAVE THE POINTER
026340                                          POP     <R1,R2,R3>              ;RESTORE R1, R2 AND R3
026346  000240                                  NOP     ;TEMP
026350  000207                                  RTS     PC                      ;RETURN
```

```
                                        .SBTTL  PORT DRIVER RECEI\E
                                        ;******************************
                                        ;
                                        ;Port Driver Receive
                                        ;
                                        ;Called By      : CDRECV
                                        ;Inputs         : URDSRG - RESPONSE descriptor ring.
                                        ;                 UCDSRG - Command descriptor ring.
                                        ;Outputs        : RESP   - Number of new RESPONSES.
                                        ;Registers Used : R1   RESPONSE descriptor ring pointer.
                                        ;

026352                                  PDRECV::
026352                                          PUSH    <R1>                    ;SAVE R1
026354  016401  000214                          MOV     URDSRG(R4),R1           ;SET R1 TO THE RESPONSE DESCRIPTOR
026360  017437  000002  010752                  MOV     @TKSA(R4),SAERR         ;SAVE THE SA FOR THE ERROR PRINTOUT
026366  005737  010752                          TST     SAERR                   ;READ THE SA REGISTER
026372  100003                                  BPL     1$                      ;BRANCH IF NO ERRORS
026374  052737  000003  010732                  BIS     #IOPDRE,IOSTAT          ;SET PORT DETECTED ERROR IN I/O STATUS

026402  006364  000010                  1$:     ASL     SLTUSE(R4)              ;SHIFT BITMAP
026406  032737  000040  003674                  BIT     #GCSRFL,PCFLAG          ;ARE WE IN GCS MODE ?
026414  001403                                  BEQ     2$                      ;NO, DO ALL RESPONSES
026416  005737  003572                          TST     RESP                    ;HAVE WE GOTTEN A RESPONSE ?
026422  001012                                  BNE     5$                      ;YES, GCS MODE ALLOW ONLY 1 RESPONSE

026424  032761  100000  000002  2$:     BIT     #OWN,HIADDR(R1)         ;IS THE SLOT SET TO US ?
026432  001006                                  BNE     5$                      ;NO , BRANCH
026434  005237  003572                          INC     RESP                    ;ADD 1 TO THE RESPONSE COUNT
026440  052764  000001  000010                  BIS     #BIT0,SLTUSE(R4)        ;SET SLOT IN-USE
026446  000403                                  BR      10$

026450  042764  000001  000010  5$:     BIC     #BIT0,SLTUSE(R4)        ;ELSE CLEAR THIS SLOT IN-USE
026456  062701  000004                  10$:    ADD     #DSPSTP,R1              ;SET THE POINTER TO THE NEXT SLOT
026462  026401  000216                          CMP     URDEND(R4),R1           ;ARE WE AT THE END OF THE RING ?
026466  001345                                  BNE     1$                      ;NO, KEEP GOING TILL WE GET THEM ALL
026470                                          POP     <R1>                    ;RESTORE R1
026472  000240                                  NOP     ;TEMP
026474  000207                                  RTS     PC                      ;RETURN
```

```
                                .SBTTL  PORT DRIVER CLEAR
                                ;******************************
                                ;
                                ;Port Driver Clear
                                ;
                                ;Called By      : CDRECV
                                ;Register Inputs: R4   Lun block pointer.
                                ;Registers Used : R1   Current location in the RESPONSE descriptor ring.
                                ;

026476                          PRTCLR::
026476                                  PUSH    <R1,R2>                 ;SAVE R1 AND R2
026502  016401  000216                  MOV     URDEND(R4),R1           ;R1 = END OF RESPONSE DESCRIPTOR RING
026506  016402  000214                  MOV     URDSRG(R4),R2           ;R2 = RESPONSE DESCRIPTOR RING
026512  162702  000004                  SUB     #4,R2                   ;BACK UP POINTER BY A LONGWORD

026516  162701  000004          1$:     SUB     #4,R1                   ;BACK UP POINTER BY A LONGWORD
026522  020201                          CMP     R2,R1                   ;BACKED UP PAST START OF RING?
026524  001410                          BEQ     20$                     ;YES - SO GET OUT
026526  000241                          CLC                             ;CLEAR THE CARRY
026530  006064  000010                  ROR     SLTUSE(R4)              ;MOVE BIT0 TO CARY BIT
026534  103003                          BCC     5$                      ;BRANCH IF SLOT NOT USED
026536  012761  100000  000002          MOV     #OWN,HIADDR(R1)         ;GIVE SLOT BACK TO PORT

026544  000764                  5$:     BR      1$                      ;LOOK FOR MORE
026546                          20$:    POP     <R2,R1>                 ;RESTORE R2 AND R1
026552  000240                          NOP     ;TEMP
026554  000207                          RTS     PC                      ;RETURN
```

```
                                .SBTTL  PORT DRIVER INITIALIZATION
                                ;*******************************
                                ;
                                ;Port Driver Initialization
                                ;
                                ;Called By      : CLSDRV
                                ;Register Inputs: R4 - Lun block pointer.
                                ;Registers Used : R1 - Current in't step in process
                                ;                 R2 - Used by the watchdog timer
                                ;                 R3   Initialization data table pointer
                                ;

026556                           PRTINT::
026556                                  PUSH    <R1,R2,R3,R5>           ;SAVE R1, R2, R3 AND R5
026566  010174  000000                  MOV     R1,@TKIP(R4)            ;INITIALIZE THE DRIVE
026572  016437  000214  027000          MOV     URDSRG(R4),INTTBL+2     ;PUT RESP DESCRIPTOR ADDRESS IN TABLE
026600  012703  026776                  MOV     #INTTBL,R3              ;PUT THE TABLE ADDRESS INTO R3
026604  012701  104000                  MOV     #S1!ERR,R1              ;SET UP TO BEGIN AT STEP 1

026610  012737  000050  010740  LOOP:   MOV     #40.,CNTHI              ;SET UP THE TIME OUT COUNTER
026616  005002                          CLR     R2                      ;CLEAR R2

026620  005202                  ILOOP:  INC     R2                      ;INCREMENT HI TIME OUT VALUE ?
026622  001003                          BNE     2$                      ;IF NOT, BRANCH
026624  005337  010740                  DEC     CNTHI                   ;ELSE, INCREMENT HI TIMEOUT
026630  001447                          BEQ     TKERR                   ;GET OUT, WE'VE TIMED OUT

026632  037401  000002          2$:     BIT     @TKSA(R4),R1            ;TEST FOR STEP BIT FROM DRIVE
026636  001770                          BEQ     ILOOP                   ;LOOP UNTIL SOMETHING SETS
026640  017437  000002  010752          MOV     @TKSA(R4),SAERR         ;SAVE THE SA FOR THE ERROR PRINTOUT
026646  005737  010752                  TST     SAERR                   ;CHECK FOR ERROR
026652  100436                          BMI     TKERR                   ;GET OUT ON ERROR
026654  012374  000002          3$:     MOV     (R3)+,@TKSA(R4)         ;WRITE WORD FROM TABLE TO CONTROLLER
026660  006301                          ASL     R1                      ;SHIFT TO NEXT STEP
026662  100403                          BMI     4$                      ;GET OUT AFTER FOURTH STEP
026664  052701  100000                  BIS     #ERR,R1                 ;ALSO CHECK FOR ERROR BIT
026670  000747                          BR      LOOP                    ;IF NOT AT LAST STEP LOOP

026672  016402  000214          4$:     MOV     URDSRG(R4),R2           ;PUT THE RESPONSE DESCRIPTOR ADD IN R2
026676  016403  000210                  MOV     URSPBF(R4),R3           ;PUT THE RESPONSE BUFFER ADDRESS IN R3
026702  010322                  5$:     MOV     R3,(R2)+                ;PUT THE BUFF ADD IN THE DESCRIPTOR
026704  005022                          CLR     (R2)+                   ;CLEAR THE NEXT WORD
026706  062703  000104                  ADD     #DRBSTP,R3              ;STEP TO THE NEXT BUFFER SLOT
026712  026403  000212                  CMP     URBEND(R4),R3           ;ARE WE AT THE END OF THE BUFFER ?
026716  001371                          BNE     5$                      ;NO, KEEP GOING

026720  016402  000220                  MOV     UCDSRG(R4),R2           ;PUT THE CMD DESCRIPTOR ADDRESS IN R2
026724  012703  003756                  MOV     #DCMDBF,R3              ;PUT THE CMD BUFFER ADDRESS IN R3
026730  010322                  10$:    MOV     R3,(R2)+                ;PUT THE BUFF ADD IN THE DESCRIPTOR
026732  005022                          CLR     (R2)+                   ;CLEAR THE NEXT WORD
026734  062703  000050                  ADD     #DCBSTP,R3              ;STEP TO THE NEXT BUFFER SLOT
026740  022703  004216                  CMP     #DCBEND,R3              ;ARE WE AT THE END OF THE BUFFER ?
026744  001371                          BNE     10$                     ;NO, KEEP GOING
026746  000403                          BR      IDONE                   ;ALL DONE

026750  012737  000006  010732  TKERR:  MOV     #INTERR,IOSTAT          ;SET UP FOR A FATAL ERROR
```

```
3838 026756  005337  003540          IDONE:  DEC     RSPCNT
3839 026762                                  POP     <R5,R3,R2,R1>           ;RESTORE THE REGISTERS
3840 026772  000240                          NOP     ;TEMP
3841 026774  000207                          RTS     PC                      ;RETURN
3842
3843                                 ;INIT DATA TABLE
3844
3845 026776  111400                  INTTBL:         .WORD   TKINIT
3846 027000  000000                                  .WORD   0
3847 027002  000000                                  .WORD   0
3848 027004  000001                                  .WORD   GO
```

```
3850                                    .SBTTL  GCS RESPONSE HANDLER
3851                                    ;*****************************
3852                                    ;
3853                                    ;GCS RESPONSE HANDLER
3854                                    ;
3855                                    ;Called By       :
3856                                    ;Calls To        :
3857                                    ;Register Inputs :
3858                                    ;                :
3859                                    ;Register Inputs :
3860                                    ;
3861
3862 027006                            GCSHDL::
3863 027006  023761  010734  000020             CMP     CMSTSV,P.CMST(R1)       ;ANY PROGRESS ?
3864 027014  101017                             BHI     5$                      ;YES, CLKEAN UP THE MESS
3865 027016  042737  000040  003674             BIC     #GCSRFL,PCFLAG          ;CLEAR THE GCS MODE FLAG
3866 027024  005037  003572                     CLR     RESP                    ;TAKE OFF THE RESPONSE
3867 027030  005037  003556                     CLR     HNDLRP                  ;TAKE OFF THE RESPONSE
3868 027034  112737  000002  010732             MOVB    #IOHUNG,IOSTAT          ;SET HUNG CONTROLLER BIT
3869 027042  C04737  034022                     JSR     PC,CORDMP
3870 027046  000240                             NOP
3871 027050  000137  027572                     JMP     GCSEXT                  ;GET OUT
3872
3873 027054                            5$:      PUSH    <R1,R2>                 ;
3874 027060  016401  000016                     MOV     COLSAV(R4),R1           ;PUT THE OLD POINTER IN R1
3875 027064  162701  000004                     SUB     #4,R1                   ;ADJUST TO INCLUDE DESCRIPTOR WORDS
3876 027070  016402  000014                     MOV     CNUSAV(R4),R2           ;PUT THE NEW POINTER IN R2
3877 027074  162702  000004                     SUB     #4,R2                   ;ADJUST TO INCLUDE DESCRIPTOR WORDS
3878 027100  022701  003752                     CMP     #CMDBF1,R1              ;OLD POINTER AT BF1 ?
3879 027104  001407                             BEQ     OLD1                    ;YES, GO HANDLE IT
3880 027106  022701  004022                     CMP     #CMDBF2,R1              ;OLD POINTER AT BF2 ?
3881 027112  001434                             BEQ     OLD2                    ;YES, GO HANDLE IT
3882 027114  022701  004072                     CMP     #CMDBF3,R1              ;OLD POINTER AT BF3 ?
3883 027120  001461                             BEQ     OLD3                    ;YES, GO HANDLE IT
3884 027122  000510                             BR      OLD4                    ;NO, GO HANDLE BF4
3885
3886 027124  022702  004072            OLD1:    CMP     #CMDBF3,R2              ;NEW POINTER AT BF3 ?
3887 027130  001004                             BNE     5$                      ;NO, TRY AGAIN
3888 027132  004737  027576                     JSR     PC,EXC1A2               ;GO MOVE COMMAND 1 TO 2
3889 027136  000137  027420                     JMP     ADJUST                  ;GO ADJUST THE OLD POINTER
3890 027142  022702  004142            5$:      CMP     #CMDBF4,R2              ;NEW POINTER AT BF4 ?
3891 027146  001006                             BNE     10$                     ;NO, TRY AGAIN
3892 027150  004737  027620                     JSR     PC,EXC2A3               ;GO MOVE COMMAND 2 TO 3
3893 027154  004737  027576                     JSR     PC,EXC1A2               ;GO MOVE COMMAND 1 TO 2
3894 027160  000137  027420                     JMP     ADJUST                  ;GO ADJUST THE OLD POINTER
3895 027164  004737  027642            10$:     JSR     PC,EXC3A4               ;GO MOVE COMMAND 3 TO 4
3896 027170  004737  027620                     JSR     PC,EXC2A3               ;GO MOVE COMMAND 2 TO 3
3897 027174  004737  027576                     JSR     PC,EXC1A2               ;GO MOVE COMMAND 1 TO 2
3898 027200  000137  027420                     JMP     ADJUST                  ;GO ADJUST THE OLD POINTER
3899
3900 027204  022702  004142            OLD2:    CMP     #CMDBF4,R2              ;NEW POINTER AT BF4 ?
3901 027210  001004                             BNE     5$                      ;NO, TRY AGAIN
3902 027212  004737  027620                     JSR     PC,EXC2A3               ;GO MOVE COMMAND 2 TO 3
3903 027216  000137  027420                     JMP     ADJUST                  ;GO ADJUST THE OLD POINTER
3904 027222  022702  003752            5$:      CMP     #CMDBF1,R2              ;NEW POINTER AT BF1 ?
3905 027226  001006                             BNE     10$                     ;NO, TRY AGAIN
3906 027230  004737  027642                     JSR     PC,EXC3A4               ;GO MOVE COMMAND 3 TO 4
```

```
3907 027234  004737  027620                    JSR     PC,EXC2A3          ;GO MOVE COMMAND 2 TO 3
3908 027240  000137  027420                    JMP     ADJUST             ;GO ADJUST THE OLD POINTER
3909 027244  004737  027664          10$:      JSR     PC,EXC4A1          ;GO MOVE COMMAND 4 TO 1
3910 027250  004737  027642                    JSR     PC,EXC3A4          ;GO MOVE COMMAND 3 TO 4
3911 027254  004737  027620                    JSR     PC,EXC2A3          ;GO MOVE COMMAND 2 TO 3
3912 027260  000137  027420                    JMP     ADJUST             ;GO ADJUST THE OLD POINTER
3913
3914 027264  022702  003752          OLD3:     CMP     #CMDBF1,R2         ;NEW POINTER AT BF1 ?
3915 027270  001004                            BNE     5$                 ;NO, TRY AGAIN
3916 027272  004737  027642                    JSR     PC,EXC3A4          ;GO MOVE COMMAND 3 TO 4
3917 027276  000137  027420                    JMP     ADJUST             ;GO ADJUST THE OLD POINTER
3918 027302  022702  004022          5$:       CMP     #CMDBF2,R2         ;NEW POINTER AT BF2 ?
3919 027306  001006                            BNE     10$                ;NO, TRY AGAIN
3920 027310  004737  027664                    JSR     PC,EXC4A1          ;GO MOVE COMMAND 4 TO 1
3921 027314  004737  027642                    JSR     PC,EXC3A4          ;GO MOVE COMMAND 3 TO 4
3922 027320  000137  027420                    JMP     ADJUST             ;GO ADJUST THE OLD POINTER
3923 027324  004737  027576          10$:      JSR     PC,EXC1A2          ;GO MOVE COMMAND 1 TO 2
3924 027330  004737  027664                    JSR     PC,EXC4A1          ;GO MOVE COMMAND 4 TO 1
3925 027334  004737  027642                    JSR     PC,EXC3A4          ;GO MOVE COMMAND 3 TO 4
3926 027340  000137  027420                    JMP     ADJUST             ;GO ADJUST THE OLD POINTER
3927
3928 027344  022702  004022          OLD4:     CMP     #CMDBF2,R2         ;NEW POINTER AT BF2 ?
3929 027350  001004                            BNE     5$                 ;NO, TRY AGAIN
3930 027352  004737  027664                    JSR     PC,EXC4A1          ;GO MOVE COMMAND 4 TO 1
3931 027356  000137  027420                    JMP     ADJUST             ;GO ADJUST THE OLD POINTER
3932 027362  022702  004072          5$:       CMP     #CMDBF3,R2         ;NEW POINTER AT BF3 ?
3933 027366  001006                            BNE     10$                ;NO, TRY AGAIN
3934 027370  004737  027576                    JSR     PC,EXC1A2          ;GO MOVE COMMAND 1 TO 2
3935 027374  004737  027664                    JSR     PC,EXC4A1          ;GO MOVE COMMAND 4 TO 1
3936 027400  000137  027420                    JMP     ADJUST             ;GO ADJUST THE OLD POINTER
3937 027404  004737  027620          10$:      JSR     PC,EXC2A3          ;GO MOVE COMMAND 2 TO 3
3938 027410  004737  027576                    JSR     PC,EXC1A2          ;GO MOVE COMMAND 1 TO 2
3939 027414  004737  027664                    JSR     PC,EXC4A1          ;GO MOVE COMMAND 4 TO 1
3940
3941 027420                          ADJUST:   POP     <R2,R1>            ;
3942 027424  042737  000040  003674            BIC     #GCSRFL,PCFLAG     ;CLEAR THE GCS MODE FLAG
3943 027432  016137  000020  010734            MOV     P.CMST(R1),CMSTSV  ;PUT THE CMD STATUS INTO THE SAVE LOC
3944 027440  005037  003572                    CLR     RESP               ;TAKE OFF THE RESPONSE
3945 027444  005037  003556                    CLR     HNDLRP             ;TAKE OFF THE RESPONSE
3946 027450  005364  000006                    DEC     CMDSEQ(R4)         ;ADJUST THE CMDSEQ NUMBER BACK 1
3947 027454  042737  100000  010732            BIC     #NURESP,IOSTAT     ;CLEAR THE NEW RESPONSE FLAG IN IOSTAT
3948 027462  105237  010755                    INCB    CRDLIM             ;ADD 1 TO THE CREDIT LIMIT
3949
3950 027466  062764  000050  000016            ADD     #DCBSTP,COLSAV(R4) ;ADJUST THE OLD COMMAND POINTER
3951 027474  022764  004216  000016            CMP     #DCBEND,COLSAV(R4) ;IS IT AT THE END OF THE RING ?
3952 027502  001003                            BNE     5$                 ;NO, BRANCH
3953 027504  012764  003756  000016            MOV     #DCMDBF,COLSAV(R4) ;YES, SET IT TO THE TOP OF THE RING
3954
3955 027512  062705  000050          5$:       ADD     #DCBSTP,R5         ;ADJUST THE OLD COMMAND POINTER
3956 027516  022705  004216                    CMP     #DCBEND,R5         ;IS IT AT THE END OF THE RING ?
3957 027522  001002                            BNE     10$                ;NO, BRANCH
3958 027524  012705  003756                    MOV     #DCMDBF,R5         ;YES, SET IT TO THE TC. OF THE RING
3959
3960 027530  062764  000104  000022  10$:      ADD     #DRBSTP,ROLSAV(R4) ;ADJUST THE OLD RESPONSE POINTER
3961 027536  026464  000212  000022            CMP     URBEND(R4),ROLSAV(R4) ;IS IT AT THE END OF THE BUFFER ?
3962 027544  001003                            BNE     15$                ;NO, KEEP GOING
3963 027546  016464  000210  000022            MOV     URSPBF(R4),ROLSAV(R4) ;YES, SET IT TO BEGINNING OF THE BUFFER
```

```
3964
3965 027554  062701  000104          15$:    ADD     #DRBSTP,R1              ;ADJUST R1
3966 027560  026401  000212                  CMP     URBEND(R4),R1           ;IS IT AT THE END OF THE BUFFER ?
3967 027564  001002                          BNE     GCSEXT                  ;NO, GET OUT
3968 027566  016401  000210                  MOV     URSPBF(R4),R1           ;YES, SET IT TO BEGINNING OF THE BUFFER
3969
3970 027572                          GCSEXT:
3971 027572  000240                          NOP     ;TEMP
3972 027574  000207                          RTS     PC                      ;RETURN
3973
3974
3975 027576  012701  003752          EXC1A2: MOV     #CMDBF1,R1              ;SET R1 TO BF1
3976 027602  012702  004022                  MOV     #CMDBF2,R2              ;SET R2 TO BF2
3977 027606  012122                  5$:     MOV     (R1)+,(R2)+             ;MOV BF1 CONTENTS TO BF2
3978 027610  022701  004022                  CMP     #CMDBF2,R1              ;HAVE WE MOVED THEM ALL
3979 027614  001374                          BNE     5$                      ;NO, KEEP MOVING IT
3980 027616  000207                          RTS     PC                      ;YES, GET OUT
3981
3982 027620  012701  004022          EXC2A3: MOV     #CMDBF2,R1              ;SET R1 TO BF2
3983 027624  012702  004072                  MOV     #CMDBF3,R2              ;SET R2 TO BF3
3984 027630  012122                  5$:     MOV     (R1)+,(R2)+             ;MOV BF2 CONTENTS TO BF3
3985 027632  022701  004072                  CMP     #CMDBF3,R1              ;HAVE WE MOVED THEM ALL
3986 027636  001374                          BNE     5$                      ;NO, KEEP MOVING IT
3987 027640  000207                          RTS     PC                      ;YES, GET OUT
3988
3989 027642  012701  004072          EXC3A4: MOV     #CMDBF3,R1              ;SET R1 TO BF3
3990 027646  012702  004142                  MOV     #CMDBF4,R2              ;SET R2 TO BF4
3991 027652  012122                  5$:     MOV     (R1)+,(R2)+             ;MOV BF3 CONTENTS TO BF4
3992 027654  022701  004142                  CMP     #CMDBF4,R1              ;HAVE WE MOVED THEM ALL
3993 027660  001374                          BNE     5$                      ;NO, KEEP MOVING IT
3994 027662  000207                          RTS     PC                      ;YES, GET OUT
3995
3996 027664  012701  004142          EXC4A1: MOV     #CMDBF4,R1              ;SET R1 TO BF4
3997 027670  012702  003752                  MOV     #CMDBF1,R2              ;SET R2 TO BF1
3998 027674  012122                  5$:     MOV     (R1)+,(R2)+             ;MOV BF4 CONTENTS TO BF1
3999 027676  022702  004022                  CMP     #CMDBF2,R2              ;HAVE WE MOVED THEM ALL
4000 027702  001374                          BNE     5$                      ;NO, KEEP MOVING IT
4001 027704  000207                          RTS     PC                      ;YES, GET OUT
```

```
                                .SBTTL  RESPONSE HANDLER
                                ;******************************
                                ;
                                ;RESPONSE HANDLER
                                ;
                                ;Called By       : CMDSEQ
                                ;Calls To        : ERRDEI, ERRDEL, ERRDEC, CMPDAT, DQCMD
                                ;Register Inputs : R1 - UNIT NUMBER
                                ;                : R4 - LUN BLOCK POINTER
                                ;Register Inputs : R3   POINTER TO CURRENT RESPONSE PACKET
                                ;

027706                          RSPHDL::
027706                                  PUSH    <R3>
027710  005037  003552                  CLR     RESPON                  ;0 HERE TELLS CMDSEQ ALL'S OKAY
027714  105737  010732                  TSTB    IOSTAT                  ;DID WE HAVE I/O TYPE FAILURE?
027720  001404                          BEQ     5$                      ;BRANCH AROUND IF NOT
027722  004737  031724                  JSR     PC,ERRDEI               ;ELSE DECODE AND PRINT IT
027726  000137  030216                  JMP     75$                     ;GET OUT NOW

027732  016403  000022          5$:     MOV     ROLSAV(R4),R3           ;GET OLD RESPONSE BUFFER POINTER
027736  005737  003556                  TST     HNDLRP                  ;DID WE HAVE ANY RESPONSES ?
027742  001002                          BNE     10$                     ;YES, SEE WHAT THEY ARE
027744  000137  030216                  JMP     75$                     ;NO, GET OUT OF HERE

027750  032763  000200  000010  10$:    BIT     #OP.END,P.OPCD(R3)      ;IS IT AN END PACKET?
027756  001003                          BNE     15$                     ;YES, BRANCH
027760  004737  032264                  JSR     PC,ERRDEL               ;GO HANDLE ERROR LOG PACKET
027764  000500                          BR      65$                     ;SEE IF THERE'S MORE RESPONSES

027766  005763  000012          15$:    TST     P.STS(R3)               ;WAS STATUS "NORMAL"?
027772  001445                          BEQ     30$                     ;YES - BRANCH
027774  022763  000022  000012          CMP     #ST.SEX,P.STS(R3)       ;IS IT SERIOUS EXCEPTION STATUS?
030002  001004                          BNE     20$                     ;BRANCH IF NOT
030004  052764  000004  000026          BIS     #NOTALY,LUNFLG(R4)      ;YES, SET THE NO-TALLY FLAG
030012  000463                          BR      60$                     ;GO DE-QUE THE COMMANDS

030014  000240          20$:    NOP     ;TEMP
030016  005737  003560                  TST     MRETRY                  ;ARE WE IN MANUAL RETRY MODE
030022  001426                          BEQ     25$                     ;BRANCH IF NOT
030024  016337  000012  003722          MOV     P.STS(R3),R13           ;GET STATUS
030032  042737  177740  003722          BIC     #177740,R13             ;STRIP UNWANTED BITS
030040  022737  000010  003722          CMP     #ST.DAT,R13             ;IS IT A DATA ERROR
030046  001014                          BNE     25$                     ;BRANCH IF NOT
030050  052764  000006  000026          BIS     #SEREXC!NOTALY,LUNFLG(R4);SERIOUS EXCEPTION AND NO TALLY FLAG
030056  005237  003562                  INC     MANCNT                  ;KEEP TRACK OF NUMBER OF ACTUAL WRITE/READ RETRIES
030062  005337  003560                  DEC     MRETRY                  ;COUNT DOWN MAUAL RETRIES
030066  001035                          BNE     60$                     ;BRANCH IF NOT FINISHED MANUAL RETRIES
030070  012737  000002  003564          MOV     #2,ARETRY               ;SET UP AUTO RETRY COUNTER
030076  000431                          BR      60$                     ;CONTINUE

030100  004737  031122          25$:    JSR     PC,ERRDEC               ;GO HANDLE ERROR STATUS
030104  000405                          BR      50$

030106  000241          30$:    CLC                                     ;CLEAR THE CARRY BIT
030110  006164  000030                  ROL     LEOTFL(R4)              ;ROTATE THE CARRY INTO THE LEOT FLAG
030114  004737  030230                  JSR     PC,RETDON               ;GO SEE IF RETRY COMPLETE
```

```
4060
4061 030120  032761  000001  003526  50$:    BIT     @AVB,DRINUS(R1)          ;HAVE WE DROPPED THE UNIT ?
4062 030126  001433                          BEQ     75$                      ;YES - GET OUT
4063 030130  022763  000241  000010          CMP     @OP.END!OP.RD,P.OPCD(R3) ;DID WE READ THIS TIME ?
4064 030136  001011                          BNE     60$                      ;NO - SKIP DATA COMPARE
4065 030140  022737  000005  002114          CMP     @5,L$TEST                ;ARE WE IN TEST 5 ?
4066 030146  001003                          BNE     55$                      ;NO - DO DATA COMPARE
4067 030150  105737  002236                  TSTB    T5CMP                    ;DO DATA COMPARES IN TEST 5 ?
4068 030154  001402                          BEQ     60$                      ;NO, SKIP DATA COMPARE
4069
4070 030156  004737  032732          55$:    JSR     PC,CMPDAT                ;DO COMPARE DATA
4071
4072 030162  004737  030460          60$:    JSR     PC,DQCMD                 ;DEQUEUE THE COMMAND
4073
4074 030166  062703  000104          65$:    ADD     @DRBSTP,R3               ;ADJUST POINTER TO NEXT PACKET
4075 030172  026403  000212                  CMP     URBEND(R4),R3            ;END OF RESPONSE BUFFER?
4076 030176  001002                          BNE     70$                      ;NO - BRANCH AROUND
4077 030200  016403  000210                  MOV     URSPBF(R4),R3            ;PUT POINTER AT BEGINNING OF BUFFER
4078
4079 030204  005337  003556          70$:    DEC     HNDLRP                   ;DECREMENT RESPONSE COUNTER
4080 030210  001402                          BEQ     75$                      ;ALL DONE, GET OUT
4081 030212  000137  027750                  JMP     10$                      ;GO HANDLE ANOTHER ONE
4082
4083 030216  010364  000022          75$:    MOV     R3,ROLSAV(R4)            ;SAVE OLD RESPONSE BUFFER POINTER
4084 030222                                  POP     <R3>
4085 030224  000240                          NOP     ;TEMP
4086 030226  000207                          RTS     PC
```

```
4088                                    .SBTTL  RETRY DONE
4089                                    ;**********************************
4090                                    ;
4091                                    ; RETRY DONE
4092                                    ;
4093
4094 030230                            RETDON::
4095 030230  005737  003560                    TST     MRETRY                  ;ARE WE IN MANUAL RETRY MODE
4096 030234  001412                            BEQ     40$                     ;BRANCH IF NOT
4097 030236  032737  000001  003560            BIT     #1,MRETRY               ;JUST FINISHED A SPACE REVERSE
4098 030244  001003                            BNE     35$                     ;BRANCH IF NOT
4099 030246  005337  003560                    DEC     MRETRY                  ;COUNT DOWN MANUAL RETRIES (TOGGLE BIT 0)
4100 030252  000501                            BR      100$
4101
4102 030254  005037  003560            35$:    CLR     MRETRY                  ;MANUAL RETRY SUCCESSFUL, CLEAR COUNTER
4103 030260  000414                            BR      50$
4104
4105 030262  005737  003564            40$:    TST     ARETRY                  ;ARE WE IN ATUO RETRY MODE ?
4106 030266  001473                            BEQ     100$                    ;BRANCH IF NOT
4107 030270  032737  000001  003564            BIT     #1,ARETRY               ;JUST FINISHED A SPACE REVERSE
4108 030276  001003                            BNE     45$                     ;BRANCH IF NOT
4109 030300  005337  003564                    DEC     ARETRY                  ;COUNT DOWN AUTO RETRIES (TOGGLE BIT 0)
4110 030304  000464                            BR      100$
4111
4112 030306  005037  003564            45$:    CLR     ARETRY                  ;AUTO RETRY SUCCESSFUL, CLEAR COUNTER
4113
4114 030312                            50$:    PUSH    <R1,R2>                 ;SAVE R1 AND R2
4115 030316  012701  011316                    MOV     #SDATT,R1               ;PUT THE ERROR ADDRESS IN R1
4116 030322  012702  013166                    MOV     #L$ERRTBL,R2            ;PUT THE ERROR TABLE ADDRESS IN R2
4117 030326  012122                            MOV     (R1)+,(R2)+             ;MOVE ERROR TABLE CONTENTS
4118 030330  012122                            MOV     (R1)+,(R2)+             ;MOVE ERROR TABLE CONTENTS
4119 030332  012122                            MOV     (R1)+,(R2)+             ;MOVE ERROR TABLE CONTENTS
4120 030334  012122                            MOV     (R1)+,(R2)+             ;MOVE ERROR TABLE CONTENTS
4121 030336                                    POP     <R2,R1>                 ;RESTORE R2 AND R1
4122 030342  004737  032624                    JSR     PC,ERRTLY               ;TALLY THE ERROR
4123 030346  105037  013167                    CLRB    ERRTYP+1                ;CLEAR THE UPPER BYTE
4124 030352  105737  002230                    TSTB    SOERRP                  ;ARE SOFT ERRORS ENABLED ?
4125 030356  001017                            BNE     60$                     ;YES, GO PRINT THE ERROR
4126 030360  122737  000003  013166            CMPB    #SOFT,L$ERRTBL          ;IS IT A SOFT ERROR ?
4127 030366  001013                            BNE     60$                     ;NO, PRINT IT
4128 030370  022737  000020  003716            CMP     #WR,R11                 ;IS IT A WRITE ?
4129 030376  001411                            BEQ     70$                     ;DON'T PRINT IT
4130 030400  022737  000100  003716            CMP     #WTM,R11                ;IS IT A WRITE TAPE MARK ?
4131 030406  001405                            BEQ     70$                     ;DON'T PRINT IT
4132 030410  105737  002231                    TSTB    RDSOER                  ;ARE WE PRINTING SOFT READ ERRORS ?
4133 030414  001402                            BEQ     70$                     ;NO, DON'T PRINT IT
4134 030416  004737  032714            60$:    JSR     PC,PRIERR               ;PRINT THE ERROR
4135 030422  005037  003562            70$:    CLR     MANCNT                  ;CLEAR THE MANUAL RETRY COUNT
4136 030426  042764  000200  000026            BIC     #RETFLG,LUNFLG(R4)      ;AND THE FLAG
4137 030434  042737  000100  003674            BIC     #CMDONE,PCFLAG          ;CLEAR THE COMMAND DONE FLAG
4138 030442  013737  003542  003536            MOV     CCTSAV,CMDCNT           ;RESTORE COMMAND COUNT
4139 030450  063737  003542  003540            ADD     CCTSAV,RSPCNT           ;ADJUST THE RESPONSE COUNT
4140
4141 030456  000207                    100$:   RTS     PC
4142
```

```
4144                                    .SBTTL  DE-QUEUE COMMAND
4145                                    ;**********************************
4146                                    ;
4147                                    ; DE-QUEJE COMMAND
4148                                    ;
4149                                    ;Called By       : RSPHDL
4150                                    ;Calls To        : LGSTAT
4151                                    ;Register Inputs : R2 - OLD POINTER TO PROGRAM COMMAND RING
4152                                    ;Register Outputs: R2 - UPDATED
4153                                    ;
4154
4155 030460                            DQCMD::
4156 030460  004737  030520                    JSR     PC,LGSTAT               ;CALL LOG STATS
4157 030464  062702  000014                    ADD     #PCBSTP,R2              ;ADJUST THE OLD COMMAND POINTER
4158 030470  022702  003522                    CMP     #PCBEND,R2              ;ARE WE AT THE END OF THE BUFFER ?
4159 030474  001002                           BNE     5$                      ;NO, KEEP GOING
4160 030476  012702  003442                    MOV     #PCMDBF,R2              ;YES, SET IT BACK TO THE TOP
4161
4162 030502  005337  003540            5$:     DEC     RSPCNT                  ;DECREMENT THE RESPONSE COUNTER
4163 030506  012737  177777  010734            MOV     #-1,CMSTSV              ;RESET THE GCS PROGRESS COUNT
4164 030514  000240                           NOP     ;TEMP
4165 030516  000207                           RTS     PC                      ;RETURN
4166
```

```
                                        .SBTTL  LOG STATISTICS
                                        ;**********************************
                                        ;
                                        ; LOG STATISTICS
                                        ;
                                        ;Called By       : DQCMD
                                        ;Register Inputs : R2 - OLD PROGRAM COMMAND POINTER
                                        ;                  R4 - LUN BLOCK POINTER
                                        ;
                                        ;
030520                                  LGSTAT::
030520                                          PUSH    <R4>                    ;SAVE R4
030522  032764  000004  000026                  BIT     #NOTALY,LUNFLG(R4)      ;IS THIS NOT TO BE TALLIED ?
030530  001166                                  BNE     TLYEXT                  ;YES, GET OUT
030532  122762  000040  000000                  CMPB    #ACC,CMD(R2)            ;SEE IF COMMAND A READ OR WRITE
030540  103562                                  BLO     TLYEXT                  ;NO, EXIT SUBROUTINE
030542  105762  000000                          TSTB    CMD(R2)                 ;IS IT A NULL ?
030546  001557                                  BEQ     TLYEXT                  ;YES, EXIT SUBROUTINE
030550  062704  000134                          ADD     #GWRBY1,R4              ;ADD OFFSET TO BYTE COUNT STORAGE
030554  022737  000002  003732                  CMP     #TF.PE,FORMAT           ;ARE WE IN PE OR GCR ?
030562  001002                                  BNE     TALLY                   ;GO TALLY THE GCR BYTE COUNT
030564  062704  000020                          ADD     #20,R4                  ;ADJUST R4 FOR PE
030570  122762  000020  000000  TALLY:          CMPB    #WR,CMD(R2)             ;IS IT A WRITE ?
030576  001052                                  BNE     5$                      ;NO, HANDLE READ
030600  066214  000002                          ADD     ITMOFF(R2),(R4)         ;YES, ADD THE BYTES WRITTEN TO TOTAL
030604  021427  001747          1$:             CMP     (R4),#999.              ;IS IT HIGER THAN 999. ?
030610  003405                                  BLE     2$                      ;BRANCH IF IT'S NOT
030612  162714  001750                          SUB     #1000.,(R4)             ;SUBTRACT 1000. FROM THE LOWER ORDER WORD
030616  005264  000002                          INC     2(R4)                   ;INCREMENT THE SECOND WORD
030622  000770                                  BR      1$
030624  026427  000002  001747  2$:             CMP     2(R4),#999.             ;IS IT HIGER THAN 999. ?
030632  003406                                  BLE     3$                      ;BRANCH IF IT'S NOT
030634  162764  001750  000002                  SUB     #1000.,2(R4)            ;SUBTRACT 1000. FROM THE LOWER ORDER WORD
030642  005264  000004                          INC     4(R4)                   ;INCREMENT THE THIRD WORD
030646  000766                                  BR      2$
030650  026427  000004  001747  3$:             CMP     4(R4),#999.             ;IS IT HIGER THAN 999. ?
030656  003406                                  BLE     4$                      ;BRANCH IF IT'S NOT
030660  162764  001750  000004                  SUB     #1000.,4(R4)            ;SUBTRACT 1000. FROM THE LOWER ORDER WORD
030666  005264  000006                          INC     6(R4)                   ;INCREMENT THE FOURTH WORD
030672  000766                                  BR      3$
030674  026427  000006  001747  4$:             CMP     6(R4),#999.             ;IS IT HIGER THAN 999. ?
030702  003501                                  BLE     TLYEXT                  ;BRANCH IF IT'S NOT
030704  005014                                  CLR     (R4)                    ;CLEAR
030706  005064  000002                          CLR     2(R4)                   ;       WRITE
030712  005064  000004                          CLR     4(R4)                   ;             BYTE
030716  005064  000006                          CLR     6(R4)                   ;                  COUNTS
030722  000471                                  BR      TLYEXT                  ;EXIT
                                        ;
030724  022763  000016  000012  5$:             CMP     #ST.TM,P.STS(R3)        ;WAS THIS A TAPE MARK DURING READ
030732  001465                                  BEQ     TLYEXT                  ;YES, GET OUT
030734  022763  000010  000012                  CMP     #10,P.STS(R3)           ;WAS THIS A DATA ERROR DURING READ
030742  001461                                  BEQ     TLYEXT                  ;YES, GET OUT
030744  022763  000350  000012                  CMP     #350,P.STS(R3)          ;WAS THIS A DATA ERROR DURING READ
030752  001455                                  BEQ     TLYEXT                  ;YES, GET OUT
030754  066364  000040  000010                  ADD     P.TRBC(R3),10(R4)       ;YES, ADD THE BYTES READ TO TOTAL
030762  026427  000010  001747  6$:             CMP     10(R4),#999.            ;IS IT HIGER THAN 999. ?
030770  003406                                  BLE     7$                      ;BRANCH IF IT'S NOT
```

```
4225 030772  162764  001750  000010          SUB     #1000.,10(R4)          ;SUBTRACT 1000. FROM THE LOWER ORDER WORD
4226 031000  005264  000012                  INC     12(R4)                 ;INCREMENT THE SECOND WORD
4227 031004  000766                          BR      6$
4228 031006  026427  000012  001747  7$:     CMP     12(R4),#999.           ;IS IT HIGER THAN 999. ?
4229 031014  003406                          BLE     8$                     ;BRANCH IF IT'S NOT
4230 031016  162764  001750  000012          SUB     #1000.,12(R4)          ;SUBTRACT 1000. FROM THE LOWER ORDER WORD
4231 031024  005264  000014                  INC     14(R4)                 ;INCREMENT THE SECOND WORD
4232 031030  000766                          BR      7$
4233 031032  026427  000014  001747  8$:     CMP     14(R4),#999.           ;IS IT HIGER THAN 999. ?
4234 031040  003406                          BLE     9$                     ;BRANCH IF IT'S NOT
4235 031042  162764  001750  000014          SUB     #1000.,14(R4)          ;SUBTRACT 1000. FROM THE LOWER ORDER WORD
4236 031050  005264  000016                  INC     16(R4)                 ;INCREMENT THE SECOND WORD
4237 031054  000766                          BR      8$
4238 031056  026427  000016  001747  9$:     CMP     16(R4),#999.           ;IS IT HIGER THAN 999. ?
4239 031064  003410                          BLE     TLYEXT                 ;BRANCH IF IT'S NOT
4240 031066  005064  000010                  CLR     10(R4)                 ;CLEAR
4241 031072  005064  000012                  CLR     12(R4)                 ;      READ
4242 031076  005064  000014                  CLR     14(R4)                 ;           BYTE
4243 031102  005064  000016                  CLR     16(R4)                 ;                COUNTS
4244 031106                          TLYEXT: POP     <R4>                   ;RESTORE R4
4245 031110  042764  000004  000026          BIC     #NCTALY,LUNFLG(R4)     ;CLEAR THE NO-TALLY FLAG BEFORE EXITING
4246 031116  000240                          NOP     ;TEMP
4247 031120  000207                          RTS     PC                     ;RETURN
```

```
4249                                    .SBTTL ERROR DECODE
4250                                    ;**********************************
4251                                    ;
4252                                    ; ERROR DECODE
4253                                    ;
4254                                    ;Called By        : RSPHDL
4255                                    ;Calls To         : ERRTLY, PRIERR
4256                                    ;Register Inputs : R2 - OLD PROGRAM COMMAND BUFFER POINTER
4257                                    ;
4258
4259 031122                            ERRDEC::
4260 031122                                    PUSH    <R5>                    ;SAVE R5
4261 031124  016205  000010                    MOV     XFERST(R2),R5           ;PUT THE COMMAND STATUS IN R5
4262 031130  022705  000400                    CMP     #ST.ONL,R5              ;IS IT A UNIT ONLINE ERROR ?
4263 031134  001005                            BNE     5$                      ;BRANCH IF IT ISN'T
4264 031136  012737  000100  003620            MOV     #ONLB,WRKMSK            ;SET THE ERROR BIT IN THE MASK
4265 031144  000137  031712                    JMP     MSKTST                  ;GO TEST IF IT'S O.K.
4266 031150  042705  177740            5$:     BIC     #177740,R5              ;CLEAR THE UNWANTED BITS
4267 031154  022705  000010                    CMP     #ST.DAT,R5              ;IS IT A DATA ERROR (RETRY)
4268 031160  001017                            BNE     10$                     ;BRANCH IF NOT
4269 031162  105737  002225                    TSTB    SERREC                  ;USER DISABLE RETRIES ?
4270 031166  001414                            BEQ     10$                     ;BRANCH IF SO
4271 031170  005737  003564                    TST     ARETRY                  ;DID WE FAIL IN AUTO RETRY MODE ?
4272 031174  001011                            BNE     10$                     ;BRANCH IF SO
4273 031176  012737  000016  003560            MOV     #14.,MRETRY             ;SET FOR 7 RETRIES, 7 SPACE RECORDS
4274 031204  004737  032500                    JSR     PC,RUNJAM               ;GO ADJUST THE COUNTERS
4275 031210  005237  003562                    INC     MANCNT                  ;INCREMENT FOR FIRST RETRY
4276 031214  000137  032142                    JMP     EDCEXT                  ;RETURN
4277
4278 031220  022762  002000  000010    10$:    CMP     #2000,XFERST(R2)        ;IS EOT SET IN TRANSFER STATUS ?
4279 031226  001055                            BNE     20$                     ;BRANCH IF IT ISN'T
4280 031230  004737  030230                    JSR     PC,RETDON               ;SEE IF WE'RE DOING RETRIES
4281 031234  122762  000020  000000            CMPB    #WR,CMD(R2)             ;IS IT A WRITE ?
4282 031242  001410                            BEQ     11$                     ;YES, SET UP FOR EOT
4283 031244  022762  000100  000000            CMP     #WTM,CMD(R2)            ;IS IT A WRITE TAPE MARK ?
4284 031252  001404                            BEQ     11$                     ;YES, SET UP FOR EOT
4285 031254  004737  030230                    JSR     PC,RETDON               ;SEE IF WE'RE DOING RETRIES
4286 031260  000137  032142                    JMP     EDCEXT                  ;RETURN
4287
4288 031264  052761  000004  003526    11$:    BIS     #EOT,DRINUS(R1)         ;SET THE DRIVE TO EOT
4289 031272  005237  003706                    INC     UEOT                    ;INC THE EOT FLAG
4290 031276  163737  003536  003540            SUB     CMDCNT,RSPCNT           ;SET RESPONSE COUNT TO NUMBER OUT
4291 031304  005037  003536                    CLR     CMDCNT                  ;ISSUE NO MORE COMMANDS
4292 031310  032764  000010  000026            BIT     #EOTPR,LUNFLG(R4)       ;HAS EOT BEEN PRINTED FOR THIS DRIVE ?
4293 031316  001017                            BNE     15$                     ;DON'T PRINT IT AGAIN
4294 031320                                    PUSH    <R1>                    ;SAVE R1
4295 031322  006001                            ROR     R1                      ;DIVIDE R1 BY 2
4296 031324                                    PRINTF  #UNTEOT,R1              ;PRINT UNIT AT EOT MESSAGE
     031324  010146                            MOV     R1,-(SP)
     031326  012746  020253                    MOV     #UNTEOT,-(SP)
     031332  012746  000002                    MOV     #2,-(SP)
     031336  010600                            MOV     SP,R0
     031340  104417                            TRAP    C$PNTF
     031342  062706  000006                    ADD     #6,SP
4297 031346                                    POP     <R1>                    ;RESTORE R1
4298 031350  052764  000010  000026            BIS     #EOTPR,LUNFLG(R4)       ;EOT BEEN PRINTED FOR THIS DRIVE
4299 031356  000137  032142            15$:    JMP     EDCEXT                  ;GET OUT
```

```
4300
4301 031362  012737  011036  003710  20$:    MOV     @CMDT,R8            ;PUT THE ERROR TABLE ADDRESS IN R8
4302 031370  022705  000013                  CMP     #13,R5              ;IS IT A DRIVE ERROR ?
4303 031374  001003                          BNE     25$                 ;NO, CONTINUE
4304 031376  052737  000010  003674          BIS     @DRERFL,PCFLAG      ;SET THE DRIVE ERROR FLAG
4305 031404  022705  000023          25$:    CMP     #23,R5              ;IS IT A VALID STATUS ?
4306 031410  103003                          BHIS    30$                 ;IT'S VALID, BRANCH
4307 031412  012705  000024                  MOV     #24,R5              ;MAKE SURE ITS NOT MORE THAN 24
4308 031416  000552                          BR      ERREXT              ;TAKE THE ERROR EXIT
4309
4310 031420  012737  000002  003552  30$:    MOV     @SEREXC,RESPON      ;SET SERIOUS EXCEPTION
4311 031426  016337  000000  003724          MOV     P.CRF(R3),SECRNS     SAVE THE CURRENT COMMAND REF #
4312 031434  022705  000006                  CMP     @ST.WPR,R5          ;IS IT A WRITE PROTECT ERROR ?
4313 031440  001004                          BNE     35$                 ;BRANCH IF IT ISN'T
4314 031442  012737  000020  003620          MOV     @WPRB,WRKMSK        ;SET THE ERROR BIT IN THE MASK
4315 031450  000520                          BR      MSKTST              ;GO TEST IF IT'S O.K.
4316
4317 031452  022705  000016          35$:    CMP     @ST.TM,R5           ;IS IT A TAPE MARK ERROR ?
4318 031456  001070                          BNE     50$                 ;BRANCH IF IT ISN'T
4319 031460  C52764  000002  000026          BIS     @SEREXC,LUNFLG(R4)  ;SET SERIOUS EXCEPTION
4320 031466  000261                          SEC                         ;
4321 031470  006164  000030                  ROL     LEOTFL(R4)          ;
4322 031474  042764  177774  000030          BIC     #177774,LEOTFL(R4)  ;
4323 031502  022764  000003  000030          CMP     #3,LEOTFL(R4)       .
4324 031510  001041                          BNE     40$                 ;
4325 031512  032764  000010  000026          BIT     @EOTPR,LUNFLG(R4)   ;HAS LEOT BEEN PRINTED FOR THIS DRIVE ?
4326 031520  001033                          BNE     36$                 ;DON'T PRINT IT AGAIN
4327 031522  052764  000010  000026          BIS     @EOTPR,LUNFLG(R4)   ;LEOT BEEN PRINTED FOR THIS DRIVE
4328 031530  052761  000004  003526          BIS     @EOT,DRINUS(R1)     ;
4329 031536  005237  003706                  INC     UEOT                ;INC THE EOT FLAG
4330 031542                                  PUSH    <R1>                ;SAVE R1
4331 031544  006001                          ROR     R1                  ;DIVIDE R1 BY 2
4332 031546                                  PRINTF  @UNTLOT,R1          ;PRINT UNIT AT EOT MESSAGE
     031546  010146                          MOV     R1,-(SP)
     031550  012746  020307                  MOV     @UNTLOT, (SP)
     031554  012746  000002                  MOV     #2,-(SP)
     031560  010600                          MOV     SP,R0
     031562  104417                          TRAP    C$PNTF
     031564  062706  000006                  ADD     #6,SP
4333 031570                                  POP     <R1>                ;RESTORE R1
4334 031572  163737  003536  003540          SUB     CMDCNT,RSPCNT       ;SET RESPONSE COUNT TO NUMBER OUT
4335 031600  005037  003536                  CLR     CMDCNT              ;ISSUE NO MORE COMMANDS
4336 031604  005037  003552                  CLR     RESPON              ;MAKE SURE WE
4337 031610  000137  032142          36$:    JMP     EDCEXT              ;GET OUT
4338
4339 031614  132763  000010  000011  40$:    BITB    @EF.EOT,P.FLGS(R3)  ;IS THE TAPE MARK AT EOT ?
4340 031622  001402                          BEQ     45$                 ;NO, KEEP ON GOING
4341 031624  000137  032142                  JMP     EDCEXT              ;YES, GET OUT
4342 031630  012737  000010  003620  45$:    MOV     @TMB,WRKMSK         ;SET THE ERROR BIT IN THE MASK
4343 031636  000425                          BR      MSKTST              ;GO TEST IF IT'S O.K.
4344
4345 031640  022705  000020          50$:    CMP     @ST.RDT,R5          ;IS IT A RECORD DATA TRUNCATED ERROR ?
4346 031644  001007                          BNE     55$                 ;BRANCH IF IT ISN'T
4347 031646  000241                          CLC                         ;CLEAR THE CARRY BIT
4348 031650  006164  000030                  ROL     LEOTFL(R4)          ;ROTATE THE CARRY INTO THE LEOT FLAG
4349 031654  012737  000002  003620          MOV     @RDTB,WRKMSK        ;SET THE ERROR BIT IN THE MASK
4350 031662  000413                          BR      MSKTST              ;GO TEST IF IT'S O.K.
```

```
4351
4352 031664  022705  000023          55$:    CMP     @ST.LED,R5              ;IS IT A LOGICAL END OF TAPE ERROR ?
4353 031670  001002                  BNE     60$                             ;BRANCH IF IT ISN'T
4354 031672  000137  032142          JMP     EDCEXT                          ;GET OUT IF LEOT DETECTED
4355
4356 031676  022705  000004          60$:    CMP     @ST.AVL,R5              ;IS IT A UNIT AVAILABLE ERROR ?
4357 031702  001020                  BNE     ERREXT                          ;BRANCH IF IT ISN'T
4358 031704  012737  000040  003620          MOV     @AVLB,WRKMSK            ;SET THE ERROR BIT IN THE MASK
4359
4360 031712  033737  003620  003616  MSKTST: BIT     WRKMSK,TSTMSK           ;IS IT AN ACCEPTABLE ERROR ?
4361 031720  001110                  BNE     EDCEXT                          ;GET OUT IF IT IS
4362 031722  000410                  BR      ERREXT                          ;OTHERWISE PRINT THE ERROR
4363
4364 031724                  ERRDEI::
4365 031724                          PUSH    <R5>                            ;SAVE R5
4366 031726  113705  010732          MOVB    IOSTAT,R5                       ;PUT THE I/O ERROR CODE INTO R5
4367 031732  012737  010756  003710  MOV     @IOERTB,R8                      ;SET THE ERROR TABLE ADDRESS IN R8
4368 031740  042705  177770          BIC     @177770,R5                      ;CLEAR OFF ALL UNWANTED BITS
4369
4370 031744  005305          ERREXT: DEC     R5                              ;SUBTRACT 1 FROM R5
4371 031746  006305                  ASL     R5                              ;MULTIPLY R5 BY 10(8)
4372 031750  006305                  ASL     R5                              ;
4373 031752  006305                  ASL     R5                              ;
4374 031754  063705  003710          ADD     R8,R5                           ;ADD THE TABLE ADDRESS TO R5
4375 031760                          PUSH    <R3>
4376 031762  012703  013166          MOV     @L$ERRTBL,R3                    ;SET R3 TO THE GENERIC ERROR TABLE
4377 031766  012523                  MOV     (R5)+,(R3)+                     ;MOVE ERROR TABLE CONTENTS
4378 031770  012523                  MOV     (R5)+,(R3)+                     ;MOVE ERROR TABLE CONTENTS
4379 031772  012523                  MOV     (R5)+,(R3)+                     ;MOVE ERROR TABLE CONTENTS
4380 031774  011513                  MOV     (R5),(R3)                       ;MOVE ERROR TABLE CONTENTS
4381 031776                          POP     <R3>
4382
4383 032000  022762  000010  000010          CMP     @EV.LGP,XFERST(R2)      ;IS IS A LONG GAP ENCOUNTERED ?
4384 032006  001006                  BNE     ERTLY                           ;NO, KEEP GOING
4385 032010  112737  000001  013166          MOVB    @DEVFAT,ERRTYP          ;YES, DROP THE UNIT
4386 032016  004737  034022                  JSR     PC,CORDMP               ;;;;GO DO IT
4387 032022  000240                          NOP                             ;;;;
4388 032024  004737  032624          ERTLY:  JSR     PC,ERRTLY               ;TALLY THE ERROR
4389 032030  105037  013167                  CLRB    ERRTYP+1                ;CLEAR UPPER BYTE
4390 032034  105737  002230                  TSTB    SOERRP                  ;ARE SOFT ERRORS ENABLED ?
4391 032040  001017                  BNE     6$                              ;YES, GO PRINT THE ERROR
4392 032042  122737  000003  013166          CMPB    @SOFT,L$ERRTBL          ;IS IT A SOFT ERROR ?
4393 032050  001013                  BNE     6$                              ;NO, PRINT IT
4394 032052  022737  000020  003716          CMP     @WR,R11                 ;IS IT A WRITE ?
4395 032060  001411                  BEQ     8$                              ;DON'T PRINT IT
4396 032062  022737  000100  003716          CMP     @WTM,R11                ;IS IT A WRITE TAPE MARK ?
4397 032070  001405                  BEQ     8$                              ;DON'T PRINT IT
4398 032072  105737  002231                  TSTB    RDSOER                  ;ARE WE PRINTING SOFT READ ERRORS ?
4399 032076  001402                  BEQ     8$                              ;NO, DON'T PRINT IT
4400 032100  004737  032714          6$:     JSR     PC,PRIERR               ;GO PRINT THE ERROR
4401
4402 032104  132737  000001  002233  8$:     BITB    @BIT0,DMPFLG            ;SHOULD WE DUMP PROGRAM TABLES?
4403 032112  001403                  BEQ     10$                             ;NO - BRANCH
4404 032114  004737  034022                  JSR     PC,CORDMP               ;GO DO IT
4405 032120  000240                          NOP
4406 032122  022737  000001  013166  10$:    CMP     @DEVFAT,ERRTYP          ;IS IT A FATAL ERROR ?
4407 032130  001004                  BNE     EDCEXT                          ;NO EXIT
```

```
4408 032132  010100                        MOV     R1,R0                   ;MOVE UNIT # * 2 TO R0
4409 032134  006000                        ROR     R0                      ;DIVIDE BY 2
4410 032136  004737  040012                JSR     PC,DROPUN               ;DROP DRIVE FROM TESTING
4411 032142                        EDCEXT: POP     <R5>                    ;RESTORE REGISTERS
4412 032144  000240                        NOP     ;TEMP
4413 032146  000207                        RTS     PC                      ;RETURN
4414
4415 032150     045     116     045  RET1::  .ASCIZ  /%N%A***** BEFORE  %03%S2%04%A *****%N/
4416 032216     045     116     045  RET2::  .ASCIZ  /%N%A***** AFTER   %03%S2%04%A *****%N/
4417                                .EVEN
```

```
                                .SBTTL ERROR LOG DECODE
                                ;**********************************
                                ;
                                ; ERROR LOG DECODE
                                ;
                                ;Called By       :
                                ;Calls To        :
                                ;Inputs          :
                                ;Outputs         :
                                ;Register Inputs :
                                ;Register Outputs:

032264                          ERRDEL::
032264                                  PUSH    <R3,R5>
032270  116237  000000  003716          MOVB    CMD(R2),R11             ;GET THE COMMAND PRIMITIVE FOR LATER USE
032276  042737  177407  003716          BIC     #177407,R11             ;GET JUST THE ROOT PRIMATIVE
032304  122763  000005  000010          CMPB    #FM.TPE,L.FMT(R3)       ;TAPE TRANSFER ERROR LOG?
032312  001003                          BNE     1$                      ;NO, DECODE IT
032314  012705  011346                  MOV     #TPEERL,R5              ;PRINT PACKET
032320  000420                          BR      PRTEXT                  ;GET OUT
032322  122763  000000  000010  1$:     CMPB    #FM.CNT,L.FMT(R3)       ;CONTROLLER ERROR LOG?
032330  001374                          BNE     1$                      ;NO, SEE WHAT IT IS
032332  012705  011326                  MOV     #CNTERL,R5              ;PRINT PACKET
032336  000411                          BR      PRTEXT                  ;GET OUT
032340  122763  000001  000010  5$:     CMPB    #FM.BAD,L.FMT(R3)       ;HOST MEMORY ACCESS ERROR LOG?
032346  001003                          BNE     10$                     ;NO, GET OUT
032350  012705  011336                  MOV     #BADERL,R5              ;SET UP TO PRINT HOST MEM ACC ERL
032354  000402                          BR      PRTEXT                  ;GET OUT
032356  012705  011356          10$:    MOV     #UNKERL,R5              ;SET UP TO UNKNOWN FORMAT ERROR LOG
032362                          PRTEXT: PUSH    <R1>                    ;SAVE R1
032364  116237  000000  003716          MOVB    CMD(R2),R11             ;GET THE COMMAND PRIMITIVE FOR LATER USE
032372  012701  013166                  MOV     #L$ERRTBL,R1            ;R1 = SUPERVISORS ERROR TABLE
032376  012521                          MOV     (R5)+,(R1)+             ;COPY PROGRAM'S ERROR TABLE
032400  012521                          MOV     (R5)+,(R1)+             ;  TO SUPERVISOR'S
032402  012521                          MOV     (R5)+,(R1)+             ;     ERROR
032404  011511                          MOV     (R5),(R1)               ;       TABLE
032406                                  POP     <R1>                    ;RESTORE R1
032410  004737  032624                  JSR     PC,ERRTLY               ;TALLY ERROR FIRST
032414  105037  013167                  CLRB    ERRTYP+1                ;DISCARD INFO USED BY ERRTLLY
032420  105737  002230                  TSTB    SOERRP                  ;ARE SOFT ERRORS ENABLED ?
032424  001017                          BNE     1$                      ;YES, GO PRINT THE ERROR
032426  122737  000003  013166          CMPB    #SOFT,L$ERRTBL          ;IS IT A SOFT ERROR ?
032434  001013                          BNE     1$                      ;NO, PRINT IT
032436  022737  000020  003716          CMP     #WR,R11                 ;IS IT A WRITE ?
032444  001411                          BEQ     EDLEXT                  ;DON'T PRINT IT
032446  022737  000100  003716          CMP     #WTM,R11                ;IS IT A WRITE TAPE MARK ?
032454  001405                          BEQ     EDLEXT                  ;DON'T PRINT IT
032456  105737  002231                  TSTB    RDSOER                  ;ARE WE PRINTING SOFT READ ERRORS ?
032462  001402                          BEQ     EDLEXT                  ;NO, DON'T PRINT IT
032464  004737  032714          1$:     JSR     PC,PRIERR               ;GO PRINT IT
032470                          EDLEXT: POP     <R5,R3>
032474  000240                          NOP     ;TEMP
032476  000207                          RTS     PC
```

```
                                .SBTTL  RETRY UNJAM
                                ;**********************************
                                ;
                                ;RETRY UNJAM
                                ;
                                ;Called By       :
                                ;Calls To        :
                                ;Inputs          :
                                ;Outputs         :
                                ;Register Inputs :
                                ;Register Outputs:
                                ;
032500                          RUNJAM::
032500  016337  000000  003724          MOV     P.CRF(R3),SECRNS        ;SAVE THE CURRENT COMMAND REF #
032506  013737  003536  003542          MOV     CMDCNT,CCTSAV           ;SAVE THE COMMAND COUNT
032514  163737  003536  003540          SUB     CMDCNT,RSPCNT           ;SET RESPONCE COUNT TO NUMBER OUT
032522  005037  003536                  CLR     CMDCNT                  ;ISSUE NO MORE COMMANDS
032526  016437  000006  003614          MOV     CMDSEQ(R4),SAVDIF       ;SET UP TO UNJAM THE QUEUES
032534  163737  003724  003614          SUB     SECRNS,SAVDIF           ;SUBTRACT CURRENT FROM THE HIGHEST
032542  063737  003614  003542          ADD     SAVDIF,CCTSAV           ;ADJUST THE COMMAND COUNT SAVE

032550  163764  003614  000034          SUB     SAVDIF,OBJFDL(R4)       ;ADUST THE OBJECT COUNT
032556  103002                          BCC     5$                      ;GET OUT IF NO CARRY
032560  005364  000036                  DEC     OBJFDH(R4)              ;OTHERWISE, ADJUST THE HIGH WORD

032564  022737  000004  003602  5$:     CMP     #N,SUBCNT
032572  001002                          BNE     10$
032574  005037  003602                  CLR     SUBCNT
032600  005237  003602          10$:    INC     SUBCNT
032604  005337  003614                  DEC     SAVDIF
032610  001365                          BNE     5$

032612  052764  000006  000026          BIS     #SEREXC!NOTALY,LUNFLG(R4);SERIOUS EXCEPTION AND NO TALLY FLAG
032620  000240                          NOP     ;TEMP
032622  000207                          RTS     PC                      ;RETURN
```

```
4510                                    .SBTTL  ERROR TALLY
4511                                    ;**********************************
4512                                    ;
4513                                    ; ERROR TALLY
4514                                    ;
4515                                    ;Called By       : ERRDEC, ERRDEI, ERRDEL
4516                                    ;
4517
4518 032624                            ERRTLY::
4519 032624                                    PUSH    <R1,R2>                 ;SAVE R1 AND R2
4520 032630  113701  013167                    MOVB    ERRTYP+1,R1             ;GET THE ERROR TYPE IN R1
4521 032634  022737  000002  003732            CMP     @TF.PE,FORMAT           ;ARE WE IN PE MODE ?
4522 032642  001002                            BNE     5$                      ;NO, GO TALLY ERROR
4523 032644  062701  000036                    ADD     @36,R1                  ;YES ADJUST R1 FOR PE TALLIES
4524 032650  060401                    5$:     ADD     R4,R1                   ;ADD THE OFFSET TO THE LUN POINTER
4525 032652  116202  000000                    MOVB    CMD(R2),R2              ;GET THE COMMAND PRIMATIVE
4526 032656  042702  000007                    BIC     @7,R2                   ;CLEAR OFF THE MODIFIERS
4527 032662  022702  000040                    CMP     @ACC,R2                 ;IS IT A UNIT ACCESS TYPE COMMAND ?
4528 032666  103405                            BLO     10$                     ;GO DO UNIT ACCESS ERROR
4529 032670  022702  000020                    CMP     @WR,R2                  ;IS IT A WRTIE COMMAND ?
4530 032674  001402                            BEQ     10$                     ;YES,GO TALLY ERROR
4531 032676  062701  000002                    ADD     @2,R1                   ;NO, ADD READ OFFSET TO ERRTYP
4532 032702  005211                    10$:    INC     (R1)                    ;INC THE ERROR COUNT
4533 032704                                    POP     <R2,R1>                 ;RESTORE R1 AND R2
4534 032710  000240                            NOP     ;TEMP
4535 032712  000207                            RTS     PC                      ;EXIT
4536
```

```
4538                                    .SBTTL  PRINT ERROR
4539                                    ;**********************************
4540                                    ;
4541                                    ; PRINT ERROR
4542                                    ;
4543                                    ;Called By       : ERRDEC, ERRDEI, ERRDEL
4544                                    ;Calls To        : ERROR
4545                                    ;Register Inputs :
4546                                    ;Register Outputs:
4547
4548 032714                            PRIERR::
4549 032714  010337  003740                    MOV     R3,R3SAVE               ;SAVE R3
4550 032720  010437  003742                    MOV     R4,R4SAVE               ;SAVE R4
4551 032724                                    ERROR                           ;ERROR MACRO
     032724  104460                            TRAP    C$ERROR
4552 032726  000240                            NOP     ;TEMP
4553 032730  000207                            RTS     PC                      ;RETURN
4554
```

```
                                .SBTTL  COMPARE DATA
                                ;**********************************
                                ;
                                ; COMPARE DATA
                                ;
                                ;Called By       :
                                ;Calls To        :
                                ;Inputs          :
                                ;Outputs         :
                                ;Register Inputs :
                                ;Register Outputs:
                                ;
032732                          CMPDAT::
032732                                  PUSH    <R1,R2,R3,R5>           ;SAVE R1,R2,R3,R5
032732  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
032734  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
032736  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
032740  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
032742  026363  000040  000014          CMP     P.TRBC(R3),P.BCNT(R3)   ;AS MANY BYTES READ AS WRITTEN ?
032750  001424                          BEQ     5$                      ;BRANCH IF YES
032752  012705  011306                  MOV     #RLST,R5                ;PUT THE RLS TABLE ADDRESS IN R5
032756  012702  013166                  MOV     #L$ERRTBL,R2            ;PUT THE ERROR TABLE ADDRES IN R2
032762  012522                          MOV     (R5)+,(R2)+             ;MOV THE RLS TABLE TO THE ERROR TABLE
032764  012522                          MOV     (R5)+,(R2)+             ;
032766  012522                          MOV     (R5)+,(R2)+             ;
032770  011512                          MOV     (R5),(R2)               ;
032772                                  POP     <R5,R3,R2,R1>           ;RESTORE REGISTERS
032772  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
032774  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
032776  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
033000  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
033002  004737  032624                  JSR     PC,ERRTLY               ;GO TALLY THE ERROR
033006  105037  013167                  CLRB    ERRTYP+1                ;CLEAR THE LUN POINTER
033012  004737  032714                  JSR     PC,PRIERR               ;GO PRINT THE ERROR
033016  000137  033414                  JMP     45$                     ;GET OUT IF THERE WAS AN ERROR
033022  005037  003622          5$:     CLR     CMPERR                  ;CLEAR LOCATION CMPERR
033026  042764  000020  000026          BIC     #ODDFLG,LUNFLG(R4)      ;CLEAR THE ODD BYTE COUNT FLAG
033034  016337  000014  003710          MOV     P.BCNT(R3),R8           ;PUT THE TAPE RECORD BYTE COUNT IN R8
033042  005037  003712                  CLR     R9                      ;CLEAR THE BYTE ADDRESS COUNTER
033046  032737  000001  003710          BIT     #BIT0,R8                ;IS THE BYTE COUNT ODD
033054  001406                          BEQ     10$                     ;BRANCH IF NOT
033056  042737  000001  003710          BIC     #BIT0,R8                ;MAKE THE COUNT EVEN
033064  052764  000020  000026          BIS     #ODDFLG,LUNFLG(R4)      ;SET THE ODD BYTE FLAG
033072  012701  003624          10$:    MOV     #BYTADD,R1              ;LET R1 POINT TO THE ADDRESS TABLE
033076  022737  000003  002114          CMP     #3,L$TEST               ;ARE WE IN TEST 3 ?
033104  001003                          BNE     11$                     ;NO, SO JUST SET RDBUF IN BUFADR
033106  012702  070612                  MOV     #WRTBUF-2,R2            ;LET R2 POINT TO THE WRITE BUFFER
033112  000402                          BR      12$
033114  012702  070614          11$:    MOV     #WRTBUF,R2              ;LET R2 POINT TO THE WRITE BUFFER
033120  012703  050614          12$:    MOV     #RDBUF,R3               ;LET R3 POINT TO THE READ BUFFER
033124  012705  003650                  MOV     #DATBL,R5               ;LET R5 POINT TO THE ERROR DATA TABLE
033130  022322                  14$:    CMP     (R3)+,(R2)+             ;COMPARE THE FIRST WORD OF DATA
033132  001447                          BEQ     25$                     ;BRANCH IF THEY ARE EQUAL
033134  126362  177776  177776          CMPB    LOBYTE(R3),LOBYTE(R2)   ;COMPARE THE LOW BYTE
033142  001415                          BEQ     15$                     ;BRANCH IF EQUAL
033144  005237  003622                  INC     CMPERR                  ;ADD 1 TO THE ERROR COUNT
033150  022701  003650                  CMP     #TBLEND,R1              ;IS THERE ROOM TO SAVE THIS DATA ?
```

```
 4605 033154  001410                          BEQ     15$                     ;BRANCH IF NOT
 4606 033156  116315  177776                  MOVB    LOBYTE(R3),(R5)         ;LOW READ BYTE IN SAVE BYTE
 4607 033162  116265  177776  000001          MOVB    LOBYTE(R2),ONE(R5)      ;LOW WRITE BYTE IN SAVE BYTE
 4608 033170  013721  003712                  MOV     R9,(R1)+                ;SAVE THE ADDRESS FOR PRINTING
 4609 033174  005725                          TST     (R5)+                   ;POINT R5 TO NEXT TABLE LOCATION
 4610 033176  005237  003712          15$:    INC     R9                      ;ADD 1 TO THE BYTE COUNT
 4611 033202  126362  177777  177777          CMPB    HIBYTE(R3),HIBYTE(R2)   ;COMPARE THE HIGH BYTE
 4612 033210  001415                          BEQ     20$                     ;BRANCH IF THEY ARE THE SAME
 4613 033212  005237  003622                  INC     CMPERR                  ;ADD 1 TO THE ERROR COUNT
 4614 033216  022701  003650                  CMP     #TBLEND,R1              ;ROOM TO SAVE THIS DATA ?
 4615 033222  001410                          BEQ     20$                     ;BRANCH IF NOT
 4616 033224  116315  177777                  MOVB    HIBYTE(R3),(R5)         ;HI READ BYTE TO LOW SAVE BYTE
 4617 033230  116265  177777  000001          MOVB    HIBYTE(R2),ONE(R5)      ;HIGH WRITE BYTE TO HIGH SAVE BYTE
 4618 033236  013721  003712                  MOV     R9,(R1)+                ;SAVE THE ADDRESS TO PRINT
 4619 033242  005725                          TST     (R5)+                   ;POINT R5 TO THE NEXT TABLE LOCATION
 4620 033244  005237  003712          20$:    INC     R9                      ;ADD 1 TO THE BYTE COUNTER
 4621 033250  000403                          BR      30$                     ;BRANCH
 4622 033252  062737  000002  003712  25$:    ADD     #2,R9                   ;ADD 2 TO THE BYTE COUNT
 4623 033260  023737  003712  003710  30$:    CMP     R9,R8                   ;HAVE WE COMPARED THEM ALL ?
 4624 033266  001320                          BNE     14$                     ;NO, GO DO SOME MORE
 4625 033270  032764  000020  000026          BIT     #ODDFLG,LUNFLG(R4)      ;WAS IT AN ODD BYTE COUNT ?
 4626 033276  001414                          BEQ     35$                     ;BRANCH IF NOT
 4627 033300  121312                          CMPB    (R3),(R2)               ;COMPARE THE LOW BYTES
 4628 033302  001412                          BEQ     35$                     ;BRANCH IF THEY MATCH
 4629 033304  005237  003622                  INC     CMPERR                  ;ADD 1 TO THE ERROR COUNT
 4630 033310  022701  003650                  CMP     #TBLEND,R1              ;IS THERE ROOM FOR THIS DATA ?
 4631 033314  001405                          BEQ     35$                     ;BRANCH IF NOT
 4632 033316  111315                          MOVB    (R3),(R5)               ;LOW READ BYTE IN THE LOW SAVE BYTE
 4633 033320  111265  000001                  MOVB    (R2),ONE(R5)            ;LOW WRITE BYTE TO HIGH SAVE BYTE
 4634 033324  013721  003712                  MOV     R9,(R1)+                ;SAVE THE ADDRESS TO PRINT
 4635 033330  005737  003622          35$:    TST     CMPERR                  ;DID WE HAVE ANY ERRORS ?
 4636 033334  001423                          BEQ     40$                     ;BRANCH TO EXIT IF NOT
 4637 033336  012705  011276                  MOV     #DCMPT,R5               ;POINT R5 TO THE DATA COMPARE ERROR TABLE
 4638 033342  012702  013166                  MOV     #L$ERRTBL,R2            ;POINT R2 TO THE ERROR TABLE
 4639 033346  012522                          MOV     (R5)+,(R2)+             ;MOVE THE DATA COMPARE COMPARE TABLE
 4640 033350  012522                          MOV     (R5)+,(R2)+             ;TO THE ERROR TABLE
 4641 033352  012522                          MOV     (R5)+,(R2)+             ;
 4642 033354  011512                          MOV     (R5),(R2)               ;
 4643 033356                                  POP     <R5,R3,R2,R1>           ;RESTORE THE REGISTERS
      033356  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
      033360  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
      033362  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
      033364  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 4644 033366  004737  032624                  JSR     PC,ERRTLY               ;GO TALLY THE ERROR
 4645 033372  105037  013167                  CLRB    ERRTYP+1                ;CLEAR THE UPPER BYTE OF ERROR TYPE
 4646 033376  004737  032714                  JSR     PC,PRIERR               ;GO PRINT THE ERROR
 4647 033402  000404                          BR      45$                     ;EXIT
 4648 033404                          40$:    POP     <R5,R3,R2,R1>           ;RESTORE THE REGISTERS
      033404  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
      033406  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
      033410  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
      033412  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 4649 033414  000240                  45$:    NOP     ;TEMP
 4650 033416  000207                          RTS     PC                      ;RETURN
```

```
                                .SBTTL  UNJAM
                                ;**********************************
                                ;
                                ; UNJAM
                                ;
                                ;Called By       :
                                ;Calls To        :
                                ;Inputs          :
                                ;Outputs         :
                                ;Register Inputs :
                                ;Register Outputs:
                                ;
033420                          UNJAM::
033420  005737  003522                  TST     DUMPKT                  ;ARE WE ISSUEING NULL COMMANDS
033424  001444                          BEQ     15$                     ;YES , THEN EXIT
033426  023764  003724  000006          CMP     SECRNS,CMDSEQ(R4)       ;IS IT THE ONLY COMMAND OUT ?
033434  001440                          BEQ     15$                     ;YES , THEN EXIT
033436  016437  000006  003614          MOV     CMDSEQ(R4),SAVDIF       ;SET UP TO UNJAM THE QUEUES
033444  163737  003724  003614          SUB     SECRNS,SAVDIF           ;SUBTRACT CURRENT FROM THE HIGHEST
033452  063737  003614  003536          ADD     SAVDIF,CMDCNT           ;ADJUST THE COMMAND COUNT
033460  063737  003614  003540          ADD     SAVDIF,RSPCNT           ;ADJUST THE RESPONSE COUNT
033466  042737  000100  003674          BIC     #CMDONE,PCFLAG          ;CLEAR THE ALL COMMANDS ISSUED FLAG
033474  163764  003614  000034          SUB     SAVDIF,OBJFDL(R4)       ;ADUST THE OBJECT COUNT
033502  103002                          BCC     5$                      ;GET OUT IF NO CARRY
033504  005364  000036                  DEC     OBJFDH(R4)              ;OTHERWISE, ADJUST THE HIGH WORD
033510  022737  000004  003602  5$:     CMP     #N,SUBCNT
033516  001002                          BNE     10$
033520  005037  003602                  CLR     SUBCNT
033524  005237  003602          10$:    INC     SUBCNT
033530  005337  003614                  DEC     SAVDIF
033534  001365                          BNE     5$
033536  052764  000002  000026  15$:    BIS     #SEREXC,LUNFLG(R4)      ;SET THE SERIOUS EXCEPTION FLAG
033544  000240                          NOP     ;TEMP
033546  000207                          RTS     PC                      ;RETURN
033550                                  ENDMOD
```

```
                                        .SBTTL  CLEAR EOT
                                        ;**********************************
                                        ;
                                        ; CLEAR EOT
                                        ;

033550                                  CLREOT::
033550                                          PUSH    <R1,R2,R4>              ;SAVE R1, R2, AND R4
033550  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
033552  010246                                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
033554  010446                                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
033556  005002                                  CLR     R2                      ;CLEAR OUT R2
033560  012701  177776                          MOV     #-2.,R1                 ;SET R1 TO THE FIRST UNIT
033564  012704  002322                          MOV     #LUNO,R4                ;LET R4 EQUAL THE FIRDT LUN
033570  062701  000002          5$:             ADD     #2.,R1                  ;ADD 2 TO THE UNIT POINTER
033574  062702  000001                          ADD     #1.,R2                  ;ADD 1 TO R2
033600  062704  000224                          ADD     #LUNSTP,R4              ;SET R4 TO THE NEXT LUN
033604  042761  000004  003526                  BIC     #EOT,DRINUS(R1)         ;CLEAR THE EOT BIT IN DRINUS
033612  005037  003706                          CLR     UEOT                    ;CLEAR THE EOT FLAG
033616  C23702  002012          10$:            CMP     L$UNIT,R2               ;HAVE WE DONE THEM ALL
033622  001362                                  BNE     5$                      ;NO, KEEP GOING TILL ALL DONE
033624                                          POP     <R4,R2,R1>              ;RESTORE R4, R2, AND R1
033624  012604                                  MOV     (SP)+,R4        ;;POP STACK INTO R4
033626  012602                                  MOV     (SP)+,R2        ;;POP STACK INTO R2
033630  012601                                  MOV     (SP)+,R1        ;;POP STACK INTO R1
033632  000240                                  NOP     ;TEMP
033634  000207                                  RTS     PC                      ;RETURN
```

```
                                .SBTTL  SEED SETUP AND SAVE
                                ;**********************************
                                ;
                                ; SEED SETUP
                                ;

033636                          SDSTUP::
033636                                  PUSH    <R1,R4>                 ;
033636  010146                          MOV     R1,-(SP)                ;;PUSH R1 ON STACK
033640  010446                          MOV     R4,-(SP)                ;;PUSH R4 ON STACK
033642  013701  002012                  MOV     L$UNIT,R1               ;
033646  012704  002322                  MOV     #LUNO,R4                ;
033652  016464  000174  000202  5$:     MOV     SED1(R4),SEED1(R4)      ;
033660  016464  000176  000204          MOV     SED2(R4),SEED2(R4)      ;
033666  016464  000200  000206          MOV     SED3(R4),SEED3(R4)      ;
033674  062704  000224                  ADD     #LUNSTP,R4              ;
033700  005301                          DEC     R1                      ;
033702  001363                          BNE     5$                      ;
033704                                  POP     <R4,R1>                 ;
033704  C12604                          MOV     (SP)+,R4                ;;POP STACK INTO R4
033706  012601                          MOV     (SP)+,R1                ;;POP STACK INTO R1
033710  000240                          NOP     ;TEMP
033712  000207                          RTS     PC                      ;

                                ;**********************************
                                ;
                                ; SEED SAVE
                                ;

033714                          SDSAVE::
033714                                  PUSH    <R1,R4>                 ;
033714  010146                          MOV     R1,-(SP)                ;;PUSH R1 ON STACK
033716  010446                          MOV     R4,-(SP)                ;;PUSH R4 ON STACK
033720  013701  002012                  MOV     L$UNIT,R1               ;
033724  012704  002322                  MOV     #LUNO,R4                ;
033730  016464  000202  000174  5$:     MOV     SEED1(R4),SED1(R4)      ;
033736  016464  000204  000176          MOV     SEED2(R4),SED2(R4)      ;
033744  016464  000206  000200          MOV     SEED3(R4),SED3(R4)      ;
033752  062704  000224                  ADD     #LUNSTP,R4              ;
033756  005301                          DEC     R1                      ;
033760  001363                          BNE     5$                      ;
033762                                  POP     <R4,R1>                 ;
033762  012604                          MOV     (SP)+,R4                ;;POP STACK INTO R4
033764  012601                          MOV     (SP)+,R1                ;;POP STACK INTO R1
033766  000240                          NOP     ;TEMP
033770  000207                          RTS     PC                      ;
```

```
                                .SBTTL  PATTERN CLEAR
                                ;**********************************
                                ;
                                ; PATTERN CLEAR
                                ;
                                ;THIS ROUTINE DOES NOT SAVE R4 AND THEREFORE ›HOULD NOT BE CALLED FROM ANY
                                ;PLACE OTHER THAN A TEST.

033772                          PATCLR:
033772  012704  002322                MOV     @LUN0,R4                ;
033776  005064  000024          1$:   CLR     PATSAV(R4)              ;
034002  022704  003216                CMP     @LUN3,R4                ;
034006  001403                       BEQ     2$                      ;
034010  062704  000224                ADD     @LUNSTP,R4              ;
034014  000770                       BR      1$                      ;
034016  000240                2$:    NOP     ;TEMP
034020  000207                       RTS     PC                      ;
```

```
4773                                    .SBTTL  CORE DUMP
4774                                    ;**********************************
4775                                    ;
4776                                    ; CORE DUMP
4777                                    ;
4778                                    ;THIS ROUTINE IS DESIGNED TO DUMP ALL CRITICAL MEMORY LOCATIONS ON
4779                                    ;OCCURRENCE OF ERRORS, WHEN ENEABLED BY THE OPERATOR VIA THE SOFTWARE
4780                                    ;QUESTIONS.  IT IS INTENDED PRIMARILY AS AN AID TO DEBUGGING THE
4781                                    ;PROGRAM, BUT MAY PROVE U EFUL IN ANALYZING CERTAIN DEVICE ERRORS
4782                                    ;AS WELL.
4783
4784 034022                             CORDMP:
4785 034022                                     PUSH    <R1,R2>
     034022  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
     034024  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
4786 034026                                     PRINTF  @DUMP,R1,R2,R3,R4,R5
     034026  010546                            MOV     R5,-(SP)
     034030  010446                            MOV     R4,-(SP)
     034032  010346                            MOV     R3,-(SP)
     034034  010246                            MOV     R2,-(SP)
     034036  010146                            MOV     R1,-(SP)
     034040  012746  020344                    MOV     @DUMP,-(SP)
     034044  012746  000006                    MOV     @6,-(SP)
     034050  010600                            MOV     SP,R0
     034052  104417                            TRAP    C$PNTF
     034054  062706  000016                    ADD     @16,SP
4787
4788                                    ;       MOV     @LUNO,R1                ;PUT STARING ADDRESS IN R1
4789                                    ;       MOV     @LUNO,R2                ;AND ANOTHER COPY IN R2
4790                                    ;1$:    PRINTF  @DUMP2,R1,(R2),2(R2),4(R2),6(R2)
4791                                    ;
4792                                    ;       ADD     @10,R1                  ;UPDATE R1
4793                                    ;       ADD     @10,R2                  ;UPDATE R2
4794                                    ;       CMP     @IOERTB,R1              ;ARE WE AT THE END OF DUMP AREA
4795                                    ;       BHI     1$                      ;KEEP GOING IF NOT
4796
4797 034060  012701  002322                    MOV     @LUNO,R1                ;PUT STARING ADDRESS IN R1
4798 034064  012702  002322                    MOV     @LUNO,R2                ;AND ANOTHER COPY IN R2
4799 034070                             1$:     PRINTF  @DUMP2,R1,(R2),2(R2),4(R2),6(R2)
     034070  016246  000006                    MOV     6(R2),-(SP)
     034074  016246  000004                    MOV     4(R2),-(SP)
     034100  016246  000002                    MOV     2(R2),-(SP)
     034104  011246                            MOV     (R2),-(SP)
     034106  010146                            MOV     R1,-(SP)
     034110  012746  020466                    MOV     @DUMP2,-(SP)
     034114  012746  000006                    MOV     @6,-(SP)
     034120  010600                            MOV     SP,R0
     034122  104417                            TRAP    C$PNTF
     034124  062706  000016                    ADD     @16,SP
4800
4801 034130  062701  000010                    ADD     @10,R1                  ;UPDATE R1
4802 034134  062702  000010                    ADD     @10,R2                  ;UPDATE R2
4803 034140  022701  002546                    CMP     @LUN1,R1                ;ARE WE AT THE END OF DUMP AREA
4804 034144  101351                            BHI     1$                      ;KEEP GOING IF NOT
4805
4806 034146  012701  003442                    MOV     @PCMDBF,R1              ;PUT STARING ADDRESS IN R1
4807 034152  012702  003442                    MOV     @PCMDBF,R2              ;AND ANOTHER COPY IN R2
```

```
4808 034156                                    PRINTF  #LINE
     034156  012746  020524                    MOV     #LINE,-(SP)
     034162  012746  000001                    MOV     #1, (SP)
     034166  010600                            MOV     SP,R0
     034170  104417                            TRAP    C$PNTF
     034172  062706  000004                    ADD     #4,SP
4809 034176                            2$:     PRINTF  #DUMP2,R1,(R2),2(R2),4(R2),6(R2)
     034176  016246  000006                    MOV     6(R2),-(SP)
     034202  016246  000004                    MOV     4(R2),-(SP)
     034206  016246  000002                    MOV     2(R2),-(SP)
     034212  011246                            MOV     (R2),-(SP)
     034214  010146                            MOV     R1,-(SP)
     034216  012746  020466                    MOV     #DUMP2,-(SP)
     034222  012746  000006                    MOV     #6,-(SP)
     034226  010600                            MOV     SP,R0
     034230  104417                            TRAP    C$PNTF
     034232  062706  000016                    ADD     #16,SP
4810
4811 034236  062701  000010                    ADD     #10,R1                  ;UPDATE R1
4812 034242  062702  000010                    ADD     #10,R2                  ;UPDATE R2
4813 034246  022701  005252                    CMP     #RSPBF1,R1              ;ARE WE AT THE END OF DUMP AREA
4814 034252  101351                            BHI     2$                      ;KEEP GOING IF NOT
4815
4816 034254  012701  010732                    MOV     #IOSTAT,R1              ;PUT STARING ADDRESS IN R1
4817 034260  012702  010732                    MOV     #IOSTAT,R2              ;AND ANOTHER COPY IN R2
4818 034264                                    PRINTF  #LINE
     034264  012746  020524                    MOV     #LINE, (SP)
     034270  012746  000001                    MOV     #1,-(SP)
     034274  010600                            MOV     SP,R0
     034276  104417                            TRAP    C$PNTF
     034300  062706  000004                    ADD     #4,SP
4819 034304                            3$.     PRINTF  #DUMP2,R1,(R2),2(R2),4(R2),6(R2)
     034304  016246  000006                    MOV     6(R2),-(SP)
     034310  016246  000004                    MOV     4(R2),-(SP)
     034314  016246  000002                    MOV     2(R2),-(SP)
     034320  011246                            MOV     (R2),-(SP)
     034322  010146                            MOV     R1,-(SP)
     034324  012746  020466                    MOV     #DUMP2, (SP)
     034330  012746  000006                    MOV     #6,-(SP)
     034334  010600                            MOV     SP,R0
     034336  104417                            TRAP    C$PNTF
     034340  062706  000016                    ADD     #16,SP
4820
4821 034344  062701  000010                    ADD     #10,R1                  ;UPDATE R1
4822 034350  062702  000010                    ADD     #10,R2                  ;UPDATE R2
4823 034354  022701  010756                    CMP     #IOERTB,R1              ;ARE WE AT THE END OF DUMP AREA
4824 034360  101351                            BHI     3$                      ;KEEP GOING IF NOT
4825
4826 034362                                    PRINTF  #LINE
     034362  012746  020524                    MOV     #LINE,-(SP)
     034366  012746  000001                    MOV     #1,-(SP)
     034372  010600                            MOV     SP,R0
     034374  104417                            TRAP    C$PNTF
     034376  062706  000004                    ADD     #4,SP
4827 034402                                    POP     <R2,R1>
4828 034406  000240                            NOP     ;TEMP
4829 034410  000207                            RTS     PC
```

CORE DUMP

4830

```
4832                                    .SBTTL BUFFER DUMP
4833                                    ;**********************************
4834                                    ;
4835                                    ; BUFFER DUMP
4836                                    ;
4837                                    ;THIS ROUTINE WILL PRINT THE READ BUFFER FOR EVERY RECORD READ IN
4838                                    ;TEST 5.
4839
4840 034412                            BUFDMP:
4841 034412                                    PUSH    <R1,R2>
4842 034416  016301  000014                    MOV     P.BCNT(R3),R1           ;GET NUMBER OF BYTES X-FERRED
4843 034422  012702  050614                    MOV     #RDBUF,R2               ;GET BUFFER ADDRESS
4844 034426                                    PRINTF  #DUMP1,R1               ;PRINT NUMBER OF BYTES
     034426  010146                            MOV     R1,-(SP)
     034430  012746  020425                    MOV     #DUMP1,-(SP)
     034434  012746  000002                    MOV     #2,-(SP)
     034440  010600                            MOV     SP,R0
     034442  104417                            TRAP    C$PNTF
     034444  062706  000006                    ADD     #6,SP
4845
4846 034450                            1$:     PRINTF  #DUMP2,(R2),2(R2),4(R2),6(R2),10(R2)
     034450  016246  000010                    MOV     10(R2),-(SP)
     034454  016246  000006                    MOV     6(R2),-(SP)
     034460  016246  000004                    MOV     4(R2),-(SP)
     034464  016246  000002                    MOV     2(R2),-(SP)
     034470  011246                            MOV     (R2),-(SP)
     034472  012746  020466                    MOV     #DUMP2,-(SP)
     034476  012746  000006                    MOV     #6,-(SP)
     034502  010600                            MOV     SP,R0
     034504  104417                            TRAP    C$PNTF
     034506  062706  000016                    ADD     #16,SP
4847 034512  062702  000012                    ADD     #12,R2                  ;ADJUST BUFFER POINTER
4848 034516  162701  000012                    SUB     #12,R1                  ;ADJUST BYTE COUNT
4849 034522  001401                            BEQ     5$                      ;IF ZERO GET OUT
4850 034524  100351                            BPL     1$                      ;KEEP GOING IF POSITIVE
4851
4852 034526                            5$:     POP     <R2,R1>
4853 034532  000240                            NOP     ;TEMP
4854 034534  000207                            RTS     PC
4855
```

```
                                .TITLE MISCELLANEOUS SECTIONS
                                .SBTTL  REPORT CODING SECTION

034536                                  BGNMOD

                                ;++
                                ; THE REPORT CODING SECTION CONTAINS THE
                                ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                                ;--

034536                                  BGNRPT
034536                          L$RPT::

034536                                  PUSH    <R1,R4,R5>
034536  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
034540  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
034542  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
034544  032737  000200  003674          BIT     #DROPIT,PCFLAG          ;ARE WE DROPPING A UNIT
034552  001015                          BNE     5$                      ;YES, ONLY PRINT STATS FOR THIS UNIT
034554  005001                          CLR     R1                      ;SET R1 TO FIRST UNIT
034556  005037  002074                  CLR     L$LUN                   ;START WITH UNIT 0
034562  012704  002322                  MOV     #LUNO,R4                ;START WITH LUN BLOCK FOR UNIT 0
034566  013705  002012                  MOV     L$UNIT,R5               ;INIT UNIT COUNTER
034572  032761  000010  003526  1$:     BIT     #DROP,DRINUS(R1)        ;HAS THE DRIVE BEEN DROPPED ?
034600  001402                          BEQ     5$                      ;NO, PRINT ITS STATS
034602  000137  035652                  JMP     15$                     ;OTHERWISE GET THE NEXT DRIVE

034606                          5$:     PRINTS  #STAT01,L$LUN
034606  013746  002074                  MOV     L$LUN,-(SP)
034612  012746  035770                  MOV     #STAT01,-(SP)
034616  012746  000002                  MOV     #2,-(SP)
034622  010600                          MOV     SP,R0
034624  104416                          TRAP    C$PNTS
034626  062706  000006                  ADD     #6,SP

034632                                  PRINTS  #STAT02
034632  012746  036036                  MOV     #STAT02,-(SP)
034636  012746  000001                  MOV     #1,-(SP)
034642  010600                          MOV     SP,R0
034644  104416                          TRAP    C$PNTS
034646  062706  000004                  ADD     #4,SP

034652                                  PRINTS  #STAT04
034652  012746  036103                  MOV     #STAT04,-(SP)
034656  012746  000001                  MOV     #1,-(SP)
034662  010600                          MOV     SP,R0
034664  104416                          TRAP    C$PNTS
034666  062706  000004                  ADD     #4,SP

034672                                  PRINTS  #STAT05,GSTEWR(R4),GSTERD(R4),GSTEUA(R4)
034672  016446  000044                  MOV     GSTEUA(R4),-(SP)
034676  016446  000042                  MOV     GSTERD(R4),-(SP)
034702  016446  000040                  MOV     GSTEWR(R4),-(SP)
034706  012746  036157                  MOV     #STAT05,-(SP)
034712  012746  000004                  MOV     #4,-(SP)
034716  010600                          MOV     SP,R0
034720  104416                          TRAP    C$PNTS
```

```
                034722  062706  000012          ADD     #12,SP
        4929
        4930    034726                          PRINTS  #STAT06,GSFTWR(R4),GSFTRD(R4)
                034726  016446  000050          MOV     GSFTRD(R4),-(SP)
                034732  016446  000046          MOV     GSFTWR(R4),-(SP)
                034736  012746  036215          MOV     #STAT06,-(SP)
                034742  012746  000003          MOV     #3,-(SP)
                034746  010600                  MOV     SP,R0
                034750  104416                  TRAP    C$PNTS
                034752  062706  000010          ADD     #10,SP
        4931
        4932    034756                          PRINTS  #STAT07,GHRDWR(R4),GHRDRD(R4),GHRDUA(R4)
                034756  016446  000056          MOV     GHRDUA(R4),-(SP)
                034762  016446  000054          MOV     GHRDRD(R4),-(SP)
                034766  016446  000052          MOV     GHRDWR(R4),-(SP)
                034772  012746  036256          MOV     #STAT07,-(SP)
                034776  012746  000004          MOV     #4,-(SP)
                035002  010600                  MOV     SP,R0
                035004  104416                  TRAP    C$PNTS
                035006  062706  000012          ADD     #12,SP
        4933
        4934    035012                          PRINTS  #STAT08,GMEDER(R4)
                035012  016446  000060          MOV     GMEDER(R4),-(SP)
                035016  012746  036314          MOV     #STAT08,-(SP)
                035022  012746  000002          MOV     #2,-(SP)
                035026  010600                  MOV     SP,R0
                035030  104416                  TRAP    C$PNTS
                035032  062706  000006          ADD     #6,SP
        4935
        4936    035036                          PRINTS  #STAT09,GDCERR(R4)
                035036  016446  000062          MOV     GDCERR(R4),-(SP)
                035042  012746  036360          MOV     #STAT09,-(SP)
                035046  012746  000002          MOV     #2,-(SP)
                035052  010600                  MOV     SP,R0
                035054  104416                  TRAP    C$PNTS
                035056  062706  000006          ADD     #6,SP
        4937
        4938    035062                          PRINTS  #STAT10,GOTHWR(R4),GOTHRD(R4),GOTHUA(R4)
                035062  016446  000070          MOV     GOTHUA(R4),-(SP)
                035066  016446  000066          MOV     GOTHRD(R4),-(SP)
                035072  016446  000064          MOV     GOTHWR(R4),-(SP)
                035076  012746  036424          MOV     #STAT10,-(SP)
                035102  012746  000004          MOV     #4,-(SP)
                035106  010600                  MOV     SP,R0
                035110  104416                  TRAP    C$PNTS
                035112  062706  000012          ADD     #12,SP
        4939
        4940    035116                          PRINTS  #STAT11,GWRBY4(R4),GWRBY3(R4),GWRBY2(R4),GWRBY1(R4)
                035116  016446  000134          MOV     GWRBY1(R4),-(SP)
                035122  016446  000136          MOV     GWRBY2(R4),-(SP)
                035126  016446  000140          MOV     GWRBY3(R4),-(SP)
                035132  016446  000142          MOV     GWRBY4(R4),-(SP)
                035136  012746  036462          MOV     #STAT11,-(SP)
                035142  012746  000005          MOV     #5,-(SP)
                035146  010600                  MOV     SP,R0
                035150  104416                  TRAP    C$PNTS
                035152  062706  000014          ADD     #14,SP
```

```
4941
4942 035156                                  PRINTS  #STAT12,GRDBY4(R4),GRDBY3(R4),GRDBY2(R4),GRDBY1(R4)
     035156  016446  000144                  MOV     GRDBY1(R4),-(SP)
     035162  016446  000146                  MOV     GRDBY2(R4),-(SP)
     035166  016446  000150                  MOV     GRDBY3(R4),-(SP)
     035172  016446  000152                  MOV     GRDBY4(R4),-(SP)
     035176  012746  036534                  MOV     #STAT12,-(SP)
     035202  012746  000005                  MOV     #5,-(SP)
     035206  010600                          MOV     SP,R0
     035210  104416                          TRAP    C$PNTS
     035212  062706  000014                  ADD     #14,SP
4943
4944 035216                                  PRINTS  #STAT13,GCRDRP(R4)
     035216  016446  000072                  MOV     GCRDRP(R4),-(SP)
     035222  012746  036606                  MOV     #STAT13,-(SP)
     035226  012746  000002                  MOV     #2,-(SP)
     035232  010600                          MOV     SP,R0
     035234  104416                          TRAP    C$PNTS
     035236  062706  000006                  ADD     #6,SP
4945
4946 035242                                  PRINTS  #STAT03
     035242  012746  036061                  MOV     #STAT03,-(SP)
     035246  012746  000001                  MOV     #1,-(SP)
     035252  010600                          MOV     SP,R0
     035254  104416                          TRAP    C$PNTS
     035256  062706  000004                  ADD     #4,SP
4947
4948 035262                                  PRINTS  #STAT04
     035262  012746  036103                  MOV     #STAT04,-(SP)
     035266  012746  000001                  MOV     #1,-(SP)
     035272  010600                          MOV     SP,R0
     035274  104416                          TRAP    C$PNTS
     035276  062706  000004                  ADD     #4,SP
4949
4950 035302                                  PRINTS  #STAT05,PSTEWR(R4),PSTERD(R4),PSTEUA(R4)
     035302  016446  000102                  MOV     PSTEUA(R4),-(SP)
     035306  016446  000100                  MOV     PSTERD(R4),-(SP)
     035312  016446  000076                  MOV     PSTEWR(R4),-(SP)
     035316  012746  036157                  MOV     #STAT05,-(SP)
     035322  012746  000004                  MOV     #4,-(SP)
     035326  010600                          MOV     SP,R0
     035330  104416                          TRAP    C$PNTS
     035332  062706  000012                  ADD     #12,SP
4951
4952 035336                                  PRINTS  #STAT06,PSFTWR(R4),PSFTRD(R4)
     035336  016446  000106                  MOV     PSFTRD(R4),-(SP)
     035342  016446  000104                  MOV     PSFTWR(R4),-(SP)
     035346  012746  036215                  MOV     #STAT06,-(SP)
     035352  012746  000003                  MOV     #3,-(SP)
     035356  010600                          MOV     SP,R0
     035360  104416                          TRAP    C$PNTS
     035362  062706  000010                  ADD     #10,SP
4953
4954 035366                                  PRINTS  #STAT07,PHRDWR(R4),PHRDRD(R4),PHRDUA(R4)
     035366  016446  000114                  MOV     PHRDUA(R4),-(SP)
     035372  016446  000112                  MOV     PHRDRD(R4),-(SP)
     035376  016446  000110                  MOV     PHRDWR(R4),-(SP)
```

```
                035402  012746  036256          MOV     #STAT07,-(SP)
                035406  012746  000004          MOV     #4,-(SP)
                035412  010600                  MOV     SP,R0
                035414  104416                  TRAP    C$PNTS
                035416  062706  000012          ADD     #12,SP
          4955
          4956  035422                          PRINTS  #STAT08,PMEDER(R4)
                035422  016446  000116          MOV     PMEDER(R4),-(SP)
                035426  012746  036314          MOV     #STAT08,-(SP)
                035432  012746  000002          MOV     #2,-(SP)
                035436  010600                  MOV     SP,R0
                035440  104416                  TRAP    C$PNTS
                035442  062706  000006          ADD     #6,SP
          4957
          4958  035446                          PRINTS  #STAT09,PDCERR(R4)
                035446  016446  000120          MOV     PDCERR(R4),-(SP)
                035452  012746  036360          MOV     #STAT09,-(SP)
                035456  012746  000002          MOV     #2,-(SP)
                035462  010600                  MOV     SP,R0
                035464  104416                  TRAP    C$PNTS
                035466  062706  000006          ADD     #6,SP
          4959
          4960  035472                          PRINTS  #STAT10,POTHWR(R4),POTHRD(R4),POTHUA(R4)
                035472  016446  000126          MOV     POTHUA(R4),-(SP)
                035476  016446  000124          MOV     POTHRD(R4),-(SP)
                035502  016446  000122          MOV     POTHWR(R4),-(SP)
                035506  012746  036424          MOV     #STAT10,-(SP)
                035512  012746  000004          MOV     #4,-(SP)
                035516  010600                  MOV     SP,R0
                035520  104416                  TRAP    C$PNTS
                035522  062706  000012          ADD     #12,SP
          4961
          4962  035526                          PRINTS  #STAT11,PWRBY4(R4),PWRBY3(R4),PWRBY2(R4),PWRBY1(R4)
                035526  016446  000154          MOV     PWRBY1(R4),-(SP)
                035532  016446  000156          MOV     PWRBY2(R4),-(SP)
                035536  016446  000160          MOV     PWRBY3(R4),-(SP)
                035542  016446  000162          MOV     PWRBY4(R4),-(SP)
                035546  012746  036462          MOV     #STAT11,-(SP)
                035552  012746  000005          MOV     #5,-(SP)
                035556  010600                  MOV     SP,R0
                035560  104416                  TRAP    C$PNTS
                035562  062706  000014          ADD     #14,SP
          4963
          4964  035566                          PRINTS  #STAT12,PRDBY4(R4),PRDBY3(R4),PRDBY2(R4),PRDBY1(R4)
                035566  016446  000164          MOV     PRDBY1(R4),-(SP)
                035572  016446  000166          MOV     PRDBY2(R4),-(SP)
                035576  016446  000170          MOV     PRDBY3(R4),-(SP)
                035602  016446  000172          MOV     PRDBY4(R4),-(SP)
                035606  012746  036534          MOV     #STAT12,-(SP)
                035612  012746  000005          MOV     #5,-(SP)
                035616  010600                  MOV     SP,R0
                035620  104416                  TRAP    C$PNTS
                035622  062706  000014          ADD     #14,SP
          4965
          4966  035626                          PRINTS  #STAT13,PEDRP(R4)
                035626  016446  000130          MOV     PEDRP(R4),-(SP)
                035632  012746  036606          MOV     #STAT13,-(SP)
```

```
                035636  012746  000002                  MOV     #2,-(SP)
                035642  010600                          MOV     SP,R0
                035644  104416                          TRAP    C$PNTS
                035646  062706  000006                  ADD     #6,SP
     4967
     4968       035652  032737  000200  003674  15$:    BIT     #DROPIT,PCFLAG          ;ARE WE DROPPING A UNIT
     4969       035660  001036                          BNE     25$                     ;YES. ONLY PRINT STATS FOR THIS UNIT
     4970       035662  062704  000224                  ADD     #LUNSTP,R4              ;R4 POINTS TO NEXT LUN BLOCK
     4971       035666  062701  000002                  ADD     #2,R1                   ;POINT R1 TO THE NEXT UNIT
     4972       035672  005237  002074                  INC     L$LUN                   ;POINTS TO NEXT UNIT NUMBER
     4973       035676  005305                          DEC     R5                      ;ANY UNITS LEFT TO REPORT?
     4974       035700  001402                          BEQ     20$                     ;BRANCH IF NOT
     4975       035702  000137  034572                  JMP     1$                      ;ELSE, DO IT AGAIN
     4976       035706  105737  002216          20$:    TSTB    CLOCK                   ;IS THE CLOCK ENABLED
     4977       035712  001421                          BEQ     25$                     ;NO, THEN CAN'T PRINT TIME
     4978       035714                                  PRINTF  @TIME,<B,HOURS>,<B,MINUTE>,<B,SECOND>,
                035714  005046                          CLR     -(SP)
                035716  153716  002221                  BISB    SECOND,(SP)
                035722  005046                          CLR     -(SP)
                035724  153716  002220                  BISB    MINUTE,(SP)
                035730  005046                          CLR     -(SP)
                035732  153716  002217                  BISB    HOURS,(SP)
                035736  012746  020037                  MOV     #TIME,-(SP)
                035742  012746  000004                  MOV     #4,-(SP)
                035746  010600                          MOV     SP,R0
                035750  104417                          TRAP    C$PNTF
                035752  062706  000012                  ADD     #12,SP
     4979       035756                          25$:    POP     <R5,R4,R1>              ;RESTORE REGS
     4980       035764                                  EXIT    RPT
                035764  000167                          .WORD   J$JMP
                035766  000652                          .WORD   L10007-2-.
     4981
     4993
```

```
                                        ;FORMAT STATEMENTS FOR PRINT CALLS

035770      045      116      045  STAT01: .ASCIZ  ?≠N≠N≠ASTATISTICAL REPORT FOR UNIT ≠D1?
036036      045      116      045  STAT02: .ASCIZ  ?≠N≠A IN GCR MODE≠N?
036061      045      116      045  STAT03: .ASCIZ  ?≠N≠A IN PE MODE≠N?
036103      045      116      045  STAT04: .ASCIZ  ?≠N≠S8≠S8≠S5≠AWRITE≠S3≠AREAD≠S3≠AUNIT ACCESS?
036157      045      116      045  STAT05: .ASCIZ  ?≠N≠ASTATUS ERRORS    ≠D8≠D8≠D8?
036215      045      116      045  STAT06: .ASCIZ  ?≠N≠ASOFT ERRORS      ≠D8≠D8≠S7≠AO?
036256      045      116      045  STAT07: .ASCIZ  ?≠N≠ANON-RECV         ≠D8≠D8≠D8?
036314      045      116      045  STAT08: .ASCIZ  ?≠N≠AMEDIA            ≠D8≠S7≠AO≠S7≠AO?
036360      045      116      045  STAT09: .ASCIZ  ?≠N≠ADATA CMP ERRS    ≠S7≠AO≠D8≠S7≠AO?
036424      045      116      045  STAT10: .ASCIZ  ;≠N≠AOTHERS           ≠D8≠D8≠D8?
036462      045      116      045  STAT11: .ASCIZ  ?≠N≠ABYTES WRITTEN    ≠D3≠A,≠Z3≠A,≠Z3≠A,≠Z3?
036534      045      116      045  STAT12: .ASCIZ  ?≠.≠ABYTES READ       ≠D3≠A,≠Z3≠A,≠Z3≠A,≠Z3?
036606      045      116      045  STAT13: .ASCIZ  ?≠N≠ATIMES DROPPED    ≠D8≠N≠N?
                                        .EVEN

036642                                  ENDRPT
036642                              L10007:
036642  104425                          TRAP    C$RPT
```

```
                                        .SBTTL  INITIALIZE SECTION

                                        ;++
                                        ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
                                        ; AT THE BEGINNING OF EACH PASS.
                                        ;--

036644                                          BGNINIT
036644                                  L$INIT::

                                        STINIT::
036644                                          READEF  #EF.START
036644  012700  000040                          MOV     #EF.START,R0
036650  104447                                  TRAP    C$REFG
036652                                          BCOMPLETE       START
036652  103414                                  BCS     START

036654                                          READEF  #EF.RESTART
036654  012700  000037                          MOV     #EF.RESTART,R0
036660  104447                                  TRAP    C$REFG
036662                                          BCOMPLETE       START
036662  103410                                  BCS     START

036664                                          READEF  #EF.NEW
036664  012700  000035                          MOV     #EF.NEW,R0
036670  104447                                  TRAP    C$REFG
036672                                          BCOMPLETE       NUPASS
036672  103471                                  BCS     NUPASS

036674                                          READEF  #EF.CONTINUE
036674  012700  000036                          MOV     #EF.CONTINUE,R0
036700  104447                                  TRAP    C$REFG
036702                                          BCOMPLETE       NUPASS
036702  103465                                  BCS     NUPASS

036704  112737  000001  003750  START:  MOVB    #1,DAYS                 ;SET TO FIRST DAY
036712  005037  003700                  CLR     PASCNT                  ;CLEAR THE PASS COUNTER
036716  005037  003622                  CLR     CMPERR                  ;CLEAR THE COMPARE ERROR COUNTER
036722  012737  000004  003732          MOV     #TF.GCR,FORMAT          ;SET INITIAL TEST DENSITY TO GCR
036730  105737  002226                  TSTB    DENSITY                 ;INITIAL TEST DENSITY GCR ?
036734  001003                          BNE     1$                      ;BRANCH IF SO
036736  012737  000002  003732          MOV     #TF.PE,FORMAT           ;SET INITIAL TEST DENSITY TO PE

036744  013737  003732  003734  1$:     MOV     FORMAT,INFORM           ;SAVE INITIAL TEST DENSITY
036752  012701  002322                  MOV     #LUNO,R1                ;SET R1 TO THE FIRST LUN
036756  010102                  5$:     MOV     R1,R2                   ;LET R2 = R1
036760  062702  000210                  ADD     #URSPBF,R2              ;LET R2 = THE END OF THE CLEAR AREA
036764  005021                  10$:    CLR     (R1)+                   ;CLEAR THE LUN LOCATION
036766  020201                          CMP     R2,R1                   ;ARE WE AT THE END OF THE CLEAR AREA
036770  001375                          BNE     10$                     ;NO, KEEP CLEARING
036772  062701  000014                  ADD     #14,R1                  ;SET R1 TO THE NEXT LUN BLOCK
036776  022701  003442                  CMP     #PCMDBF,R1              ;HAVE WE DONE THEM ALL
037002  001365                          BNE     5$                      ;GO CLEAR THE NEXT LUN BLOCK

037004  005021                  15$:    CLR     (R1)+                   ;CLEAR THE LOCATION AND GET THE NEXT
037006  022701  003536                  CMP     #CMDCNT,R1              ;HAVE WE  CLEARED THEM ALL ?
037012  001374                          BNE     15$                     ;NO, KEEP GOING
```

INITIALIZE SECTION

```
5058
5059 037014  012704  002322                MOV     #LUNO,R4                ;SET R4 TO THE FIRST LUN
5060 037020  013702  002012                MOV     L$UNIT,R2               ;SET UP R2
5061 037024  012764  001233  000174  25$:   MOV     #RS1,SED1(R4)           ;SET UP THE SEED IN THE LUN BLOCK
5062 037032  012764  007622  000176        MOV     #RS2,SED2(R4)           ;SET UP THE SEED IN THE LUN BLOCK
5063 037040  012764  000000  000200        MOV     #RS3,SED3(R4)           ;SET UP THE SEED IN THE LUN BLOCK
5064
5065 037046  062704  000224          30$:  ADD     #LUNSTP,R4              ;SET UP THE NEXT LUN BLOCK
5066 037052  005302                        DEC     R2                      ;DECREMENT R2
5067 037054  001363                        BNE     25$                     ;DID YOU DO THEM ALL
5068
5069 037056                        NUPASS: BRESET
     037056  104433                        TRAP    C$RESET
5070
5071 037060  005037  003562                CLR     MANCNT                  ;CLEAR THRE RETRY COUNTER
5072 037064  013702  002012                MOV     L$UNIT,R2               ;SET UP R2
5073 037070  012704  002322          15$:  MOV     #LUNO,R4                ;SET R4 TO THE FIRST LUN
5074
5075 037074  005037  003674          25$:  CLR     PCFLAG                  ;CLEAR THE PROGRAM CONTROL FLAG.
5076 037100  005037  003706                CLR     UEOT                    ;CLEAR THE EOT FLAG
5077 037104  013737  002012  003704        MOV     L$UNIT,UDROP            ;SET UP THE DROP UNIT FLAG
5078 037112  005237  003700                INC     PASCNT                  ;ADD 1 TO PASS COUNTER
5079
5080 037116  012702  003752          30$:  MOV     #CMDBF1,R2              ;PUT COMMAND BUFFER ADDRESS IN R2
5081 037122  005022                  35$:  CLR     (R2)+                   ;CLEAR THE BUFFERS
5082 037124  022702  010412                CMP     #DSRNGO,R2              ;ARE WE AT THE END OF THE BUFFER ?
5083 037130  001374                        BNE     35$                     ;KEPP GOING TILL WE ARE
5084
5085 037132  012704  002322                MOV     #LUNO,R4                ;SET R4 TO THE FIRST LUN
5086 037136  013702  002012                MOV     L$UNIT,R2               ;SET UP R2
5087 037142  005001                        CLR     R1                      ;CLEAR R1
5088 037144  005003                        CLR     R3                      ;CLEAR R3
5089
5090 037146  032761  000020  003526  40$:  BIT     #FAIL,DRINUS(R1)        ;HAS THIS DRIVE FAILED ?
5091 037154  001054                        BNE     60$                     ;YES, GET NEXT UNIT
5092
5093 037156  032761  000010  003526  45$:  BIT     #DROP,DRINUS(R1)        ;DID THIS DRIVE DROP LAST TIME
5094 037164  001002                        BNE     50$                     ;YES, KEEP GOING
5095 037166  005064  000032                CLR     UNDROP(R4)              ;OTHERWISE CLEAR THE DROP COUNTER
5096 037172  012761  000001  003526  50$:  MOV     #AVB,DRINUS(R1)         ;SET UP ALL DRIVES TO AVAILABLE
5097
5098 037200  012764  003756  000014  55$:  MOV     #DCMDBF,CMUSAV(R4)      ;SET UP NEW COMMAND BUFFER SAVE
5099 037206  012764  003756  000016        MOV     #DCMDBF,COLSAV(R4)      ;SET UP OLD COMMAND BUFFER SAVE
5100 037214  016464  000220  000012        MOV     UCDSAG(R4),CMDSSV(R4)   ;SET UP COMMAND DESCRIPTOR SAVE
5101 037222  016464  000210  000020        MOV     URSPBF(R4),RNUSAV(R4)   ;SET UP NEW RESPONSE BUFFER SAVE
5102 037230  016464  000210  000022        MOV     URSPBF(R4),ROLSAV(R4)   ;SET UP OLD RESPONSE BUFFER SAVE
5103
5104 037236  005064  000006                CLR     CMDSEQ(R4)              ;CLEAR THE COMMAND REFERENCE NUMBER
5105 037242  005064  000034                CLR     OBJFDL(R4)              ;CLEAR THE LOW OBJECT FIELD
5106 037246  005064  000036                CLR     OBJFDH(R4)              ;CLEAR THE HIGH OBJECT FIELD
5107 037252  005064  000010                CLR     SLTUSE(R4)              ;CLEAR THE SLOT IN USE FLAG
5108 037256  010300                        MOV     R3,R0                   ;
5109
5110 037260                                GPHARD  R0,R0
     037260  104442                        TRAP    C$GPHRD
5111 037262                                BNCOMPLETE      60$
     037262  103011                        BCC     60$
```

INITIALIZE SECTION

```
5112
5113 037264  011064  000000                 MOV     (R0),TKIP(R4)           ;
5114 037270  012064  000002                 MOV     (R0)+,TKSA(R4)          ;
5115 037274  062764  000002  000002         ADD     #2,TKSA(R4)             ;
5116 037302  011064  000004                 MOV     (R0),TKUNIT(R4)         ;
5117
5118 037306  062701  000002          60$:   ADD     #2,R1                   ;SET R1 TO THE NEXT UNIT
5119 037312  062703  000001                 ADD     #1.,R3                  ;GET NEXT UNIT
5120 037316  062704  000224                 ADD     #LUNSTP,R4              ;SET UP THE NEXT LUN BLOCK
5121 037322  005302                         DEC     R2                      ;DECREMENT R2
5122 037324  001310                         BNE     40$                     ;DID YOU DO THEM ALL
5123
5124 037326  042737  000002  003674         BIC     #NCLKFL,PCFLAG          ;GET READY TO TEST FOR CLOCK PRESENT
5125 037334                                 SETVEC  #4,#NOCLK,#PRIOO        ;SET VECTOR 4 IN CASE NO CLOCK
     037334  012746  000000                 MOV     #PRIOO,-(SP)
     037340  012746  020646                 MOV     #NOCLK,-(SP)
     037344  012746  000004                 MOV     #4,-(SP)
     037350  012746  000003                 MOV     #3,-(SP)
     037354  104437                         TRAP    C$SVEC
     037356  062706  000010                 ADD     #10,SP
5126 037362  005737  177546                 TST     KWCSR                   ;IS THE CLOCK THERE ?
5127 037366  000240                         NOP
5128 037370  000240                         NOP
5129
5130 037372                                 CLRVEC  #4                      ;RETURN VECTOR TO TRAP CATCHER
     037372  012700  000004                 MOV     #4,R0
     037376  104436                         TRAP    C$CVEC
5131 037400  032737  000002  003674         BIT     #NCLKFL,PCFLAG          ;WAS A CLOCK PRESENT ?
5132 037406  001016                         BNE     ISTART                  ;NO CLOCK, START REGULAR INIT
5133 037410                                 SETVEC  #100,#KWHDL,#PRIOO      ;SET UP THE CLOCK VECTOR
     037410  012746  000000                 MOV     #PRIOO,-(SP)
     037414  012746  020702                 MOV     #KWHDL,-(SP)
     037420  012746  000100                 MOV     #100, (SP)
     037424  012746  000003                 MOV     #3,-(SP)
     037430  104437                         TRAP    C$SVEC
     037432  062706  000010                 ADD     #10,SP
5134 037436  012737  000100  177546         MOV     #100,KWCSR              ;ENABLE THE CLOCK INTERUPTS
5135
5136 037444  005001                 ISTART: CLR     R1                      ;SET R1 TO FIST UNIT
5137 037446  005037  002074                 CLR     L$LUN                   ;SET L$LUN TO FIRST UNIT
5138 037452  012704  002322                 MOV     #LUNO,R4                ;SET R4 TO THE FIRST LUN BLOCK
5139
5140 037456  032761  000001  003526   1$:   BIT     #AVB,DRINUS(R1)         ;SEE IF DRIVE IS PRESENT AND AVAILABLE
5141 037464  001501                         BEQ     15$                     ;GET THE NEXT DRIVE IF IT ISN'T
5142 037466  032761  000004  003526         BIT     #EOT,DRINUS(R1)         ;CHECK IF THE DRIVE IS AT EOT
5143 037474  001075                         BNE     15$                     ;GET NEXT DRIVE IF IT IS
5144
5145 037476  012764  000377  000010         MOV     #377,SLTUSE(R4)         ;SET ALL RESPONSE SLOTS TO THE PORT
5146 037504  004737  026476                 JSR     PC,PRTCLR               ;GO DO IT
5147 037510  112737  000004  010755         MOVB    #4,CRDLIM               ;CREDITS START AT 4 FOR NEW LUN
5148
5149 037516  012705  037724                 MOV     #INITIT,R5              ;PUT INIT TEST TABLE ADDRESS IN R5
5150 037522  004737  021366                 JSR     PC,CMMDSQ               ;GO DO INIT ON THIS DRIVE
5151 037526  032761  000001  003526         BIT     #AVB,DRINUS(R1)         ;SEE IF DRIVE IS PRESENT AND AVAILABLE
5152 037534  001455                         BEQ     15$                     ;GET THE NEXT DRIVE IF IT ISN'T
5153
5154 037536  012764  000377  000010         MOV     #377,SLTUSE(R4)         ;SET ALL RESPONSE SLOTS TO THE PORT
```

INITIALIZE SECTION

```
5155 037544  004737  026476                JSR     PC,PRTCLR               ;GO DO IT
5156 037550  112737  000004  010755        MOVB    #4,CRDLIM               ;CREDITS START AT 4 FOR NEW LUN
5157 037556  062705  000006                ADD     #TSTSTP,R5              ;POINT R5 TO THE SCC COMMAND
5158 037562  004737  021366                JSR     PC,CMMDSQ               ;GO DO SCC ON THIS DRIVE
5159 037566  032761  000001  003526        BIT     #AVB,DRINUS(R1)         ;SEE IF DRIVE IS PRESENT AND AVAILABLE
5160 037574  001435                        BEQ     15$                     ;GET THE NEXT DRIVE IF IT ISN T
5161
5162 037576  012764  000377  000010  5$:   MOV     #377,SLTUSE(R4)         ;SET ALL RESPONSE SLOTS TO THE PORT
5163 037604  004737  026476                JSR     PC,PRTCLR               ;GO DO IT
5164 037610  112737  000004  010755        MOVB    #4,CRDLIM               ;CREDITS START AT 4 FOR NEW LUN
5165 037616  062705  000006                ADD     #TSTSTP,R5              ;POINT R5 TO THE ONL COMMAND
5166 037622  004737  021366                JSR     PC,CMMDSQ               ;GO DO ONLINE ON THIS DRIVE
5167
5168 037626  012764  000377  000010  10$:  MOV     #377,SLTUSE(R4)         ;SET ALL RESPONSE SLOTS TO THE PORT
5169 037634  004737  026476                JSR     PC,PRTCLR               ;GO DO IT
5170 037640  112737  000004  010755        MOVB    #4,CRDLIM               ;CREDITS START AT 4 FOR NEW LUN
5171 037646  062705  000006                ADD     #TSTSTP,R5              ;POINT R5 TO THE GUS COMMAND
5172 037652  012737  000040  003616        MOV     #AVLB,TSTMSK            ;ALLOW UNIT AVAILABLE ERRORS
5173 037660  004737  021366                JSR     PC,CMMDSQ               ;GO DO GUS ON THIS DRIVE
5174 037664  005037  003616                CLR     TSTMSK                  ;ALLOW NO ERRORS
5175
5176 037670  022701  000006        15$:   CMP     #6.,R1                  ;HAVE WE DONE THEM ALL ?
5177 037674  001411                        BEQ     EXTINT                  ;GET OUT
5178 037676  062701  000002                ADD     #UNTSTP,R1              ;GET NEXT UNIT
5179 037702  062704  000224                ADD     #LUNSTP,R4              ;SET UP THE NEXT LUN BLOCK
5180 037706  005237  002074                INC     L$LUN                   ;GET NEXT UNIT
5181 037712                                BREAK
     037712  104422                        TRAP    C$BRK
5182 037714  000137  037456                JMP     1$                      ;GO DO THE NEXT ONE
5183
5184 037720                        EXTINT: EXIT    INIT
     037720  104432                        TRAP    C$EXIT
     037722  000032                        .WORD   L10010-.
5185
5186                                ;INIT TEST TABLE
5187 037724     170                INITIT::        .BYTE   INIT    ;INITIALIZATION TABLE
5188 037725     000                                .BYTE   NULPAT
5189 037726  000000                                .WORD   0
5190 037730  000001                                .WORD   1
5191 037732     230                                 BYTE   SCC     ;SET CONTROLLER CHARACTERISTICS TABLE
5192 037733     000                                .BYTE   NULPAT
5193 037734  000000                                .WORD   0
5194 037736  000001                                .WORD   1
5195 037740     140                                .BYTE   ONL     ;ONLINE TABLE
5196 037741     000                                .BYTE   NULPAT
5197 037742  000000                                .WORD   0
5198 037744  000001                                .WORD   1
5199 037746     220                                .BYTE   GUS     ;GET UNIT STATUS TABLE
5200 037747     000                                .BYTE   NULPAT
5201 037750  000000                                .WORD   0
5202 037752  000001                                .WORD   1
5203                                       .EVEN
5204
5205 037754                                ENDINIT
     037754                        L10010:
     037754  104411                        TRAP    C$INIT
```

INITIALIZE SECTION

```
   5207 037756                              BGNAUTO
        037756                    L$AUTO::
   5208 037756                              ENDAUTO
        037756                    L10011:
        037756  104461                     TRAP    C$AUTO
```

CLEANUP CODING SECTION

```
                                        .SBTTL  CLEANUP CODING SECTION
                                        ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
                                        ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
                                        ;--

 037760                                         BGNCLN
 037760                                 L$CLEAN::

 037760  032737  000002  003674                BIT     @NCLKFL,PCFLAG          ;WAS A CLOCK PRESENT ?
 037766  001005                                 BNE     5$                      ;NO CLOCK, DO REPORT
 037770  005037  177546                        CLR     KWCSR
 037774                                         CLRVEC  #100
 037774  012700  000100                        MOV     #100,R0
 040000  104436                                TRAP    C$CVEC
 040002                                 5$:     DORPT
 040002  104424                                TRAP    C$DRPT
 040004                                 EXTCLN: EXIT    CLN
 040004  104432                                TRAP    C$EXIT
 040006  000002                                .WORD   L10012-.

                                        .EVEN

 040010                                         ENDCLN
 040010                                 L10012:
 040010  104412                                TRAP    C$CLEAN
```

```
                                        .SBTTL  DROP UNIT SECTION
                                        ;++
                                        ; THE DROP UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                        ; TO NO LONGER BE TESTED.
                                        ;--

040012                                  DROPUN::
040012                                          PUSH    <R1>                    ;SAVE R1
040012  010146                                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
040014  010001                                 MOV     R0,R1                   ;POINT R1 TO THE DRINUS TABLE
040016  006101                                 ROL     R1                      ;MULTIPLY BY 2
040020  022737  000002  003732                 CMP     @TF.PE,FORMAT           ;ARE WE IN PE MODE ?
040026  001003                                 BNE     5$                      ;NO, GO INCREMENT GCR DROPS
040030  005264  000130                         INC     PEDRP(R4)               ;YES,  INCREMENT PE DROPS
040034  000402                                 BR      10$                     ;KEEP GOING
040036  005264  000072          5$:            INC     GCRDRP(R4)              ;INCREMENT GCR DROPS
040042  052737  000200  003674  10$:           BIS     @DROPIT,PCFLAG          ;SET THE UNIT DROP FLAG
040050                                          PUSH    <R0>                    ;SAVE R0
040050  010046                                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
040052                                          DORPT                           ;GO PRINT UNIT STATS
040052  104424                                 TRAP    C$DRPT
040054                                          POP     <R0>                    ;RESTORE R0
040054  012600                                 MOV     (SP)+,R0        ;;POP STACK INTO R0
040056  042737  000200  003674                 BIC     @DROPIT,PCFLAG          ;CLEAR THE DROP FLAG
040064  105737  002232                         TSTB    NOCLR                   ;DO WE WANT TO CLEAR STATS ON EPROR ?
040070  001415                                 BEQ     20$                     ;NO, DON'T CLEAR THE STATS
040072                                          PUSH    <R2,R3>
040072  010246                                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
040074  010346                                 MOV     R3,-(SP)        ;;PUSH R3 ON STACK
040076  012702  000040                         MOV     @GSTEWR,R2              ;STARTING ADDRESS OF STATS IN R2
040102  012703  000174                         MOV     @SED1,R3                ;END ADDRESS OF STATS IN R3
040106  060402                                 ADD     R4,R2                   ;ADD THE LUN BLOCK ADDRESS TO R2
040110  060403                                 ADD     R4,R3                   ;ADD THE LUN BLOCK ADDRESS TO R3
040112  005022                  15$:            CLR     (R2)+                   ;CLEAR THE LOCATION
040114  020203                                 CMP     R2,R3                   ;ARE WE AT THE END OF THE STATS
040116  001375                                 BNE     15$                     ;NO, KEEP CLEARING
040120                                          POP     <R3,R2>
040120  012603                                 MOV     (SP)+,R3        ;;POP STACK INTO R3
040122  012602                                 MOV     (SP)+,R2        ;;POP STACK INTO R2
040124  042761  000001  003526  20$:           BIC     @AVB,DRINUS(R1)         ;CLEAR THE AVB BIT IN DRIVE IN USE TABLE
040132  032761  000004  003526                 BIT     @EOT,DRINUS(R1)         ;IS THE DRIVE AT EOT ?
040140  001404                                 BEQ     25$                     ;BRANCH IF NOT
040142  042761  000004  003526                 BIC     @EOT,DRINUS(R1)         ;CLEAR THE EOT BIT IN DRIVE IN USE TABLE
040150  000402                                 BR      30$                     ;GET OUT
040152  005337  003704          25$:           DEC     UDROP                   ;SUBTRACT 1 TO DROPPED FLAG
040156  052761  000010  003526  30$:           BIS     @DROP,DRINUS(R1)        ;SET DRIVE IN USE TABLE TO DROPPED
040164  005264  000032                         INC     UNDROP(R4)              ;ADD 1 TO THE UNIT DROP COUNT
040170  022764  000012  000032                 CMP     @10.,UNDROP(R4)         ;DO WE HAVE 10. ERRORS ?
040176  001004                                 BNE     35$                     ;NO, GET OUT
040200  052761  000020  003526                 BIS     @FAIL,DRINUS(R1)        ;SET THE DRIVE TO FAIL
040206                                          DODU    R0
040206  104451                                 TRAP    C$DODU
040210  005037  003552          35$:           CLR     RESPON                  ;CLEAR THE RESPONSE STATUS
040214                                          POP     <R1>                    ;RESTORE R1
040214  012601                                 MOV     (SP)+,R1        ;;POP STACK INTO R1
040216                                          DELAY   20.                     ;DELAY FOR AWHILE
```

DROP UNIT SECTION

```
        040216  012727  000024                  MOV     #20.,(PC)+
        040222  000000                          .WORD   0
        040224  013727  002116                  MOV     L$DLY,(PC)+
        040230  000000                          .WORD   0
        040232  005367  177772                  DEC     -6(PC)
        040236  001375                          BNE     .-4
        040240  005367  177756                  DEC      22(PC)
        040244  001367                          BNE     .-20
  5293  040246  010174  000000                  MOV     R1,@TKIP(R4)            ;FLUSH THE DRIVE
  5294  040252  012764  000377  000010          MOV     #377,SLTUSE(R4)         ;SET ALL RESPONCE SLOTS TO PORT
  5295  040260  004737  026476                  JSR     PC,PRTCLR               ;GO CLEAR THE PORT
  5296  040264  000240                          NOP     ;TEMP
  5297  040266  000207                          RTS     PC                      ;RETURN
  5298
  5299
  5300  040270                          BGNDU
        040270                  L$DU::
  5301  040270                          EXIT    DU
        040270  000167                  .WORD   J$JMP
        040272  C00000                  .WORD   L10013-2-.
  5302
  5314
  5315                                  .EVEN
  5316
  5317  040274                          ENDDU
        040274                  L10013:
        040274  104453                  TRAP    C$DU
```

```
                                .SBTTL  ADD UNIT SECTION

                                ;++
                                ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
                                ;  TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
                                ;  TO THE TEST CYCLE.
                                ;--

040276                                  BGNAU
040276                          L$AU::

                                        EXIT    AU
040276  000167                          .WORD   J$JMP
040300  000000                          .WORD   L10014-2-.

                                        .EVEN

                                        ENDAU
040302                          L10014:
040302  104452                          TRAP    C$AU

040304                                  ENDMOD
```

```
 5357
 5358                                   .TITLE HARDWARE TESTS
 5369
 5370                                   .SBTTL  TEST 1: Basic Function Test
 5406
 5407 040304                                   BGNMOD
 5408
 5409                                   ;++
 5410                                   ;This test will execute a subset of the  legal  commands  on  the  unit
 5411                                   ;under  test.   It  serves as a quick verify test to ascertain that the
 5412                                   ;unit can move tape and write/read  predictably,  without  error.   The
 5413                                   ;subset of legal commands will be issued in a coherent manner.
 5414                                   ;--
 5415
 5422
 5428
 5429 040304                                   BGNTST
      040304                           T1::
 5430
 5431 040304  C05737  003704           START1: TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
 5432 040310  001014                           BNE     5$                      ;NO, CONTINUE
 5433 040312                                   PRINTF  #BYPASS,L$TEST          ;PRINT THE TEST BYPASSED MESSAGE
      040312  013746  002114                   MOV     L$TEST,-(SP)
      040316  012746  020527                   MOV     #BYPASS,-(SP)
      040322  012746  000002                   MOV     #2,-(SP)
      040326  010600                           MOV     SP,R0
      040330  104417                           TRAP    C$PNTF
      040332  062706  000006                   ADD     #6,SP
 5434 040336  000137  042306                   JMP     T1EXIT                  ;GET OUT IF NONE LEFT TO TEST
 5435
 5436 040342  105737  002216           5$.     TSTB    CLOCK                   ;IS THE CLOCK ENABLED
 5437 040346  001421                           BEQ     G01                     ;NO, THEN CAN'T PRINT TIME
 5438 040350                                   PRINTF  #TIME,<B,HOURS>,<B,MINUTE>,<B,SECOND>,
      040350  005046                           CLR     -(SP)
      040352  153716  002221                   BISB    SECOND,(SP)
      040356  005046                           CLR     -(SP)
      040360  153716  002220                   BISB    MINUTE,(SP)
      040364  005046                           CLR     -(SP)
      040366  153716  002217                   BISB    HOURS,(SP)
      040372  012746  020037                   MOV     #TIME,-(SP)
      040376  012746  000004                   MOV     #4,-(SP)
      040402  010600                           MOV     SP,R0
      040404  104417                           TRAP    C$PNTF
      040406  062706  000012                   ADD     #12,SP
 5439
 5440 040412  004737  033550           G01:    JSR     PC,CLREOT               ;MAKE SURE EOT STATUS IS CLEAR
 5441 040416  012737  000100  003616           MOV     #ONLB,TSTMSK            ;ALLOW ALREADY ONLINE STATUS
 5442
 5443 040424  022737  000002  003732           CMP     #TF.PE,FORMAT           ;ARE WE DOING PE ?
 5444 040432  001011                           BNE     1$                      ;NO, PRINT GCR
 5445 040434                                   PRINTF  #TSTPE                  ;PRINT TESTING IN PE
      040434  012746  020610                   MOV     #TSTPE,-(SP)
      040440  012746  000001                   MOV     #1,-(SP)
      040444  010600                           MOV     SP,R0
      040446  104417                           TRAP    C$PNTF
      040450  062706  000004                   ADD     #4,SP
 5446 040454  000410                           BR      3$                      ;START TEST
```

```
5447 040456                                1$:     PRINTF  @TSTGCR                 ;PRINT TESTING IN GCR
     040456  012746  020562                        MOV     @TSTGCR,-(SP)
     040462  012746  000001                        MOV     #1,-(SP)
     040466  010600                                MOV     SP,R0
     040470  104417                                TRAP    C$PNTF
     040472  062706  000004                        ADD     #4,SP
5448 040476  012705  042324                3$:     MOV     @TSTSUC,R5              ;SET UP TO DO A SET UNIT CHAR
5449 040502  004737  021042                        JSR     PC,SCHED                ;GO ISSUE THE COMMAND
5450
5451 040506  012705  042332                        MOV     @T1ONL,R5               ;SET UP TO DO AN ONLINE
5452 040512  004737  021042                        JSR     PC,SCHED                ;GO ISSUE THE COMMAND
5453 040516  005737  003704                        TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
5454 040522  001002                              BNE     5$                      ;NO, CONTINUE
5455 040524  000137  042306                      JMP     T1EXIT                  ;GET OUT IF NONE LEFT TO TEST
5456
5457 040530  005037  003616                5$:     CLR     TSTMSK                  ;ALLOW NO ERRORS
5458 040534  012705  042340                        MOV     @T1REW,R5               ;SET UP TO DO A REWIND
5459 040540  004737  021042                        JSR     PC,SCHED                ;GO ISSUE THE COMMAND
5460 040544  005737  003704                        TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
5461 040550  C01002                              BNE     10$                     ;NO, CONTINUE
5462 040552  000137  042306                      JMP     T1EXIT                  ;GET OUT IF NONE LEFT TO TEST
5463
5464 040556  012705  042346                10$:    MOV     @T1LEOT,R5              ;SET UP TO DO 2 TAPE MARK COMMANDS
5465 040562  004737  021042                        JSR     PC,SCHED                ;GO ISSUE THE COMMAND
5466 040566  005737  003704                        TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
5467 040572  001002                              BNE     15$                     ;NO, CONTINUE
5468 040574  000137  042306                      JMP     T1EXIT                  ;GET OUT IF NONE LEFT TO TEST
5469
5470 040600  012705  042370                15$:    MOV     @T1SKR,R5               ;SET UP TO SKIP REVERSE 2 TAPE MARKS
5471 040604  004737  021042                        JSR     PC,SCHED                ;GO ISSUE THE COMMAND
5472 040610  005737  003704                        TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
5473 040614  001002                              BNE     20$                     ;NO, CONTINUE
5474 040616  000137  042306                      JMP     T1EXIT                  ;GET OUT IF NONE LEFT TO TEST
5475
5476 040622  052737  000001  003616    20$:    BIS     #LEDB,TSTMSK            ;SET UP TO ALLOW LEOT DETECTED
5477 040630  012705  042500                        MOV     @T1SKD,R5               ;SET UP TO DO A SPACE TO LEOT
5478 040634  004737  021042                        JSR     PC,SCHED                ;GO ISSUE THE COMMAND
5479 040640  042737  000001  003616              BIC     #LEDB,TSTMSK            ;DISALLOW LEOT DETECTED
5480 040646  005737  003704                        TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
5481 040652  001002                              BNE     25$                     ;NO, CONTINUE
5482 040654  000137  042306                      JMP     T1EXIT                  ;GET OUT IF NONE LEFT TO TEST
5483
5484 040660  012705  042340                25$:    MOV     @T1REW,R5               ;SET UP TO DO A REWIND
5485 040664  004737  021042                        JSR     PC,SCHED                ;GO ISSUE THE COMMAND
5486 040670  005737  003704                        TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
5487 040674  001002                              BNE     30$                     ;NO, CONTINUE
5488 040676  000137  042306                      JMP     T1EXIT                  ;GET OUT IF NONE LEFT TO TEST
5489
5490 040702  012705  042506                30$:    MOV     @T1WR1,R5               ;WRITE 99, 512 BYTE RECORDS
5491 040706  004737  033636                        JSR     PC,SDSTUP               ;RESET THE RANDOM SEEDS
5492 040712  004737  021042                        JSR     PC,SCHED                ;GO ISSUE THE COMMAND
5493 040716  005737  003704                        TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
5494 040722  001002                              BNE     35$                     ;NO, CONTINUE
5495 040724  000137  042306                      JMP     T1EXIT                  ;GET OUT IF NONE LEFT TO TEST
5496
5497 040730  012705  042354                35$:    MOV     @T1WTM,R5               ;SET UP TO WRITE A TAPE MARK
5498 040734  004737  021042                        JSR     PC,SCHED                ;GO ISSUE THE COMMAND
```

```
5499 040740  005737  00370-          TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5500 040744  001002                  BNE     40$             ;NO, CONTINUE
5501 040746  000137  042306          JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5502
5503 040752  012705  042514  40$:    MOV     #T1WR2,R5       ;WRITE 84, 525 BYTE RECORDS
5504 040756  004737  033636          JSR     PC,SDSTUP       ;RESET THE RANDOM SEEDS
5505 040762  004737  021042          JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5506 040766  005737  003704          TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5507 040772  001002                  BNE     45$             ;NO, CONTINUE
5508 040774  000137  042306          JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5509
5510 041000  012705  042354  45$:    MOV     #T1WTM,R5       ;SET UP TO WRITE A TAPE MARK
5511 041004  004737  021042          JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5512 041010  005737  003704          TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5513 041014  001002                  BNE     50$             ;NO, CONTINUE
5514 041016  000137  042306          JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5515
5516 041022  012705  042522  50$:    MOV     #T1WR3,R5       ;WRITE 69, 1038 BYTE RECORDS
5517 041026  004737  033636          JSR     PC,SDSTUP       ;RESET THE RANDOM SEEDS
5518 041032  004737  021042          JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5519 041036  005737  003704          TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5520 041042  001002                  BNE     55$             ;NO, CONTINUE
5521 041044  000137  042306          JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5522
5523 041050  012705  042354  55$:    MOV     #T1WTM,R5       ;SET UP TO WRITE A TAPE MARK
5524 041054  004737  021042          JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5525 041060  005737  003704          TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5526 041064  001002                  BNE     60$             ;NO, CONTINUE
5527 041066  000137  042306          JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5528
5529 041072  012705  042530  60$:    MOV     #T1WR4,R5       ;WRITE 54, 1551 BYTE RECORDS
5530 041076  004737  033636          JSR     PC,SDSTUP       ;RESET THE RANDOM SEEDS
5531 041102  004737  021042          JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5532 041106  005737  003704          TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5533 041112  001002                  BNE     65$             ;NO, CONTINUE
5534 041114  000137  042306          JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5535
5536 041120  012705  042354  65$:    MOV     #T1WTM,R5       ;SET UP TO WRITE A TAPE MARK
5537 041124  004737  021042          JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5538 041130  005737  003704          TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5539 041134  001002                  BNE     70$             ;NO, CONTINUE
5540 041136  000137  042306          JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5541
5542 041142  012705  042536  70$:    MOV     #T1WR5,R5       ;WRITE 39, 2064 BYTE RECORDS
5543 041146  004737  033636          JSR     PC,SDSTUP       ;RESET THE RANDOM SEEDS
5544 041152  004737  021042          JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5545 041156  005737  003704          TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5546 041162  001002                  BNE     75$             ;NO, CONTINUE
5547 041164  000137  042306          JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5548
5549 041170  012705  042354  75$:    MOV     #T1WTM,R5       ;SET UP TO WRITE A TAPE MARK
5550 041174  004737  021042          JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5551 041200  005737  003704          TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5552 041204  001002                  BNE     80$             ;NO, CONTINUE
5553 041206  000137  042306          JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5554
5555 041212  012705  042544  80$:    MOV     #T1WR6,R5       ;WRITE 24, 2577 BYTE RECORDS
```

```
5556 041216  004737  033636              JSR     PC,SDSTUP       ;RESET THE RANDOM SEEDS
5557 041222  004737  021042              JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5558 041226  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5559 041232  001002                      BNE     85$             ;NO, CONTINUE
5560 041234  000137  042306              JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5561
5562 041240  012705  042346    85$:      MOV     #T1LEOT,R5      ;SET UP TO WRITE LOGICAL END OF TAPE
5563 041244  004737  021042              JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5564 041250  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5565 041254  001002                      BNE     90$             ;NO, CONTINUE
5566 041256  000137  042306              JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5567
5568 041262  012705  042340    90$:      MOV     #T1REW,R5       ;SET UP TO REWIND
5569 041266  004737  021042              JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5570 041272  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5571 041276  001002                      BNE     95$             ;NO, CONTINUE
5572 041300  000137  042306              JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5573
5574 041304  012705  042552    95$:      MOV     #T1RD1,R5       ;SET UP TO READ 100 RECORDS
5575 041310  C04737  033636              JSR     PC,SDSTUP       ;RESET THE RANDOM SEEDS
5576 041314  004737  021042              JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5577 041320  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5578 041324  001002                      BNE     100$            ;NO, CONTINUE
5579 041326  000137  042306              JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5580
5581 041332  012705  042362   100$:      MOV     #T1SKP,R5       ;SET UP TO SKIP A TAPE MARK
5582 041336  004737  021042              JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5583 041342  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5584 041346  001002                      BNE     105$            ;NO, CONTINUE
5585 041350  000137  042306              JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5586
5587 041354  012705  042376   105$:      MOV     #T1SPC1,R5      ;SET UP TO SPACE 84 RECORDS
5588 041360  004737  021042              JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5589 041364  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5590 041370  001002                      BNE     110$            ;NO, CONTINUE
5591 041372  000137  042306              JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5592
5593 041376  012705  042362   110$:      MOV     #T1SKP,R5       ;SET UP TO SKIP A TAPE MARK
5594 041402  004737  021042              JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5595 041406  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5596 041412  001002                      BNE     115$            ;NO, CONTINUE
5597 041414  000137  042306              JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5598
5599 041420  012705  042404   115$:      MOV     #T1SPC2,R5      ;SET UP TO SPACE 69 RECORDS
5600 041424  004737  021042              JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5601 041430  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5602 041434  001002                      BNE     120$            ;NO, CONTINUE
5603 041436  000137  042306              JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5604
5605 041442  012705  042420   120$:      MOV     #T1SP01,R5      ;SET UP TO SPACE 56 OBJECTS
5606 041446  004737  021042              JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5607 041452  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5608 041456  001002                      BNE     125$            ;NO, CONTINUE
5609 041460  000137  042306              JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5610
5611 041464  012705  042602   125$:      MOV     #T1RD5,R5       ;SET UP TO READ 39 RECORDS
5612 041470  004737  033636              JSR     PC,SDSTUP       ;RESET THE RANDOM SEEDS
```

```
5613 041474  004737  021042              JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5614 041500  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5615 041504  001002                      BNE     130$            ;NO, CONTINUE
5616 041506  000137  042306              JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5617
5618 041512  012705  042426    130$:     MOV     #T1SKR1,R5      ;SET UP TO SKIP REVERSE 4 TAPE MARKS
5619 041516  004737  021042              JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5620 041522  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5621 041526  001002                      BNE     135$            ;NO, CONTINUE
5622 041530  000137  042306              JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5623
5624 041534  012705  042362    135$:     MOV     #T1SKP,R5       ;SET UP TO SKIP TAPE MARK FORWARD
5625 041540  004737  021042              JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5626 041544  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5627 041550  001002                      BNE     140$            ;NO, CONTINUE
5628 041552  000137  042306              JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5629
5630 041556  012705  042560    140$:     MOV     #T1RD2,R5       ;SET UP TO READ 84 RECORDS
5631 041562  004737  033636              JSR     PC,SDSTUP       ;RESET THE RANDOM SEEDS
5632 041566  004737  021042              JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5633 041572  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5634 041576  001002                      BNE     145$            ;NO, CONTINUE
5635 041600  000137  042306              JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5636
5637 041604  012705  042434    145$:     MOV     #T1SP02,R5      ;SET UP TO SPACE 71 OBJECTS
5638 041610  004737  021042              JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5639 041614  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5640 041620  001002                      BNE     150$            ;NO, CONTINUE
5641 041622  000137  042306              JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5642
5643 041626  012705  042574    150$:     MOV     #T1RD4,R5       ;SET UP TO READ 54 RECORDS
5644 041632  004737  033636              JSR     PC,SDSTUP       ;RESET THE RANDOM SEEDS
5645 041636  004737  021042              JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5646 041642  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5647 041646  001002                      BNE     155$            ;NO, CONTINUE
5648 041650  000137  042306              JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5649
5650 041654  012705  042442    155$:     MOV     #T1SP03,R5      ;SET UP TO SPACE 66 OBJECTS
5651 041660  004737  021042              JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5652 041664  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5653 041670  001002                      BNE     160$            ;NO, CONTINUE
5654 041672  000137  042306              JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5655
5656 041676  012705  042450    160$:     MOV     #T1SPR1,R5      ;SET UP TO SPACE REVERSE 375 OBJECTS
5657 041702  004737  021042              JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5658 041706  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5659 041712  001002                      BNE     165$            ;NO, CONTINUE
5660 041714  000137  042306              JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5661
5662 041720  012705  042456    165$:     MOV     #T1SKP1,R5      ;SET UP TO SKIP FORWARD 4 TAPE MARKS
5663 041724  004737  021042              JSR     PC,SCHED        ;GO ISSUE THE COMMAND
5664 041730  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
5665 041734  001002                      BNE     170$            ;NO, CONTINUE
5666 041736  000137  042306              JMP     T1EXIT          ;GET OUT IF NONE LEFT TO TEST
5667
5668 041742  012705  042412    170$:     MOV     #T1SPC3,R5      ;SET UP TO SPACE 39 RECORDS
5669 041746  004737  021042              JSR     PC,SCHED        ;GO ISSUE THE COMMAND
```

```
5670 041752 005737 003704          TST   UDROP      ;HAVE ALL UNITS BEEN DROPPED ?
5671 041756 001002                 BNE   175$       ;NO, CONTINUE
5672 041760 000137 042306          JMP   T1EXIT     ;GET OUT IF NONE LEFT TO TEST
5673
5674 041764 012705 042362  175$:   MOV   #T1SKP,R5  ;SET UP TO SKIP A TAPE MARK
5675 041770 004737 021042          JSR   PC,SCHED   ;GO ISSUE THE COMMAND
5676 041774 005737 003704          TST   UDROP      ;HAVE ALL UNITS BEEN DROPPED ?
5677 042000 001002                 BNE   180$       ;NO, CONTINUE
5678 042002 000137 042306          JMP   T1EXIT     ;GET OUT IF NONE LEFT TO TEST
5679
5680 042006 012705 042610  180$:   MOV   #T1RD6,R5  ;SET UP TO READ 824 RECORDS
5681 042012 004737 033634          JSR   PC,SDSTUP  ;RESET THE RANDOM SEEDS
5682 042016 004737 021042          JSR   PC,SCHED   ;GO ISSUE THE COMMAND
5683 042022 005737 003704          TST   UDROP      ;HAVE ALL UNITS BEEN DROPPED ?
5684 042026 001002                 BNE   185$       ;NO, CONTINUE
5685 042030 000137 042306          JMP   T1EXIT     ;GET OUT IF NONE LEFT TO TEST
5686
5687 042034 012705 042464  185$:   MOV   #T1SKP2,R5 ;SET UP TO SKIP 2 TAPE MARKS
5688 042040 004737 021042          JSR   PC,SCHED   ;GO ISSUE THE COMMAND
5689 042044 005737 003704          TST   UDROP      ;HAVE ALL UNITS BEEN DROPPED ?
5690 042050 001002                 BNE   190$       ;NO, CONTINUE
5691 042052 000137 042306          JMP   T1EXIT     ;GET OUT IF NONE LEFT TO TEST
5692
5693 042056 012705 042472  190$:   MOV   #T1SPR2,R5 ;SET UP TO SPACE REVERSE 192 OBJECTS
5694 042062 004737 021042          JSR   PC,SCHED   ;GO ISSUE THE COMMAND
5695 042066 005737 003704          TST   UDROP      ;HAVE ALL UNITS BEEN DROPPED ?
5696 042072 001002                 BNE   195$       ;NO, CONTINUE
5697 042074 000137 042306          JMP   T1EXIT     ;GET OUT IF NONE LEFT TO TEST
5698
5699 042100 012705 042362  195$:   MOV   #T1SKP,R5  ;SET UP TO SKIP A TAPE MARK
5700 042104 004737 021042          JSR   PC,SCHED   ;GO ISSUE THE COMMAND
5701 042110 005737 003704          TST   UDROP      ;HAVE ALL UNITS BEEN DROPPED ?
5702 042114 001002                 BNE   200$       ;NO, CONTINUE
5703 042116 000137 042306          JMP   T1EXIT     ;GET OUT IF NONE LEFT TO TEST
5704
5705 042122 012705 042566  200$:   MOV   #T1RD3,R5  ;SET UP TO READ 69 RECORDS
5706 042126 004737 033636          JSR   PC,SDSTUP  ;RESET THE RANDOM SEEDS
5707 042132 004737 021042          JSR   PC,SCHED   ;GO ISSUE THE COMMAND
5708 042136 005737 003704          TST   UDROP      ;HAVE ALL UNITS BEEN DROPPED ?
5709 042142 001002                 BNE   205$       ;NO, CONTINUE
5710 042144 000137 042306          JMP   T1EXIT     ;GET OUT IF NONE LEFT TO TEST
5711
5712 042150 012705 042340  205$:   MOV   #T1REW,R5  ;SET UP TO REWIND
5713 042154 004737 021042          JSR   PC,SCHED   ;GO ISSUE THE COMMAND
5714 042160 005737 003704          TST   UDROP      ;HAVE ALL UNITS BEEN DROPPED ?
5715 042164 001002                 BNE   210$       ;NO, CONTINUE
5716 042166 000137 042306          JMP   T1EXIT     ;GET OUT IF NONE LEFT TO TEST
5717
5718 042172 052737 000001 003616 210$:   BIS   #LEDB,TSTMSK  ;SET UP TO ALLOW LEOT DETECTED
5719 042200 012705 042500          MOV   #T1SKD,R5  ;SET UP TO SKIP TO LEOT
5720 042204 004737 021042          JSR   PC,SCHED   ;GO ISSUE THE COMMAND
5721 042210 042737 000001 003616   BIC   #LEDB,TSTMSK  ;DISALLOW LEOT DETECTED
5722 042216 005737 003704          TST   UDROP      ;HAVE ALL UNITS BEEN DROPPED ?
5723 042222 001002                 BNE   215$       ;NO, CONTINUE
5724 042224 000137 042306          JMP   T1EXIT     ;GET OUT IF NONE LEFT TO TEST
5725
5726 042230 012705 042340  215$:   MOV   #T1REW,R5  ;SET UP TO REWIND
```

TEST 1: Basic Function Test

```
 5727 042234  004737  021042                JSR     PC,SCHED            ;GO ISSUE THE COMMAND
 5728
 5729 042240  005737  003702        220$:   TST     PASS1               ;IS THIS THE END OF 1ST PASS ?
 5730 042244  001020                BNE     T1EXIT              ;BRANCH IF NOT
 5731 042246  005337  003702                DEC     PASS1               ;PASS1 = 1
 5732 042252  022737  000002  003732        CMP     @TF.PE,FORMAT       ;WAS 1ST PASS DONE IN PE ?
 5733 042260  001405                BEQ     225$                ;BRANCH IF SO
 5734 042262  012737  000002  003732        MOV     @TF.PE,FORMAT       ;SET TAPE FORMAT TO PE
 5735 042270  000137  040412                JMP     GO1                 ;REPEAT TEST. THIS TIME IN PE
 5736
 5737 042274  012737  000004  003732  225$:   MOV     @TF.GCR,FORMAT      ;SET TAPE FORMAT TO GCR
 5738 042302  000137  040412                JMP     GO1                 ;REPEAT TEST. THIS TIME IN GCR
 5739
 5740 042306  005037  003702        T1EXIT: CLR     PASS1               ;RESET  1ST PASS" FLAG FOR NEXT TEST
 5741 042312  013737  003734  003732        MOV     INFORM,FORMAT       ;RESTORE INITIAL TEST FORMAT
 5742 042320                                EXIT    TST
      042320  104432                        TRAP    C$EXIT
      042322  000302                        .WORD   L10015 .
```

```
5744
5745 042324     150                TSTSUC: .BYTE   SUC             ;SET UNIT CHARACTERISTICS  COMMAND
5746 042325     000                        .BYTE   NULPAT          ;NO DATA NEEDED
5747 042326  000000                        .WORD   0               ;NO ITEM COUNT
5748 042330  000001                        .WORD   1               ;DO IT ONE TIME
5749
5750 042332     140                T1ONL:  .BYTE   ONL             ;ONLINE COMMAND
5751 042333     000                        .BYTE   NULPAT          ;NO DATA NEEDED
5752 042334  000000                        .WORD   0               ;NO ITEM COUNT
5753 042336  000001                        .WORD   1               ;DO IT ONE TIME
5754
5755 042340     160                T1REW:  .BYTE   REW             ;REWIND COMMAND
5756 042341     000                        .BYTE   NULPAT          ;NO DATA NEEDED
5757 042342  000000                        .WORD   0               ;NO ITEM COUNT
5758 042344  000001                        .WORD   1               ;DO IT ONE TIME
5759
5760 042346     100                T1LEOT: .BYTE   WTM             ;WRITE TAPE MARK
5761 042347     000                        .BYTE   NULPAT          ;NO DATA NEEDED
5762 042350  000000                        .WORD   0               ;NO ITEM COUNT
5763 042352  000002                        .WORD   2               ;DO IT TWICE
5764
5765 042354     100                T1WTM:  .BYTE   WTM             ;WRITE TAPE MARK
5766 042355     000                        .BYTE   NULPAT          ;NO DATA NEEDED
5767 042356  000000                        .WORD   0               ;NO ITEM COUNT
5768 042360  000001                        .WORD   1               ;DO IT ONE TIME
5769
5770 042362     060                T1SKP:  .BYTE   SKP             ;SKIP TAPE MARK
5771 042363     000                        .BYTE   NULPAT          ;NO DATA NEEDED
5772 042364  000001                        .WORD   1               ;SKIP 1 TAPE MARK
5773 042366  000001                        .WORD   1               ;DO IT ONE TIME
5774
5775 042370     061                T1SKR:  .BYTE   SKR             ;SKIP TAPE MARK REVERSE
5776 042371     000                        .BYTE   NULPAT          ;NO DATA NEEDED
5777 042372  000002                        .WORD   2               ;SKIP REVERSE 2 TAPE MARKS
5778 042374  000001                        .WORD   1               ;DO IT ONCE
5779
5780 042376     050                T1SPC1: .BYTE   SPC             ;SPACE RECORDS
5781 042377     000                        .BYTE   NULPAT          ;NO DATA NEEDED
5782 042400  000124                        .WORD   84.             ;SPACE 84 RECORDS
5783 042402  000001                        .WORD   1               ;DO IT ONE TIME
5784
5785 042404     050                T1SPC2: .BYTE   SPC             ;SPACE RECORDS
5786 042405     000                        .BYTE   NULPAT          ;NO DATA NEEDED
5787 042406  000105                        .WORD   69.             ;SPACE 69 RECORDS
5788 042410  000001                        .WORD   1               ;DO IT ONE TIME
5789
5790 042412     050                T1SPC3: .BYTE   SPC             ;SPACE RECORDS
5791 042413     000                        .BYTE   NULPAT          ;NO DATA NEEDED
5792 042414  000047                        .WORD   39.             ;SPACE 39 RECORDS
5793 042416  000001                        .WORD   1               ;DO IT ONE TIME
5794
5795 042420     070                T1SPO1: .BYTE   SPO             ;SPACE OBJECTS
5796 042421     000                         BYTE   NULPAT          ;NO DATA NEEDED
5797 042422  000070                        .WORD   56.             ;SPACE 56 OBJECTS
5798 042424  000001                        .WORD   1               ;DO IT ONE TIME
5799
5800 042426     061                T1SKR1: .BYTE   SKR             ;SKIP TAPE MARK REVERSE
```

```
5801 042427     000                        .BYTE   NULPAT          ;NO DATA NEEDED
5802 042430  000004                        .WORD   4               ;4 TAPE MARKS
5803 042432  000001                        .WORD   1               ;DO IT ONCE
5804
5805 042434     070                T1SP02: .BYTE   SPO             ;SPACE OBJECTS
5806 042435     000                        .BYTE   NULPAT          ;NO DATA NEEDED
5807 042436  000107                        .WORD   71.             ;SPACE 71 OBJECTS
5808 042440  000001                        .WORD   1               ;DO IT ONE TIME
5809
5810 042442     070                T1SP03: .BYTE   SPO             ;SPACE OBJECTS
5811 042443     000                        .BYTE   NULPAT          ;NO DATA NEEDED
5812 042444  000102                        .WORD   66.             ;SPACE 66 OBJECTS
5813 042446  000001                        .WORD   1               ;DO IT ONE TIME
5814
5815 042450     071                T1SPR1: .BYTE   SPR             ;SPACE OBJECTS REVERSE
5816 042451     000                        .BYTE   NULPAT          ;NO DATA NEEDED
5817 042452  000567                        .WORD   375.            ;SPACE 375 OBJECTS
5818 042454  000001                        .WORD   1               ;DO IT ONE TIME
5819
5820 042456     060                T1SKP1: .BYTE   SKP             ;SKIP TAPE MARKS
5821 042457     000                        .BYTE   NULPAT          ;NO DATA NEEDED
5822 042460  000004                        .WORD   4.              ;SKIP 4 TAPE MARKS
5823 042462  000001                        .WORD   1               ;DO IT ONE TIME
5824
5825 042464     060                T1SKP2: .BYTE   SKP             ;SKIP TAPE MARKS
5826 042465     000                        .BYTE   NULPAT          ;NO DATA NEEDED
5827 042466  000002                        .WORD   2.              ;SKIP 2 TAPE MARKS
5828 042470  000001                        .WORD   1               ;DO IT ONE TIME
5829
5830 042472     071                T1SPR2: .BYTE   SPR             ;SPACE OBJECTS REVERSE
5831 042473     000                        .BYTE   NULPAT          ;NO DATA NEEDED
5832 042474  000300                        .WORD   192.            ;SPACE 192 OBJECTS
5833 042476  000001                        .WORD   1               ;DO IT ONE TIME
5834
5835 042500     062                T1SKD:  .BYTE   SKD             ;SKIP TO LEOT
5836 042501     000                        .BYTE   NULPAT          ;NO DATA NEEDED
5837 042502  000004                        .WORD   4               ;NO ITEM COUNT
5838 042504  000001                        .WORD   1               ;DO IT ONE TIME
5839
5840 042506     020                T1WR1:  .BYTE   WR              ;WRITE RECORD
5841 042507     001                        .BYTE   PAT1            ;DATA PATERN 1 (ALL 1'S)
5842 042510  000026                        .WORD   22.             ;BYTE COUNT OF 512.
5843 042512  000143                        .WORD   99.             ;DO IT 99 TIMES
5844
5845 042514     020                T1WR2:  .BYTE   WR              ;WRITE RECORD
5846 042515     002                        .BYTE   PAT2            ;DATA PATERN 2 (ALL 0'S)
5847 042516  001015                        .WORD   525.            ;BYTE COUNT OF 525
5848 042520  000124                        .WORD   84.             ;DO IT 84 TIMES
5849
5850 042522     020                T1WR3:  .BYTE   WR              ;WRITE RECORD
5851 042523     003                        .BYTE   PAT3            ;DATA PATERN 3 (WORST MFM)
5852 042524  002016                        .WORD   1038.           ;BYTE COUNT OF 1038
5853 042526  000105                        .WORD   69.             ;DO IT 69 TIMES
5854
5855 042530     020                T1WR4:  .BYTE   WR              ;WRITE RECORD
5856 042531     004                        .BYTE   PAT4            ;DATA PATERN 4 (ALTERNATE 1'S AND 0'S)
5857 042532  003017                        .WORD   1551.           ;BYTE COUNT OF 1551
```

```
5858 042534  000066                        .WORD   54.                     ;DO IT 54 TIMES
5859
5860 042536     020              T1WR5:    .BYTE   WR                      ;WRITE RECORD
5861 042537     003                        .BYTE   PAT3                    ;DATA PATERN 3 (WORST MFM)
5862 042540  004020                        .WORD   2064.                   ;BYTE COUNT OF 2064
5863 042542  000047                        .WORD   39.                     ;DO IT 39 TIMES
5864
5865 042544     020              T1WR6:    .BYTE   WR                      ;WRITE RECORD
5866 042545     001                        .BYTE   PAT1                    ;DATA PATERN 1 (ALL 1'S)
5867 042546  005021                        .WORD   2577.                   ;BYTE COUNT OF 2577
5868 042550  000030                        .WORD   24.                     ;DO IT 24 TIMES
5869
5870 042552     010              T1RD1:    .BYTE   RD                      ;READ RECORD
5871 042553     001                        .BYTE   PAT1                    ;DATA PATERN 1 (ALL 1'S)
5872 042554  000026                        .WORD   22.                     ;BYTE COUNT OF 512.
5873 042556  000143                        .WORD   99.                     ;DO IT 99 TIMES
5874
5875 042560     010              T1RD2:    .BYTE   RD                      ;READ RECORD
5876 042561     002                        .BYTE   PAT2                    ;DATA PATERN 2 (ALL 0'S)
5877 042562  001015                        .WORD   525.                    ;BYTE COUNT OF 525
5878 042564  000124                        .WORD   84.                     ;DO IT 84 TIMES
5879
5880 042566     010              T1RD3:    .BYTE   RD                      ;READ RECORD
5881 042567     003                        .BYTE   PAT3                    ;DATA PATERN 3 (WORST MFM)
5882 042570  002016                        .WORD   1038.                   ;BYTE COUNT OF 1038
5883 042572  000105                        .WORD   69.                     ;DO IT 69 TIMES
5884
5885 042574     010              T1RD4:    .BYTE   RD                      ;READ RECORD
5886 042575     004                        .BYTE   PAT4                    ;DATA PATERN 4 (ALTERNATE 1'S AND 0'S)
5887 042576  003017                        .WORD   1551.                   ;BYTE COUNT OF 1551
5888 042600  000066                        .WORD   54.                     ;DO IT 54 TIMES
5889
5890 042602     010              T1RD5:    .BYTE   RD                      ;READ RECORD
5891 042603     003                        .BYTE   PAT3                    ;DATA PATERN 3 (WORST MFM)
5892 042604  004020                        .WORD   2064.                   ;BYTE COUNT OF 2064
5893 042606  000047                        .WORD   39.                     ;DO IT 39 TIMES
5894
5895 042610     010              T1RD6:    .BYTE   RD                      ;READ RECORD
5896 042611     001                        .BYTE   PAT1                    ;DATA PATERN 1 (ALL 1'S)
5897 042612  005021                        .WORD   2577.                   ;BYTE COUNT OF 2577
5898 042614  000030                        .WORD   24.                     ;DO IT 24 TIMES
5899
5900 042616     071              RTSPR1:   .BYTE   SPR                     ;SPACE OBJECTS REVERSE
5901 042617     000                        .BYTE   NULPAT                  ;NO DATA NEEDED
5902 042620  000001                        .WORD   1                       ;SPACE 1 OBJECT
5903 042622  000001                        .WORD   1                       ;DO IT ONE TIME
5904                                        .EVEN
5905
5906 042624                                ENDTST
     042624                     L10015:
     042624  104401                        TRAP    C$ETST
```

```
5913                                    .SBTTL  TEST 2: Quick Verify Write/Read Test
5914
5915                                    ;++
5916                                    ;This test rewinds the tape, then executes the following sequence:
5917                                    ;
5918                                    ;       1.  Write record set,
5919                                    ;       2.  Reposition over just written record set,
5920                                    ;       3.  Then read the current record set,
5921                                    ;
5922                                    ;for 5 iterations or until fatal error  is  encountered.   This  test
5923                                    ;permits  retries,  fixed  record  length (2048 bytes), fixed number of
5924                                    ;records/set (400), and predetermined data patterns.   This  test  will
5925                                    ;execute in a round-robin manner.
5926                                    ; -
5927  042626                                    BGNTST
     042626                             T2::
5928
5929  042626  005737  003704            START2: TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
5930  042632  001014                            BNE     5$                      ;GO START THE TEST
5931  042634                                    PRINTF  #BYPASS,L$TEST          ;PRINT THE TEST BYPASSED MESSAGE
      042634  013746  002114                    MOV     L$TEST,-(SP)
      042640  012746  020527                    MOV     #BYPASS, (SP)
      042644  012746  000002                    MOV     #2,-(SP)
      042650  010600                            MOV     SP,R0
      042652  104417                            TRAP    C$PNTF
      042654  062706  000006                    ADD     #6,SP
5932  042660  000137  043356                    JMP     T2EXIT                  ;GET OUT IF NONE LEFT TO TEST
5933
5934  042664  105737  002216            5$:     TSTB    CLOCK                   ;IS THE CLOCK ENABLED
5935  042670  001421                            BEQ     G02                     ;NO, THEN CAN'T PRINT TIME
5936  042672                                    PRINTF  #TIME,<B,HOURS>,<B,MINUTE>,<B,SECOND>,
      042672  005046                            CLR     -(SP)
      042674  153716  002221                    BISB    SECOND,(SP)
      042700  005046                            CLR     -(SP)
      042702  153716  002220                    BISB    MINUTE,(SP)
      042706  005046                            CLR     -(SP)
      042710  153716  002217                    BISB    HOURS,(SP)
      042714  012746  020037                    MOV     #TIME,-(SP)
      042720  012746  000004                    MOV     #4,-(SP)
      042724  010600                            MOV     SP,R0
      042726  104417                            TRAP    C$PNTF
      042730  062706  000012                    ADD     #12,SP
5937
5938  042734  004737  033550            G02:    JSR     PC,CLREOT               ;MAKE SURE EOT STATUS IS CLEAR
5939  042740  012737  000100  003616            MOV     #ONLB,TSTMSK            ;ALLOW ALREADY ONLINE STATUS
5940
5941  042746  022737  000002  003732            CMP     #TF.PE,FORMAT           ;ARE WE DOING PE ?
5942  042754  001011                            BNE     1$                      ;NO, PRINT GCR
5943  042756                                    PRINTF  #TSTPE                  ;PRINT TESTING IN PE
      042756  012746  020610                    MOV     #TSTPE,-(SP)
      042762  012746  000001                    MOV     #1,-(SP)
      042766  010600                            MOV     SP,R0
      042770  104417                            TRAP    C$PNTF
      042772  062706  000004                    ADD     #4,SP
5944  042776  000410                            BR      3$                      ;START TEST
5945  043000                            1$:     PRINTF  #TSTGCR                 ;PRINT TESTING IN GCR
      043000  012746  020562                    MOV     #TSTGCR,-(SP)
```

TEST 2. Quick Verify Write/Read Test

```
                043004  012746  000001          MOV     #1,-(SP)
                043010  010600                  MOV     SP,R0
                043012  104417                  TRAP    C$PNTF
                043014  062706  000004          ADD     #4,SP
   5946         043020  012705  042324  3$:     MOV     #TSTSUC,R5          ;SET UP TO DO A SET UNIT CHAR
   5947         043024  004737  021042          JSR     PC,SCHED            ;GO ISSUE THE COMMAND
   5948         043030  005737  003704          TST     UDROP               ;HAVE ALL UNITS BEEN DROPPED ?
   5949         043034  001002                  BNE     4$                  ;NO, CONTINUE
   5950         043036  000137  043356          JMP     T2EXIT              ;GET OUT IF NONE LEFT TO TEST
   5951
   5952         043042  005037  003676  4$:     CLR     OBJECT              ;CLEAR THE OBJECT COUNTER
   5953         043046  005037  003616          CLR     TSTMSK              ;ALLOW NO ERRORS
   5954         043052  012705  043400          MOV     #T2REW,R5           ;SET UP TO DO A REWIND
   5955         043056  004737  021042          JSR     PC,SCHED            ;GO ISSSUE A REWIND TO ALL DRIVES
   5956         043062  005737  003704          TST     UDROP               ;HAVE ALL UNITS BEEN DROPPED ?
   5957         043066  001533                  BEQ     T2EXIT              ;GET OUT IF NONE LEFT TO TEST
   5958
   5959
   5960         043070  004737  033636  5$:     JSR     PC,SDSTUP           ;RESET THE RANDOM SEEDS
   5961         043074  012705  043406          MOV     #T2WRT,R5           ;SET UP TO DO A WRITE ITERATION
   5962         043100  004737  021042          JSR     PC,SCHED            ;GO ISSSUE WRITES TO ALL DRIVES
   5963         043104  005737  003704          TST     UDROP               ;HAVE ALL UNITS BEEN DROPPED ?
   5964         043110  001522                  BEQ     T2EXIT              ;GET OUT IF NONE LEFT TO TEST
   5965
   5966         043112  012705  043414          MOV     #T2LEOT,R5          ;SET UP TO DO A WRITE LEOT
   5967         043116  004737  021042          JSR     PC,SCHED            ;GO ISSSUE WRITES TO ALL DRIVES
   5968         043122  005737  003704          TST     UDROP               ;HAVE ALL UNITS BEEN DROPPED ?
   5969         043126  001513                  BEQ     T2EXIT              ;GET OUT IF NONE LEFT TO TEST
   5970
   5971         043130  012705  043400          MOV     #T2REW,R5           ;SET UP TO DO A REWIND
   5972         043134  004737  021042          JSR     PC,SCHED            ;GO ISSSUE A REWIND TO ALL DRIVES
   5973         043140  005737  003704          TST     UDROP               ;HAVE ALL UNITS BEEN DROPPED ?
   5974         043144  001504                  BEQ     T2EXIT              ;GET OUT IF NONE LEFT TO TEST
   5975
   5976         043146  005737  003676          TST     OBJECT              ;IS THIS THE FIRST TIME THROUGH ?
   5977         043152  001412                  BEQ     10$                 ;YES, DON'T DO THE SPACE FORWARD
   5978         043154  012705  043436          MOV     #T2SPO,R5           ;SET UP TO SPACE OBJECTS
   5979         043160  013765  003676  000002  MOV     OBJECTS,ITMCNT(R5)  ;SET UP # OF OBJECTS TO SPACE FORWARD
   5980         043166  004737  021042          JSR     PC,SCHED            ;GO ISSSUE A REWIND TO ALL DRIVES
   5981         043172  005737  003704          TST     UDROP               ;HAVE ALL UNITS BEEN DROPPED ?
   5982         043176  001467                  BEQ     T2EXIT              ;GET OUT IF NONE LEFT TO TEST
   5983
   5984         043200  004737  033636  10$:    JSR     PC,SDSTUP           ;RESET THE RANDOM SEEDS
   5985         043204  012705  043422          MOV     #T2RD,R5            ;SET UP TO DO A READITERATION
   5986         043210  004737  021042          JSR     PC,SCHED            ;GO ISSSUE READS TO ALL DRIVES
   5987         043214  005737  003704          TST     UDROP               ;HAVE ALL UNITS BEEN DROPPED ?
   5988         043220  001456                  BEQ     T2EXIT              ;GET OUT IF NONE LEFT TO TEST
   5989         043222  066537  000004  003676  ADD     ITRCNT(R5),OBJECTS  ;ADD THE # OF RECORDS TO OBJECTS
   5990
   5991         043230  052737  000001  003616  BIS     #LEDB,TSTMSK        ;SET UP TO ALLOW LEOT DETECTED
   5992         043236  012705  043430          MOV     #T2SKD,R5           ;SET UP TO DO A SKIP TO LEOT
   5993         043242  004737  021042          JSR     PC,SCHED            ;GO ISSSUE READS TO ALL DRIVES
   5994         043246  042737  000001  003616  BIC     #LEDB,TSTMSK        ;DISALLOW LEOT DETECTED
   5995         043254  005737  003704          TST     UDROP               ;HAVE ALL UNITS BEEN DROPPED ?
   5996         043260  001436                  BEQ     T2EXIT              ;GET OUT IF NONE LEFT TO TEST
   5997         043262  066537  000004  003676  ADD     ITRCNT(R5),OBJECTS  ;ADD THE # OF RECORDS TO OBJECTS
   5998
```

```
 5999 043270  022737  004716  003676          CMP     @T2END,OBJECTS          ;HAVE WE DONE 2 TRACKS ?
 6000 043276  001274                          BNE     5$                      ;NO, KEEP GOING
 6001 043300  012705  043400                  MOV     @T2REW,R5               ;SET UP TO DO A REWIND
 6002 043304  004737  021042                  JSR     PC,SCHED                ;GO ISSSUE A REWIND TO ALL DRIVES
 6003
 6004 043310  005737  003702          20$:    TST     PASS1                   ;IS THIS THE END OF 1ST PASS ?
 6005 043314  001020                          BNE     T2EXIT                  ;BRANCH IF NOT
 6006 043316  005337  003702                  DEC     PASS1                   ;PASS1 = -1
 6007 043322  022737  000002  003732          CMP     @TF.PE,FORMAT           ;WAS 1ST PASS DONE IN PE ?
 6008 043330  001405                          BEQ     25$                     ;BRANCH IF SO
 6009 043332  012737  000002  003732          MOV     @TF.PE,FORMAT           ;SET TAPE FORMAT TO PE
 6010 043340  000137  042734                  JMP     GO2                     ;REPEAT TEST. THIS TIME IN PE
 6011
 6012 043344  012737  000004  003732  25$:    MOV     @TF.GCR,FORMAT          ;SET TAPE FORMAT TO GCR
 6013 043352  000137  042734                  JMP     GO2                     ;REPEAT TEST. THIS TIME IN GCR
 6014
 6015 043356  005037  003702          T2EXIT: CLR     PASS1                   ;RESET "1ST PASS" FLAG FOR NEXT TEST
 6016 043362  013737  003734  003732          MOV     INFORM,FORMAT           ;RESTORE INITIAL TEST FORMAT
 6017 043370  004737  033714                  JSR     PC,SDSAVE               ;RESET THE RANDOM SEEDS
 6018 043374                                  EXIT    TST
      043374  104432                          TRAP    C$EXIT
      043376  000046                          .WORD   L10016-.
 6019
 6020
 6021 043400     160                  T2REW:  .BYTE   REW                     ;REWIND
 6022 043401     000                          .BYTE   NULPAT
 6023 043402  000000                          .WORD   0
 6024 043404  000001                          .WORD   1
 6025
 6026 043406     020                  T2WRT:  .BYTE   WR                      ;WRITE RECORDS
 6027 043407     003                          .BYTE   PAT3
 6028 043410  010000                          .WORD   4096.
 6029 043412  000372                          .WORD   250.
 6030
 6031 043414     100                  T2LEOT: .BYTE   WTM                     ;WRITE TAPE MARK
 6032 043415     000                          .BYTE   NULPAT                  ;NO DATA NEEDED
 6033 043416  000000                          .WORD   0                       ;NO ITEM COUNT
 6034 043420  000002                          .WORD   2                       ;DO IT TWICE
 6035
 6036 043422     010                  T2RD:   .BYTE   RD                      ;READ RECORDS
 6037 043423     003                          .BYTE   PAT3
 6038 043424  010000                          .WORD   4096.
 6039 043426  000372                          .WORD   250.
 6040
 6041 043430     062                  T2SKD:  .BYTE   SKD                     ;SKIP TAPE MARK TO LEOT
 6042 043431     000                          .BYTE   NULPAT                  ;NO DATA NEEDED
 6043 043432  000062                          .WORD   50.                     ;SKIP 50 TAPE MARKS
 6044 043434  000001                          .WORD   1                       ;DO IT ONE TIME
 6045
 6046 043436     070                  T2SPO:  .BYTE   SPO                     ;SPACE OBJECTS
 6047 043437     000                          .BYTE   NULPAT
 6048 043440  000001                          .WORD   1
 6049 043442  000001                          .WORD   1
 6050
 6051                                         .EVEN
 6052 043444                                  ENDTST
      043444                          L10016:
```

TEST 2: Quick Verify Write/Read Test

```
        043444  104401                          TRAP    C$ETST
   6053
```

```
 6055                                   .SBTTL  TEST 3: Complex Write/Read Test
 6056
 6057                                   ;++
 6058                                   ;This test rewinds the tape, and executes the following sequence:
 6059                                   ;
 6060                                   ;     1.  Write 1000 records,
 6061                                   ;     2.  Write a file mark,
 6062                                   ;     3.  Repeat 1 and 2 until EOT is reached,
 6063                                   ;     4.  Write 2 file marks (LEOT),
 6064                                   ;     5.  Rewind,
 6065                                   ;     6.  Read 1000 records,
 6066                                   ;     7.  Read 1 record (should see unexpected tape mark)
 6067                                   ;     8.  Repeat 6 and 7 until LEOT.
 6068                                   ;
 6069                                   ;# of records (N), and record size will  be  randomly  selected.   This
 6070                                   ;sequence  will  permit  hardware  retries, if not user disabled.  This
 6071                                   ;test will run until EOT, LEOT or fatal error is  detected.   All  data
 6072                                   ;patterns including random data will be used in this test.
 6073                                   ;--
 6074 043446                                   BGNTST
      043446                           T3::
 6075
 6076 043446  005737  003704           START3: TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
 6077 043452  001014                           BNE     5$                      ;GO START THE TEST
 6078 043454                                   PRINTF  #BYPASS,L$TEST          ;PRINT THE TEST BYPASSED MESSAGE
      043454  013746  002114                   MOV     L$TEST,-(SP)
      043460  012746  020527                   MOV     #BYPASS,-(SP)
      043464  012746  000002                   MOV     #2,-(SP)
      043470  010600                           MOV     SP,R0
      043472  104417                           TRAP    C$PNTF
      043474  062706  000006                   ADD     #6,SP
 6079 043500  000137  044224                   JMP     T3EXIT                  ;GET OUT IF NONE LEFT TO TEST
 6080
 6081 043504  105737  002216           5$:     TSTB    CLOCK                   ;IS THE CLOCK ENABLED
 6082 043510  001421                           BEQ     GO3                     ;NO, THEN CAN'T PRINT TIME
 6083 043512                                   PRINTF  #TIME,<B,HOURS>,<B,MINUTE>,<B,SECOND>,
      043512  005046                           CLR     -(SP)
      043514  153716  002221                   BISB    SECOND,(SP)
      043520  005046                           CLR     -(SP)
      043522  153716  002220                   BISB    MINUTE,(SP)
      043526  005046                           CLR     -(SP)
      043530  153716  002217                   BISB    HOURS,(SP)
      043534  012746  020037                   MOV     #TIME,-(SP)
      043540  012746  000004                   MOV     #4,-(SP)
      043544  010600                           MOV     SP,R0
      043546  104417                           TRAP    C$PNTF
      043550  062706  000012                   ADD     #12,SP
 6084
 6085 043554  004737  053550           GO3:    JSR     PC,CLREOT               ;MAKE SURE EOT STATUS IS CLEAR
 6086 043560  012737  000100  003616           MOV     #ONLB,TSTMSK            ;ALLOW ALREADY ONLINE STATUS
 6087
 6088 043566  022737  000002  003732           CMP     #TF.PE,FORMAT           ;ARE WE DOING PE ?
 6089 043574  001011                           BNE     1$                      ;NO, PRINT GCR
 6090 043576                                   PRINTF  #TSTPE                  ;PRINT TESTING IN PE
      043576  012746  020610                   MOV     #TSTPE,-(SP)
      043602  012746  000001                   MOV     #1,-(SP)
      043606  010600                           MOV     SP,R0
```

```
        043610  104417                          TRAP    C$PNTF
        043612  062706  000004                  ADD     #4,SP
   6091 043616  000410                          BR      3$                      ;START TEST
   6092 043620                          1$:     PRINTF  #TSTGCR                 ;PRINT TESTING IN GCR
        043620  012746  020562                  MOV     #TSTGCR,-(SP)
        043624  012746  000001                  MOV     #1,-(SP)
        043630  010600                          MOV     SP,R0
        043632  104417                          TRAP    C$PNTF
        043634  062706  000004                  ADD     #4,SP
   6093 043640  012705  042324          3$:     MOV     #TSTSUC,R5              ;SET UP TO DO A SET UNIT CHAR
   6094 043644  004737  021042                  JSR     PC,SCHED                ;GO ISSUE THE COMMAND
   6095 043650  005737  003704                  TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
   6096 043654  001002                          BNE     4$                      ;NO, CONTINUE
   6097 043656  000137  044224                  JMP     T3EXIT                  ;GET OUT IF NONE LEFT TO TEST
   6098
   6099 043662  004737  033772          4$:     JSR     PC,PATCLR               ;MAKE SURE WE START WITH PATTERN 1
   6100 043666  004737  033636                  JSR     PC,SDSTUP               ;RESET THE RANDOM SEEDS
   6101
   6102 043672  005037  003616                  CLR     TSTMSK                  ;ALLOW NO ERRORS
   6103 043676  005737  003704                  TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
   6104 043702  001550                          BEQ     T3EXIT                  ;GET OUT IF NONE LEFT TO TEST
   6105
   6106 043704  012705  044246                  MOV     #T3REW,R5               ;SET UP TO DO REWIND
   6107 043710  004737  021042                  JSR     PC,SCHED                ;GO ISSUE IO ALL DRIVES
   6108 043714  005737  003704                  TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
   6109 043720  001541                          BEQ     T3EXIT                  ;GET OUT IF NONE LEFT TO TEST
   6110
   6111 043722  012705  044254          5$:     MOV     #T3WRT,R5               ;SET UP A WRITE ITERATION
   6112 043726  004737  021042                  JSR     PC,SCHED                ;GO DO IT ON ALL DRIVES
   6113 043732  005737  003704                  TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
   6114 043736  001532                          BEQ     T3EXIT                  ;GET OUT IF NONE LEFT TO TEST
   6115 043740  023737  003706  003704          CMP     UEOT,UDROP              ;ARE ALL UNITS AT EOT ?
   6116 043746  001413                          BEQ     10$                     ;YES, WRITE ONE MORE REC AND LEOT
   6117
   6118 043750  012705  044262                  MOV     #T3WTM,R5               ;SET UP TO WRITE A TAPE MARK
   6119 043754  004737  021042                  JSR     PC,SCHED                ;GO DO IT ON ALL DRIVES
   6120 043760  005737  003704                  TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
   6121 043764  001517                          BEQ     T3EXIT                  ;GET OUT IF NONE LEFT TO TEST
   6122 043766  023737  003706  003704          CMP     UEOT,UDROP              ;ARE ALL UNITS AT EOT ?
   6123 043774  001352                          BNE     5$                      ;NO, KEEP WRITING
   6124
   6125 043776  012737  000004  010744  10$:     MOV     #4,LOOPS                ;SET UP TO DO 4 TAPE MARKS
   6126 044004  004737  033550          15$:    JSR     PC,CLREOT               ;CLEAR THE EOT INDICATORS
   6127 044010  012705  044262                  MOV     #T3WTM,R5               ;SET UP TO WRITE A TAPE MARK
   6128 044014  004737  021042                  JSR     PC,SCHED                ;GO DO IT ON ALL DRIVES
   6129 044020  005737  003704                  TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
   6130 044024  001477                          BEQ     T3EXIT                  ;GET OUT IF NONE LEFT TO TEST
   6131 044026  005337  010744                  DEC     LOOPS                   ;SUBTRACT 1 FROM THE TAPE MARK COUNT
   6132 044032  001364                          BNE     15$                     ;KEEP GOING TIL THEY'RE ALL WRITTEN
   6133
   6134 044034  004737  033550                  JSR     PC,CLREOT               ;CLEAR THE EOT INDICATORS
   6135 044040  012705  044246                  MOV     #T3REW,R5               ;SET UP TO REWIND ALL DRIVES
   6136 044044  004737  021042                  JSR     PC,SCHED                ;GO DO IT
   6137 044050  005737  003704                  TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
   6138 044054  001463                          BEQ     T3EXIT                  ;GET OUT IF NONE LEFT TO TEST
   6139 044056  004737  033772                  JSR     PC,PATCLR               ;START AT PATTERN 1
   6140 044062  004737  033636                  JSR     PC,SDSTUP               ;RESET THE RANDOM SEEDS
```

```
6141
6142 044066  012705  044270          20$:    MOV     #T3RD,R5                ;SET UP TO READ AN ITERATION SET
6143 044072  004737  02.042                  JSR     PC,SCHED                ;GO ISSUE TO ALL DRIVES
6144 044076  005737  003704                  TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
6145 044102  001450                          BEQ     T3EXIT                  ;GET OUT IF NONE LEFT TO TEST
6146 044104  023737  003706  003704          CMP     UEOT,UDROP              ;ALL UNITS AT EOT ?
6147 044112  001413                          BEQ     EX3REW                  ;YES, GO REWIND ALL DRIVES
6148
6149 044114  012705  044276                  MOV     #T3SPO,R5               ;SPACE 1 OBJECT (TAPE MARK)
6150 044120  004737  021042                  JSR     PC,SCHED                ;GO DO IT AN ALL DRIVES
6151 044124  005737  003704                  TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
6152 044130  001435                          BEQ     T3EXIT                  ;GET OUT IF NONE LEFT TO TEST
6153 044132  023737  003706  003704          CMP     UEOT,UDROP              ;ALL UNITS AT EOT ?
6154 044140  001352                          BNE     20$                     ;NO, KEEP READING
6155
6156 044142  004737  033550          EX3REW: JSR     PC,CLREOT               ;CLEAR THE EOT INDICATORS
6157 044146  012705  044246                  MOV     #T3REW,R5               ;SET UP TO REWIND ALL DRIVES
6158 044152  004737  021042                  JSR     PC,SCHED                ;GO DO IT
6159
6160 044156  005737  003702          220$:   TST     PASS1                   ;IS THIS THE END OF 1ST PASS ?
6161 044162  001020                          BNE     T3EXIT                  ;BRANCH IF NOT
6162 044164  005337  003702                  DEC     PASS1                   ;PASS1 = -1
6163 044170  022737  000002  003732          CMP     #TF.PE,FORMAT           ;WAS 1ST PASS DONE IN PE ?
6164 044176  001405                          BEQ     225$                    ;BRANCH IF SO
6165 044200  012737  000002  003732          MOV     #TF.PE,FORMAT           ;SET TAPE FORMAT TO PE
6166 044206  000137  043554                  JMP     GO3                     ;REPEAT TEST. THIS TIME IN PE
6167
6168 044212  012737  000004  003732  225$:   MOV     #TF.GCR,FORMAT          ;SET TAPE FORMAT TO GCR
6169 044220  000137  043554                  JMP     GO3                     ;REPEAT TEST. THIS TIME IN GCR
6170
6171 044224  005037  003702          T3EXIT: CLR     PASS1                   ;RESET "1ST PASS" FLAG FOR NEXT TEST
6172 044230  013737  003734  003732          MOV     INFORM,FORMAT           ;RESTORE INITIAL TEST FORMAT
6173 044236  004737  033714                  JSR     PC,SDSAVE               ;RESET THE RANDOM SEEDS
6174 044242                                  EXIT    TST
     044242  104432                          TRAP    C$EXIT
     044244  000040                          .WORD   L10017-.
6175
```

```
 6177 044246     160                T3REW:  .BYTE   REW                     ;REWIND
 6178 044247     000                        .BYTE   NULPAT
 6179 044250  000000                        .WORD   0
 6180 044252  000001                        .WORD   1
 6181
 6182 044254     020                T3WRT:  .BYTE   WR                      ;WRITE RECORDS
 6183 044255     200                        .BYTE   ALLPAT
 6184 044256  000000                        .WORD   RNDBYT
 6185 044260  000000                        .WORD   RNDITR
 6186
 6187 044262     100                T3WTM:  .BYTE   WTM                     ;WRITE TAPE MARK
 6188 044263     000                        .BYTE   NULPAT
 6189 044264  000000                        .WORD   0
 6190 044266  000001                        .WORD   1
 6191
 6192 044270     010                T3RD:   .BYTE   RD                      ;READ RECORDS
 6193 044271     200                        .BYTE   ALLPAT
 6194 044272  000000                        .WORD   RNDBYT
 6195 044274  000000                        .WORD   RNDITR
 6196
 6197 044276     070                T3SPO:  .BYTE   SPO                     ;SPACE OBJECT (TAPE MARK)
 6198 044277     000                        .BYTE   NULPAT
 6199 044300  000001                        .WORD   1
 6200 044302  000001                        .WORD   1
 6201
 6202                                       .EVEN
 6203 044304                        ENDTST
      044304                        L10017:
      044304  104401                        TRAP    C$ETST
```

```
6205                                    .SBTTL  TEST 4: Write Interchange Tape
6206
6207                                    ;++
6208                                    ;This test will rewind the tape, then write until EOT or a fatal error is
6209                                    ;encountered This test  will keep track of the number of records and tape
6210                                    ;marks written.  If a fatal error  is encountered, a  message will report
6211                                    ;it, and the unit prevented from executing further write operations.
6212                                    ;--
6213 044306                                    BGNTST
     044306                             T4::
6214
6215 044306  005737  003704            START4: TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
6216 044312  001014                            BNE     5$                      ;GO START THE TEST
6217 044314                                    PRINTF  #BYPASS,L$TEST          ;PRINT THE TEST BYPASSED MESSAGE
     044314  013746  002114                    MOV     L$TEST,-(SP)
     044320  012746  020527                    MOV     #BYPASS,-(SP)
     044324  012746  000002                    MOV     #2,-(SP)
     044330  010600                            MOV     SP,R0
     044332  104417                            TRAP    C$PNTF
     044334  062706  000006                    ADD     #6,SP
6218 044340  000137  044702                    JMP     T4EXIT                  ;GET OUT IF NONE LEFT TO TEST
6219
6220 044344  105737  002216            5$:     TSTB    CLOCK                   ;IS THE CLOCK ENABLED
6221 044350  001421                            BEQ     G04                     ;NO, THEN CAN'T PRINT TIME
6222 044352                                    PRINTF  #TIME,<B,HOURS>,<B,MINUTE>,<B,SECOND>,
     044352  005046                            CLR     -(SP)
     044354  153716  002221                    BISB    SECOND,(SP)
     044360  005046                            CLR     -(SP)
     044362  153716  002220                    BISB    MINUTE,(SP)
     044366  005046                            CLR     -(SP)
     044370  153716  002217                    BISB    HOURS,(SP)
     044374  012746  020037                    MOV     #TIME,-(SP)
     044400  012746  000004                    MOV     #4,-(SP)
     044404  010600                            MOV     SP,R0
     044406  104417                            TRAP    C$PNTF
     044410  062706  000012                    ADD     #12,SP
6223
6224 044414  004737  033550            G04:    JSR     PC,CLREOT               ;MAKE SURE EOT STATUS IS CLEAR
6225 044420  012737  000100  003616            MOV     #ONLB,TSTMSK            ;ALLOW ALREADY ONLINE STATUS
6226
6227 044426  022737  000002  003732            CMP     #TF.PE,FORMAT           ;ARE WE DOING PE ?
6228 044434  001011                            BNE     1$                      ;NO, PRINT GCR
6229 044436                                    PRINTF  #TSTPE                  ;PRINT TESTING IN PE
     044436  012746  020610                    MOV     #TSTPE,-(SP)
     044442  012746  000001                    MOV     #1,-(SP)
     044446  010600                            MOV     SP,R0
     044450  104417                            TRAP    C$PNTF
     044452  062706  000004                    ADD     #4,SP
6230 044456  000410                            BR      3$                      ;START TEST
6231 044460                            1$:     PRINTF  #TSTGCR                 ;PRINT TESTING IN GCR
     044460  012746  020562                    MOV     #TSTGCR,-(SP)
     044464  012746  000001                    MOV     #1,-(SP)
     044470  010600                            MOV     SP,R0
     044472  104417                            TRAP    C$PNTF
     044474  062706  000004                    ADD     #4,SP
6232 044500  012705  042324            3$:     MOV     #TSTSUC,R5              ;SET UP TO DO A SET UNIT CHAR
6233 044504  004737  021042                    JSR     PC,SCHED                ;GO ISSUE THE COMMAND
```

TEST 4: Write Interchange Tape

```
6234 044510  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
6235 044514  001002                      BNE     4$              ;NO, CONTINUE
6236 044516  000137  044702              JMP     T4EXIT          ;GET OUT IF NONE LEFT TO TEST
6237
6238 044522  004737  033772      4$:     JSR     PC,PATCLR       ;
6239 044526  004737  033636              JSR     PC,SDSTUP       ;SET UP THE RANDOM SEEDS
6240 044532  005037  003616              CLR     TSTMSK          ;NO ALLOWABLE ERRORS
6241
6242 044536  012705  044714              MOV     #T4REW,R5       ;POINT R5 TO THE REWIND TABLE
6243 044542  004737  021042              JSR     PC,SCHED        ;GO START THE TEST
6244 044546  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
6245 044552  001453                      BEQ     T4EXIT          ;GET OUT IF NONE LEFT TO TEST
6246
6247 044554  012705  044722      5$:     MOV     #T4WRT,R5       ;POINT R5 TO THE TEST TABLE
6248 044560  004737  021042              JSR     PC,SCHED        ;GO START THE TEST
6249 044564  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
6250 044570  001444                      BEQ     T4EXIT          ;GET OUT IF NONE LEFT TO TEST
6251 044572  023737  003706  003704      CMP     UEOT,UDROP      ;ARE THEY ALL AT EOT ?
6252 044600  001413                      BEQ     10$             ;BRANCH IF IT IS
6253
6254 044602  012705  044730              MOV     #T4WTM,R5       ;POINT R5 TO THE TEST TABLE
6255 044606  004737  021042              JSR     PC,SCHED        ;GO START THE TEST
6256 044612  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
6257 044616  001431                      BEQ     T4EXIT          ;GET OUT IF NONE LEFT TO TEST
6258 044620  023737  003706  003704      CMP     UEOT,UDROP      ;ARE THEY ALL AT EOT ?
6259 044626  001352                      BNE     5$              ;BRANCH IF NOT
6260
6261 044630  012737  000004  010744  10$:     MOV     #4,LOOPS        ;SET UP TO DO 4 TAPE MARKS
6262 044636  004737  033550      15$:    JSR     PC,CLREOT       ;CLEAR THE EOT INDICATORS
6263 044642  012705  044262              MOV     #T3WTM,R5       ;SET UP TO WRITE A TAPE MARK
6264 044646  004737  021042              JSR     PC,SCHED        ;GO DO IT ON ALL DRIVES
6265 044652  005737  003704              TST     UDROP           ;HAVE ALL UNITS BEEN DROPPED ?
6266 044656  001411                      BEQ     T4EXIT          ;GET OUT IF NONE LEFT TO TEST
6267 044660  005337  010744              DEC     LOOPS           ;SUBTRACT 1 FROM THE TAPE MARK COUNT
6268 044664  001364                      BNE     15$             ;KEEP GOING TIL THEY'RE ALL WRITTEN
6269
6270 044666  004737  033550              JSR     PC,CLREOT       ;
6271 044672  012705  044714              MOV     #T4REW,R5       ;POINT R5 TO THE TEST TABLE
6272 044676  004737  021042              JSR     PC,SCHED        ;GO REWIND ALL UNITS
6273
6274 044702  013737  003734  003732  T4EXIT: MOV     INFORM,FORMAT   ;RESTORE INITIAL TEST FORMAT
6275 044710                              EXIT    TST
     044710  104432                      TRAP    C$EXIT
     044712  000024                      .WORD   L10020-.
6276
6277 044714     160              T4REW:  .BYTE   REW             ;REWIND
6278 044715     000                      .BYTE   NULPAT
6279 044716  000000                      .WORD   0
6280 044720  000001                      .WORD   1
6281
6282 044722     020              T4WRT:  .BYTE   WR              ;WRITE RECORDS
6283 044723     200                      .BYTE   ALLPAT
6284 044724  010000                      .WORD   4096.
6285 044726  001750                      .WORD   1000.
6286
6287 044730     100              T4WTM:  .BYTE   WTM             ;WRITE TAPE MARK
6288 044731     000                      .BYTE   NULPAT
```

TEST 4: Wr te Interchange Tape

```
   6289 044732  000000                          .WORD   0
   629C 044734  000001                          .WORD   1
   6291                                         .EVEN
   6292 044736                                  ENDTST
        044736                          L10020:
        044736  104401                          TRAP    C$ETST
   6293
```

```
                                        .SBTTL  TEST 5: Read Unknown Tape

                                        ;++
                                        ;This test will rewind a tape, then read until EOT, LEOT or fatal error
                                        ;is encountered.  This test will keep track of the number of records
                                        ;and files read.  If a fatal error is encountered, a message will
                                        ;report it, the tape on the unit will be rewound, and the unit
                                        ;prevented from executing further read operations.
                                        ;-
044740                                          BGNTST
044740                                  T5::

044740  005737  003704          STARTS: TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
044744  001014                          BNE     5$                      ;GO START THE TEST
044746                                  PRINTF  #BYPASS,L$TEST          ;PRINT THE TEST BYPASSED MESSAGE
044746  013746  002114                  MOV     L$TEST,-(SP)
044752  012746  020527                  MOV     #BYPASS,-(SP)
044756  012746  000002                  MOV     #2,-(SP)
044762  010600                          MOV     SP,R0
044764  104417                          TRAP    C$PNTF
044766  062706  000006                  ADD     #6,SP
044772  000137  045264                  JMP     T5EXIT                  ;GET OUT IF NONE LEFT TO TEST

044776  105737  002216          5$:     TSTB    CLOCK                   ;IS THE CLOCK ENABLED
045002  001421                          BEQ     G05                     ;NO, THEN CAN'T PRINT TIME
045004                                  PRINTF  #TIME,<B,HOURS>,<B,MINUTE>,<B,SECOND>,
045004  005046                          CLR     -(SP)
045006  153716  002221                  BISB    SECOND,(SP)
045012  005046                          CLR     (SP)
045014  153716  002220                  BISB    MINUTE,(SP)
045020  005046                          CLR     -(SP)
045022  153716  002217                  BISB    HOURS,(SP)
045026  012746  020037                  MOV     #TIME,-(SP)
045032  012746  000004                  MOV     #4,-(SP)
045036  ^10600                          MOV     SP,R0
045040  104417                          TRAP    C$PNTF
045042  062706  000012                  ADD     #12,SP

045046  004737  033550          G05:    JSR     PC,CLREOT               ;MAKE SURE EOT STATUS IS CLEAR
045052  012737  000100  003616          MOV     #ONLB,TSTMSK            ;ALLOW ALREADY ONLINE STATUS

045060  022737  000002  003732          CMP     #TF.PE,FORMAT           ;ARE WE DOING PE ?
045066  001011                          BNE     1$                      ;NO, PRINT GCR
045070                                  PRINTF  #TSTPE                  ;PRINT TESTING IN PE
045070  012746  020610                  MOV     #TSTPE,-(SP)
045074  012746  000001                  MOV     #1,-(SP)
045100  010600                          MOV     SP,R0
045102  104417                          TRAP    C$PNTF
045104  062706  000004                  ADD     #4,SP
045110  000410                          BR      3$                      ;START TEST
045112                          1$:     PRINTF  #TSTGCR                 ;PRINT TESTING IN GCR
045112  012746  020562                  MOV     #TSTGCR,-(SP)
045116  012746  000001                  MOV     #1,-(SP)
045122  010600                          MOV     SP,R0
045124  104417                          TRAP    C$PNTF
045126  062706  000004                  ADD     #4,SP
045132  012705  042324          3$:     MOV     #TSTSUC,R5              ;SET UP TO DO A SET UNIT CHAR
```

```
 6324 045136  004737  021042            JSR     PC,SCHED                ;GO ISSUE THE COMMAND
 6325 045142  005737  003704            TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
 6326 045146  001002                    BNE     4$                      ;NO, CONTINUE
 6327 045150  000137  045264            JMP     T5EXIT                  ;GET OUT IF NONE LEFT TO TEST
 6328
 6329 045154  004737  033772      4$:   JSR     PC,PATCLR               ;MAKE SURE WE START WITH PATTERN 1
 6330 045160  004737  033636            JSR     PC,SDSTUP               .SET UP THE RANDOM SEEDS
 6331
 6332 045164  005037  003616            CLR     TSTMSK                  ;NO ALLOWABLE ERRORS
 6333 045170  012705  045276            MOV     #T5REW,R5               ;POINT R5 TO THE REWIND TABLE
 6334 045174  004737  021042            JSR     PC,SCHED                ;GO START THE TEST
 6335 045200  005737  003704            TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
 6336 045204  001427                    BEQ     T5EXIT                  ;GET OUT IF NONE LEFT TO TEST
 6337
 6338 045206  012737  000012  003616    MOV     #TM8!RDTB,TSTMSK        ;TAPE MARKS AND TRUNC. RECORDS OK
 6339 045214  012705  045304      5$:   MOV     #T5RD,R5                ;POINT R5 TO THE TEST TABLE
 6340 045220  004737  021042            JSR     PC,SCHED                ;GO START THE TEST
 6341 045224  005737  003704            TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
 6342 045230  001415                    BEQ     T5EXIT                  ;GET OUT IF NONE LEFT TO TEST
 6343 045232  C23737  003706  003704    CMP     UEOT,UDROP              ;ARE THEY ALL AT EOT ?
 6344 045240  001365                    BNE     5$                      ;BRANCH IF THEY ARE NOT
 6345
 6346 045242  004737  033550     10$:   JSR     PC,CLREOT               ;
 6347 045246  012705  045276            MOV     #T5REW,R5               ;POINT R5 TO THE REWIND TABLE
 6348 045252  004737  021042            JSR     PC,SCHED                ;GO REWIND ALL UNITS
 6349 045256  012737  001751  045310    MOV     #1001.,T5RD+4           ;RESTORE ITER COUNT
 6350
 6351 045264  013737  003734  003732 T5EXIT: MOV     INFORM,FORMAT           ;RESTORE INITIAL TEST FORMAT
 6352 045272                            EXIT    TST
      045272  104432                    TRAP    C$EXIT
      045274  000016                    .WORD   L10021-.
 6353
 6354 045276     160             T5REW: .BYTE   REW                     ;REWIND
 6355 045277     000                    .BYTE   NULPAT
 6356 045300  000000                    .WORD   0
 6357 045302  000001                    .WORD   1
 6358
 6359 045304     010             T5RD:  .BYTE   RD                      ;READ RECORDS
 6360 045305     200                    .BYTE   ALLPAT
 6361 045306  010000                    .WORD   4096.
 6362 045310  001751                    .WORD   1001.
 6363
 6364                                   .EVEN
 6365 045312                            ENDTST
      045312                    L10021:
      045312  104401                    TRAP    C$ETST
```

```
6367                                    .SBTTL  TEST 6: Start/Stop Write/Read Test
6368
6369                                    ;++
6370                                    ;This test rewinds the tape, and executes the following sequence:
6371                                    ;
6372                                    ;       1.  Write 1300 records one at a time,
6373                                    ;       2.  Write 2 file marks (LEOT),
6374                                    ;       3.  Rewind,
6375                                    ;       4.  Read 1300 records one at a time,
6376                                    ;       5.  Skip to LEOT.
6377                                    ;       6.  Rewind,
6378                                    ;
6379                                    ;This sequence  will  permit  hardware  retries, if not user disabled.
6380                                    ;This test will run until exhaustion of the command count or fatal error
6381                                    ;is  detected.   All  data patterns including random data will be used
6382                                    ;in this test.
6383                                    ;--
6384 045314                                    BGNTST
     045314                             T6::
6385
6386 045314  005737  003704            START6: TST     UDKOP                   ;HAVE ALL UNITS BEEN DROPPED ?
6387 045320  001014                            BNE     5$                      ;GO START THE TEST
6388 045322                                    PRINTF  #BYPASS,L$TEST          ;PRINT THE TEST BYPASSED MESSAGE
     045322  013746  002114                    MOV     L$TEST,-(SP)
     045326  012746  020527                    MOV     #BYPASS,-(SP)
     045332  012746  000002                    MOV     #2,-(SP)
     045336  010600                            MOV     SP,R0
     045340  104417                            TRAP    C$PNTF
     045342  062706  000006                    ADD     #6,SP
6389 045346  000137  046034                    JMP     T6EXIT                  ;GET OUT IF NONE LEFT TO TEST
6390
6391 045352  105737  002216            5$:     TSTB    CLOCK                   ;IS THE CLOCK ENABLED
6392 045356  001421                            BEQ     GO6                     ;NO, THEN CAN'T PRINT TIME
6393 045360                                    PRINTF  #TIME,<B,HOURS>,<B,MINUTE>,<B,SECOND>,
     045360  005046                            CLR     -(SP)
     045362  153716  002221                    BISB    SECOND,(SP)
     045366  005046                            CLR     -(SP)
     045370  153716  002220                    BISB    MINUTE,(SP)
     045374  005046                            CLR     -(SP)
     045376  153716  002217                    BISB    HOURS,(SP)
     045402  012746  020037                    MOV     #TIME,-(SP)
     045406  012746  000004                    MOV     #4,-(SP)
     045412  010600                            MOV     SP,R0
     045414  104417                            TRAP    C$PNTF
     045416  062706  000012                    ADD     #12,SP
6394
6395 045422  004737  033550            GO6:    JSR     PC,CLREOT               ;MAKE SURE EOT STATUS IS CLEAR
6396 045426  012737  000100  003616            MOV     #ONLB,TSTMSK            ;ALLOW ALREADY ONLINE STATUS
6397
6398 045434  022737  000002  003732            CMP     #TF.PE,FORMAT           ;ARE WE DOING PE ?
6399 045442  001011                            BNE     1$                      ;NO, PRINT GCR
6400 045444                                    PRINTF  #TSTPE                  ;PRINT TESTING IN PE
     045444  012746  020610                    MOV     #TSTPE,-(SP)
     045450  012746  000001                    MOV     #1,-(SP)
     045454  010600                            MOV     SP,R0
     045456  104417                            TRAP    C$PNTF
     045460  062706  000004                    ADD     #4,SP
```

TEST 6: Start/Stop Write/Read Test

```
 6401 045464  000410                    BR      3$                      ;START TEST
 6402 045466                    1$:     PRINTF  @TSTGCR                 ;PRINT TESTING IN GCR
      045466  012746  020562            MOV     @TSTGCR,-(SP)
      045472  012746  000001            MOV     #1,-(SP)
      045476  010600                    MOV     SP,R0
      045500  104417                    TRAP    C$PNTF
      045502  062706  000004            ADD     #4,SP
 6403 045506  012705  042324    3$:     MOV     #TSTSUC,R5              ;SET UP TO DO A SET UNIT CHAR
 6404 045512  004737  021042            JSR     PC,SCHED                ;GO ISSUE THE COMMAND
 6405 045516  005737  003704            TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
 6406 045522  001002                    BNE     4$                      ;NO, CONTINUE
 6407 045524  000137  046034            JMP     T6EXIT                  ;GET OUT IF NONE LEFT TO TEST
 6408
 6409 045530  004737  033772    4$:     JSR     PC,PATCLR               ;MAKE SURE WE START WITH PATTERN 1
 6410 045534  004737  033636            JSR     PC,SDSTUP               ;RESET THE RANDOM SEEDS
 6411
 6412 045540  005037  003616            CLR     TSTMSK                  ;ALLOW NO ERRORS
 6413 045544  005737  003704            TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
 6414 045550  001531                    BEQ     T6EXIT                  ;GET OUT IF NONE LEFT TO TEST
 6415
 6416 045552  012705  046056            MOV     #T6REW,R5               ;SET UP TO DO REWIND
 6417 045556  004737  021042            JSR     PC,SCHED                ;GO ISSUE TO ALL DRIVES
 6418 045562  005737  003704            TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
 6419 045566  001522                    BEQ     T6EXIT                  ;GET OUT IF NONE LEFT TO TEST
 6420 045570  012737  002424  003546    MOV     #1300.,COUNT            ;STEP UP THE WRITE ITERATION COUNT
 6421
 6422 045576  012705  046064    5$:     MOV     #T6WRT,R5               ;SET UP A WRITE ITERATION
 6423 045602  004737  021042            JSR     PC,SCHED                ;GO DO IT ON ALL DRIVES
 6424 045606  005737  003704            TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
 6425 045612  001510                    BEQ     T6EXIT                  ;GET OUT IF NONE LEFT TO TEST
 6426 045614  005337  003546            DEC     COUNT                   ;SUBTRACT 1 FROM THE ITERATION COUNT
 6427 045620  001366                    BNE     5$                      ;MORE COMMANDS TO DO, KEEP GOING
 6428
 6429 045622  012705  046072            MOV     #T6WTM,R5               ;SET UP TO WRITE A TAPE MARK
 6430 045626  004737  021042            JSR     PC,SCHED                ;GO DO IT ON ALL DRIVES
 6431 045632  005737  003704            TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
 6432 045636  001476                    BEQ     T6EXIT                  ;GET OUT IF NONE LEFT TO TEST
 6433
 6434 045640  012705  046056            MOV     #T6REW,R5               ;SET UP TO REWIND ALL DRIVES
 6435 045644  004737  021042            JSR     PC,SCHED                ;GO DO IT
 6436 045650  005737  003704            TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
 6437 045654  001467                    BEQ     T6EXIT                  ;GET OUT IF NONE LEFT TO TEST
 6438
 6439 045656  004737  033772            JSR     PC,PATCLR               ;START AT PATTERN 1
 6440 045662  004737  033636            JSR     PC,SDSTUP               ;RESET THE RANDOM SEEDS
 6441 045666  012737  002424  003546    MOV     #1300.,COUNT            ;STEP UP THE WRITE ITERATION COUNT
 6442
 6443 045674  012705  046100    10$:    MOV     #T6RD,R5                ;SET UP TO READ AN ITERATION SET
 6444 045700  004737  021042            JSR     PC,SCHED                ;GO DO IT ON ALL DRIVES
 6445 045704  005737  003704            TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
 6446 045710  001451                    BEQ     T6EXIT                  ;GET OUT IF NONE LEFT TO TEST
 6447 045712  005337  003546            DEC     COUNT                   ;SUBTRACT 1 FROM THE ITERATION COUNT
 6448 045716  001366                    BNE     10$                     ;MORE COMMANDS TO DO, KEEP GOING
 6449
 6450 045720  052737  000001  003616    BIS     #LEDB,TSTMSK            ;SET UP TO ALLOW LEOT DETECTED
 6451 045726  012705  046106            MOV     #T6SKD,R5               ;SKIP TO LEOT
 6452 045732  004737  021042            JSR     PC,SCHED                ;GO DO IT AN ALL DRIVES
```

```
6453 045736  042737  000001  003616          BIC     @LEDB,TSTMSK            ;DISALLOW LEOT DETECTED
6454 045744  005737  003704                  TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
6455 045750  001431                          BEQ     T6EXIT                  ;GET OUT IF NONE LEFT TO TEST
6456
6457 045752  004737  033550                  JSR     PC,CLREOT               ;CLEAR THE EOT INDICATORS
6458 045756  012705  046056                  MOV     #T6REW,R5               ;SET UP TO REWIND ALL DRIVES
6459 045762  004737  021042                  JSR     PC,SCHED                ;GO DO IT
6460
6461 045766  005737  003702          20$:    TST     PASS1                   ;IS THIS THE END OF 1ST PASS ?
6462 045772  001020                          BNE     T6EXIT                  ;BRANCH IF NOT
6463 045774  005337  003702                  DEC     PASS1                   ;PASS1 = -1
6464 046000  022737  000002  003732          CMP     #TF.PE,FORMAT           ;WAS 1ST PASS DONE IN PE ?
6465 046006  001405                          BEQ     25$                     ;BRANCH IF SO
6466 046010  012737  000002  003732          MOV     #TF.PE,FORMAT           ;SET TAPE FORMAT TO PE
6467 046016  000137  045422                  JMP     G06                     ;REPEAT TEST. THIS TIME IN PE
6468
6469 046022  012737  000004  003732  25$:    MOV     #TF.GCR,FORMAT          ;SET TAPE FORMAT TO GCR
6470 046030  000137  045422                  JMP     G06                     ;REPEAT TEST. THIS TIME IN GCR
6471
6472 046034  C05037  003702          T6EXIT: CLR     PASS1                   ;RESET "1ST PASS" FLAG FOR NEXT TEST
6473 046040  013737  003734  003732          MOV     INFORM,FORMAT           ;RESTORE INITIAL TEST FORMAT
6474 046046  004737  033714                  JSR     PC,SDSAVE               ;RESET THE RANDOM SEEDS
6475 046052                                  EXIT    TST
     046052  104432                          TRAP    C$EXIT
     046054  000040                          .WORD   L10022-.
6476
```

```
6478 046056     160                T6REW:  .BYTE   REW                     ;REWIND
6479 046057     000                        .BYTE   NULPAT
6480 046060  000000                        .WORD   0
6481 046062  000001                        .WORD   1.
6482
6483 046064     020                T6WRT:  .BYTE   WR                      ;WRITE RECORDS
6484 046065     200                        .BYTE   ALLPAT
6485 046066  020000                        .WORD   8192.
6486 046070  000001                        .WORD   1.
6487
6488 046072     100                T6WTM:  .BYTE   WTM                     ;WRITE TAPE MARK
6489 046073     000                        .BYTE   NULPAT
6490 046074  000000                        .WORD   0
6491 046076  000002                        .WORD   2.
6492
6493 046100     010                T6RD:   .BYTE   RD                      ;READ RECORDS
6494 046101     200                        .BYTE   ALLPAT
6495 046102  020000                        .WORD   8192.
6496 046104  000001                        .WORD   1.
6497
6498 046106     062                T6SKD:  .BYTE   SKD                     ;SKIP TO LEOT
6499 046107     000                        .BYTE   NULPAT
6500 046110  000001                        .WORD   1
6501 046112  000001                        .WORD   1
6502
6503                                       .EVEN
6504 046114                               ENDTST
     046114                        L10022:
     046114  104401                       TRAP    C$ETST
```

```
 6506                                   .SBTTL  TEST 7: Conversation Test
 6507
 6508                                   ;++
 6509                                   ;Conversation mode will run with or without error  reports.   The  user
 6510                                   ;can  select,  from a list of commands, a sequence which can be used to
 6511                                   ;emulate a known failure mode.  Between commands, the user can  specify
 6512                                   ;unique  delays,  ranging  from 10 to 250 ms.  The user can follow each
 6513                                   ;tape  command  with  integer  values,   the   first   indicating   the
 6514                                   ;byte/record/file  count and the second indicating the # of repetitions
 6515                                   ;necessary for that command.
 6516                                   ;--
 6517 046116                                   BGNTST
      046116                           T7::
 6518
 6519 046116  005737  003704           START7: TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
 6520 046122  001014                           BNE     5$                      ;GO START THE TEST
 6521 046124                                   PRINTF  #BYPASS,L$TEST           PRINT THE TEST BYPASSED MESSAGE
      046124  013746  002114                   MOV     L$TEST,-(SP)
      046130  012746  020527                   MOV     #BYPASS,-(SP)
      046134  012746  000002                   MOV     #2,-(SP)
      046140  010600                           MOV     SP,R0
      046142  104417                           TRAP    C$PNTF
      046144  062706  000006                   ADD     #6,SP
 6522 046150  000137  046464                   JMP     T7EXIT                  ;GET OUT IF NONE LEFT TO TEST
 6523
 6524 046154  105737  002216           5$:     TSTB    CLOCK                   ;IS THE CLOCK ENABLED
 6525 046160  001421                           BEQ     GO7                     ;NO, THEN CAN'T PRINT TIME
 6526 046162                                   PRINTF  #TIME,<B,HOURS>,<B,MINUTE>,<B,SECOND>,
      046162  005046                           CLR     -(SP)
      046164  153716  002221                   BISB    SECOND,(SP)
      046170  005046                           CLR     -(SP)
      046172  153716  002220                   BISB    MINUTE,(SP)
      046176  005046                           CLR     -(SP)
      046200  153716  002217                   BISB    HOURS,(SP)
      046204  012746  020037                   MOV     #TIME,-(SP)
      046210  012746  000004                   MOV     #4,-(SP)
      046214  010600                           MOV     SP,R0
      046216  104417                           TRAP    C$PNTF
      046220  062706  000012                   ADD     #12,SP
 6527
 6528 046224  004737  033550           GO7:    JSR     PC,CLREOT               ;MAKE SURE EOT STATUS IS CLEAR
 6529 046230  012737  000100  003616           MOV     #ONLB,TSTMSK            ;ALLOW ALREADY ONLINE STATUS
 6530
 6531 046236  012705  042332                   MOV     #T1ONL,R5               ;SET UP TO DO AN ONLINE
 6532 046242  004737  021042                   JSR     PC,SCHED                ;GO ISSUE THE COMMAND
 6533 046246  005737  003704                   TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
 6534 046252  001002                           BNE     1$                      ;NO, CONTINUE
 6535 046254  000137  046464                   JMP     T7EXIT                  ;GET OUT IF NONE LEFT TO TEST
 6536
 6537 046260  005037  003554           1$:     CLR     BRCNT                   ; CLEAR THE BRANCH COUNTER
 6538 046264  012705  002232                   MOV     #T7TBL,R5               ;POINT R5 TO TEST 7 TABLE
 6539 046270  005037  003616                   CLR     TSTMSK                  ;NO ALLOWABLE ERRORS
 6540 046274  052737  000001  003616           BIS     #LEDB,TSTMSK            ;SET UP TO ALLOW LEOT DETECTED
 6541
 6542 046302  004737  033636           10$:    JSR     PC,SDSTUP               ;SET UP THE RANDOM SEEDS
 6543 046306  004737  033772                   JSR     PC,PATCLR               ;USE THE SAME PATTERN
 6544 046312  062705  000006                   ADD     #TSTSTP,R5              ;MOVE R5 TO THE NEXT TEST TABLE
```

```
6545 046316  122765  177777  000000          CMPB    #-1,CMD(R5)             ;IS IT AN END OF SEQUENCE COMMAND ?
6546 046324  001457                          BEQ     T7EXIT                  ;GET OUT IF IT IS
6547 046326  122765  000310  000000          CMPB    #310,CMD(R5)            ;IS IT A DELAY COMMAND ?
6548 046334  001011                          BNE     40$                     ;NO KEEP GOING
6549 046336  016501  000002                  MOV     ITMCNT(R5),R1           ;PUT THE DELAY VARIABLE IN R1
6550
6551 046342  013702  002116          20$:    MOV     L$DLY,R2                ;GET THE PROGRAM DELAY VARIABLE
6552 046346  005302                  30$:    DEC     R2                      ;DECREMENT R2
6553 046350  001376                          BNE     30$                     ;KEEP GOING TILL ZERO
6554 046352  005301                          DEC     R1                      ;DECREMENT R1
6555 046354  001372                          BNE     20$                     ;RESET R2 AND GO TILL R1 IS ZERO
6556 046356  000751                          BR      10$                     ;GET THE NEXT COMMAND
6557
6558 046360  122765  000300  000000  40$:    CMPB    #300,CMD(R5)            ;IS IT A BRANCH COMMAND ?
6559 046366  001031                          BNE     80$                     ;NO, GO ON
6560 046370  032737  000001  003674          BIT     #T7BRFL,PCFLAG          ; HAVE WE ALREADY LOAD THE BRANCH COUNT ?
6561 046376  001006                          BNE     50$                     ; YES, DON'T OVER WRITE IT
6562 046400  052737  000001  003674          BIS     #T7BRFL,PCFLAG          ; REMEMBER THAT YOUR IN A LOOP
6563 046406  016537  000004  003554          MOV     ITRCNT(R5),BRCNT        ; ITERATION COUNT TO BRANCH COUNT
6564
6565 046414  005337  003554          50$:    DEC     BRCNT                   ; HAVE WE BRANCHED THE REQUIRED TIMES ?
6566 046420  001004                          BNE     60$                     ; NO, CONTINUE LOOPING
6567 046422  042737  000001  003674          BIC     #T7BRFL,PCFLAG          ; CLEAR THE BRANCH FLAG
6568 046430  000724                          BR      10$                     ; GET NEXT COMMAND
6569 046432  016501  000002          60$:    MOV     ITMCNT(R5),R1           ;PUT THE COMMAND NUMBER IN R1
6570 046436  012705  002232                  MOV     #T7TBL,R5               ;POINT R5 TO TEST 7 TABLE
6571
6572 046442  062705  000006          70$:    ADD     #TSTSTP,R5              ;START STEPPING R5 TO THE COMMAND
6573 046446  005301                          DEC     R1                      ;IS THIS THE RIGHT PLACE ?
6574 046450  001374                          BNE     70$                     ;NO, KEEP GOING
6575
6576 046452  004737  021042          80$:    JSR     PC,SCHED                ;GO START THE TEST
6577 046456  005737  003704                  TST     UDROP                   ;HAVE ALL UNITS BEEN DROPPED ?
6578 046462  001307                          BNE     10$                     ;NO, KEEP TESTING
6579 046464  004737  033714          T7EXIT: JSR     PC,SDSAVE               ;SAVE THE RANDOM SEEDS
6580 046470                                  EXIT    TST
     046470  104432                          TRAP    C$EXIT
     046472  000002                          .WORD   L10023-.
6581
6582
6583                                         .EVEN
6584
6585 046474                                  ENDTST
     046474                          L10023:
     046474  104401                          TRAP    C$ETST
6586
6587 046476                                  ENDMOD
6588
```

```
 6591
 6593                                   .TITLE PARAMETER CODING
 6604
 6605                                   .SBTTL  HARDWARE PARAMETER CODING SECTION
 6633
 6634 046476                                   BGNMOD
 6635
 6636                                   ;++
 6637                                   ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
 6638                                   ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
 6639                                   ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
 6640                                   ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
 6641                                   ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
 6642                                   ; WITH THE OPERATOR.
 6643                                   ;--
 6644
 6645 046476                                   BGNHRD
      046476  000032                           .WORD L10024-L$HARD/2
      046500                           L$HARD::
 6646
 6647 046500                                   GPRMA   TKIPAD,0,0,160002,177564,YES
      046500  000031                           .WORD   T$CODE
      046502  046524                           .WORD   TKIPAD
      046504  160002                           .WORD   T$LOLIM
      046506  177564                           .WORD   T$HILIM
 6648 046510                                   GPRMD   TKUNT,2,0,777,0,251,YES
      046510  001032                           .WORD   T$CODE
      046512  046541                           .WORD   TKUNT
      046514  000777                           .WORD   777
      046516  000000                           .WORD   T$LOLIM
      046520  000251                           .WORD   T$HILIM
 6649
 6650 046522                                   EXIT HRD
      046522  021004                           .WORD   T$CODE
 6651
 6652
 6653 046524     124     113     111   TKIPAD: .ASCIZ  ?TKIP ADDRESS?
 6654 046541     124     057     115   TKUNT:  .ASCIZ  ?T/MSCP UNIT NUMBER?
 6655                                   .EVEN
 6656 046564                                   ENDHRD
                                               .EVEN
      046564                           L10024:
```

```
6658                                    .SBTTL  SOFTWARE PARAMETER CODING SECTION
6659
6660                                    ;++
6661                                    ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
6662                                    ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
6663                                    ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
6664                                    ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
6665                                    ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
6666                                    ; WITH THE OPERATOR.
6667                                    ;--
6668
6669 046564                                     BGNSFT
     046564  C01007                             .WORD L10025-L$SOFT/2
     046566                             L$SOFT::
6670
6671 046566                                     GPRML   ECLK,0,1,YES
     046566  000130                             .WORD   T$CODE
     046570  047422                             .WORD   ECLK
     046572  000001                             .WORD   1
6672 046574                                     XFERF   5$
     046574  013044                             .WORD   T$CODE
6673 046576                                     GPRMD   HOUR,0,D,177400,0,24.,,YES
     046576  000052                             .WORD   T$CODE
     046600  047453                             .WORD   HOUR
     046602  177400                             .WORD   177400
     046604  000000                             .WORD   T$LOLIM
     046606  000030                             .WORD   T$HILIM
6674 046610                                     GPRMD   MINT,2,D,000377,0,60.,,YES
     046610  001052                             .WORD   T$CODE
     046612  047535                             .WORD   MINT
     046614  000377                             .WORD   000377
     046616  000000                             .WORD   T$LOLIM
     046620  000074                             .WORD   T$HILIM
6675
6676 046622                             5$:     GPRML   CTPA,4,400,YES
     046622  002130                             .WORD   T$CODE
     046624  047602                             .WORD   CTPA
     046626  000400                             .WORD   400
6677 046630                                     XFERF   10$
     046630  007044                             .WORD   T$CODE
6678 046632                                     GPRML   SERC,6,1,YES
     046632  003130                             .WORD   T$CODE
     046634  047637                             .WORD   SERC
     046636  000001                             .WORD   1
6679 046640                                     GPRML   SERR,6,400,YES
     046640  003130                             .WORD   T$CODE
     046642  047703                             .WORD   SERR
     046644  000400                             .WORD   400
6680
6681 046646                             10$:    GPRML   DENS,10,1,YES
     046646  004130                             .WORD   T$CODE
     046650  047745                             .WORD   DENS
     046652  000001                             .WORD   1
6682 046654                                     GPRML   PRPA,10,400,YES
     046654  004130                             .WORD   T$CODE
     046656  050016                             .WORD   PRPA
     046660  000400                             .WORD   400
```

```
6683 046662                              XFERF   15$
     046662  016044                      .WORD   T$CODE
6684 046664                              GPRML   SOER,12,1,YES
     046664  005130                      .WORD   T$CODE
     046666  050051                      .WORD   SOER
     046670  000001                      .WORD   1
6685 046672                              XFERT   11$
     046672  004024                      .WORD   T$CODE
6686 046674                              GPRML   SRER,12,400,YES
     046674  005130                      .WORD   T$CODE
     046676  050114                      .WORD   SRER
     046700  000400                      .WORD   400
6687 046702                      11$:    GPRML   NOCL,14,1,YES
     046702  006130                      .WORD   T$CODE
     046704  050156                      .WORD   NOCL
     046706  000001                      .WORD   1
6688 046710                              GPRML   PDMP,14,400,YES
     046710  006130                      .WORD   T$CODE
     046712  050221                      .WORD   PDMP
     046714  000400                      .WORD   400
6689
6690 046716                      15$:    GPRML   TSPA,16,1,YES
     046716  007130                      .WORD   T$CODE
     046720  050264                      .WORD   TSPA
     046722  000001                      .WORD   1
6691 046724                              XFERF   20$
     046724  015044                      .WORD   T$CODE
6692 046726                              GPRMD   PATE,16,0,177400,0,7,YES
     046726  007032                      .WORD   T$CODE
     046730  050313                      .WORD   PATE
     046732  177400                      .WORD   177400
     046734  000000                      .WORD   T$LOLIM
     046736  000007                      .WORD   T$HILIM
6693 046740                              GPRML   TSCP,20,1,YES
     046740  010130                      .WORD   T$CODE
     046742  050331                      .WORD   TSCP
     046744  000001                      .WORD   1
6694 046746                              GPRML   CHGF,20,400,YES
     046746  010130                      .WORD   T$CODE
     046750  050371                      .WORD   CHGF
     046752  000400                      .WORD   400
6695 046754                              XFERT   25$
     046754  002024                      .WORD   T$CODE
6696 046756                      20$:    XFER    SFTEX1
     046756  076004                      .WORD   T$CODE
6697
6698 046760                      25$:    GPRMD   CMD1,22,0,000377,0,377,YES
     046760  011032                      .WORD   T$CODE
     046762  050532                      .WORD   CMD1
     046764  000377                      .WORD   000377
     046766  000000                      .WORD   T$LOLIM
     046770  000377                      .WORD   T$HILIM
6699 046772                              GPRMD   DPAT,22,0,177400,0,7,YES
     046772  011032                      .WORD   T$CODE
     046774  050424                      .WORD   DPAT
     046776  177400                      .WORD   177400
     047000  000000                      .WORD   T$LOLIM
```

SOFTWARE PARAMETER CODING SECTION

```
                047002  000007                          .WORD   T$HILIM
6700            047004                                  GPRMD   ICNT,24,D,177777,0,MAXBUF,YES
                047004  012052                          .WORD   T$CODE
                047006  050443                          .WORD   ICNT
                047010  177777                          .WORD   177777
                047012  000000                          .WORD   T$LOLIM
                047014  020000                          .WORD   T$HILIM
6701            047016                                  GPRMD   ITER,26,D,177777,0,65000,YES
                047016  013052                          .WORD   T$CODE
                047020  050505                          .WORD   ITER
                047022  177777                          .WORD   177777
                047024  000000                          .WORD   T$LOLIM
                047026  065000                          .WORD   T$HILIM
6702
6703            047030                                  GPRMD   CMD2,30,0,000377,0,377,YES
                047030  014032                          .WORD   T$CODE
                047032  050540                          .WORD   CMD2
                047034  000377                          .WORD   000377
                047036  000000                          .WORD   T$LOLIM
                047040  000377                          .WORD   T$HILIM
6704            047042                                  GPRMD   DPAT,30,0,177400,0,7,YES
                047042  014032                          .WORD   T$CODE
                047044  050424                          .WORD   DPAT
                047046  177400                          .WORD   177400
                047050  000000                          .WORD   T$LOLIM
                047052  000007                          .WORD   T$HILIM
6705            047054                                  GPRMD   ICNT,32,D,177777,0,MAXBUF,YES
                047054  015052                          .WORD   T$CODE
                047056  050443                          .WORD   ICNT
                047060  177777                          .WORD   177777
                047062  000000                          .WORD   T$LOLIM
                047064  020000                          .WORD   T$HILIM
6706            047066                                  GPRMD   ITER,34,D,177777,0,65000,YES
                047066  016052                          .WORD   T$CODE
                047070  050505                          .WORD   ITER
                047072  177777                          .WORD   177777
                047074  000000                          .WORD   T$LOLIM
                047076  065000                          .WORD   T$HILIM
6707
6708            047100                                  GPRMD   CMD3,36,0,000377,0,377,YES
                047100  017032                          .WORD   T$CODE
                047102  050546                          .WORD   CMD3
                047104  000377                          .WORD   000377
                047106  000000                          .WORD   T$LOLIM
                047110  000377                          .WORD   T$HILIM
6709            047112                                  GPRMD   DPAT,36,0,177400,0,7,YES
                047112  017032                          .WORD   T$CODE
                047114  050424                          .WORD   DPAT
                047116  177400                          .WORD   177400
                047120  000000                          .WORD   T$LOLIM
                047122  000007                          .WORD   T$HILIM
6710            047124                                  GPRMD   ICNT,40,D,177777,0,MAXBUF,YES
                047124  020052                          .WORD   T$CODE
                047126  050443                          .WORD   ICNT
                047130  177777                          .WORD   177777
                047132  000000                          .WORD   T$LOLIM
                047134  020000                          .WORD   T$HILIM
```

SOFTWARE PARAMETER CODING SECTION

```
6711 047136                                GPRMD   ITER,42,D,177777,0,65000,YES
     047136  021052                        .WORD   T$CODE
     047140  050505                        .WORD   ITER
     047142  177777                        .WORD   177777
     047144  000000                         WORD   T$LOLIM
     047146  065000                        .WORD   T$HILIM
6712 047150                                XFER    CONT1
     047150  002004                        .WORD   T$CODE
6713
6714 047152                        SFTEX1: XFER    SFTEX2
     047152  076004                        .WORD   T$CODE
6715
6716 047154                        CONT1:  GPRMD   CMD4,44,O,000377,0,377,YES
     047154  022032                        .WORD   T$CODE
     047156  050554                        .WORD   CMD4
     047160  000377                        .WORD   000377
     047162  000000                        .WORD   T$LOLIM
     047164  000377                        .WORD   T$HILIM
6717 047166                                GPRMD   DPAT,44,O,177400,0,7,YES
     047166  022032                        .WORD   T$CODE
     047170  050424                        .WORD   DPAT
     047172  177400                        .WORD   177400
     047174  000000                        .WORD   T$LOLIM
     047176  000007                        .WORD   T$HILIM
6718 047200                                GPRMD   ICNT,46,D,177777,0,MAXBUF,YES
     047200  023052                        .WORD   T$CODE
     047202  050443                        .WORD   ICNT
     047204  177777                        .WORD   177777
     047206  000000                        .WORD   T$LOLIM
     047210  020000                        .WORD   T$HILIM
6719 047212                                GPRMD   ITER,50,D,177777,0,65000,YES
     047212  024052                        .WORD   T$CODE
     047214  050505                        .WORD   ITER
     047216  177777                        .WORD   177777
     047220  000000                        .WORD   T$LOLIM
     047222  065000                        .WORD   T$HILIM
6720
6721 047224                                GPRMD   CMD5,52,O,000377,0,377,YES
     047224  025032                        .WORD   T$CODE
     047226  050562                        .WORD   CMD5
     047230  000377                        .WORD   000377
     047232  000000                        .WORD   T$LOLIM
     047234  000377                        .WORD   T$HILIM
6722 047236                                GPRMD   DPAT,52,O,177400,0,7,YES
     047236  025032                        .WORD   T$CODE
     047240  050424                        .WORD   DPAT
     047242  177400                        .WORD   177400
     047244  000000                        .WORD   T$LOLIM
     047246  000007                        .WORD   T$HILIM
6723 047250                                GPRMD   ICNT,54,D,177777,0,MAXBUF,YES
     047250  026052                        .WORD   T$CODE
     047252  050443                        .WORD   ICNT
     047254  177777                        .WORD   177777
     047256  000000                        .WORD   T$LOLIM
     047260  020000                        .WORD   T$HILIM
6724 047262                                GPRMD   ITER,56,D,177777,0,65000,YES
     047262  027052                        .WORD   T$CODE
```

```
                047264  050505                          .WORD   ITER
                047266  177777                          .WORD   177777
                047270  000000                          .WORD   T$LOLIM
                047272  065000                          .WORD   T$HILIM
        6725
        6726    047274                                  GPRMD   CMD6,60,0,000377,0,377,YES
                047274  030032                          .WORD   T$CODE
                047276  050570                          .WORD   CMD6
                047300  000377                          .WORD   000377
                047302  000000                          .WORD   T$LOLIM
                047304  000377                          .WORD   T$HILIM
        6727    047306                                  GPRMD   DPAT,60,0,177400,0,7,YES
                047306  030032                          .WORD   T$CODE
                047310  050424                          .WORD   DPAT
                047312  177400                          .WORD   177400
                047314  000000                          .WORD   T$LOLIM
                047316  000007                          .WORD   T$HILIM
        6728    047320                                  GPRMD   ICNT,62,D,177777,0,MAXBUF,YES
                047320  031052                          .WORD   T$CODE
                047322  050443                          .WORD   ICNT
                047324  177777                          .WORD   177777
                047326  000000                          .WORD   T$LOLIM
                047330  020000                          .WORD   T$HILIM
        6729    047332                                  GPRMD   ITER,64,D,177777,0,65000,YES
                047332  032052                          .WORD   T$CODE
                047334  050505                          .WORD   ITER
                047336  177777                          .WORD   177777
                047340  000000                          .WORD   T$LOLIM
                047342  065000                          .WORD   T$HILIM
        6730    047344                                  XFER    CONT2
                047344  002004                          .WORD   T$CODE
        6731
        6732    047346                          SFTEX2: XFER    SFTEX3
                047346  025004                          .WORD   T$CODE
        6733
        6734    047350                          CONT2:  GPRMD   CMD7,66,0,000377,0,377,YES
                047350  033032                          .WORD   T$CODE
                047352  050576                          .WORD   CMD7
                047354  000377                          .WORD   000377
                047356  000000                          .WORD   T$LOLIM
                047360  000377                          .WORD   T$HILIM
        6735    047362                                  GPRMD   DPAT,66,0,177400,0,7,YES
                047362  033032                          .WORD   T$CODE
                047364  050424                          .WORD   DPAT
                047366  177400                          .WORD   177400
                047370  000000                          .WORD   T$LOLIM
                047372  000007                          .WORD   T$HILIM
        6736    047374                                  GPRMD   ICNT,70,D,177777,0,MAXBUF,YES
                047374  034052                          .WORD   T$CODE
                047376  050443                          .WORD   ICNT
                047400  177777                          .WORD   177777
                047402  000000                          .WORD   T$LOLIM
                047404  020000                          .WORD   T$HILIM
        6737    047406                                  GPRMD   ITER,72,D,177777,0,65000,YES
                047406  035052                          .WORD   T$CODE
                047410  050505                          .WORD   ITER
                047412  177777                          .WORD   177777
```

```
        047414  000000                          .WORD   T$LOLIM
        047416  065000                          .WORD   T$HILIM
 6738
 6739 047420                            SFTEX3: XFER    SFTEX4
        047420  070004                          .WORD   T$CODE
 6740
 6741 047422     105     116     101    ECLK:   .ASCIZ  /ENABLE TIME OF DAY CLOCK/
 6742 047453     040     111     116    HOUR:   .ASCIZ  / INPUT HOUR IN 24 HOUR FORMAT (OMIT LEADING ZERO)/
 6743 047535     040     111     116    MINT:   .ASCIZ  / INPUT MINUTES (OMIT LEADING ZERO)/
 6744                                            .EVEN
 6745
 6746 047600                            SFTEX4: XFER    SFTEX5
        047600  106004                          .WORD   T$CODE
 6747
 6748 047602     103     110     101    CTPA:   .ASCIZ  /CHANGE CONTROLLER PARAMETERS/
 6749 047637     040     105     116    SERC:   .ASCIZ  / ENABLE CONTROLLER ERROR CORRECTION/
 6750 047703     040     105     116    SERR:   .ASCIZ  / ENABLE CONTROLLER ERROR RECOVERY/
 6751 047745     040     111     116    DENS:   .ASCIZ  / INTITIAL DENSITY OF EACH TEST IS GCR/
 6752                                            .EVEN
 6753
 6754 050014                            SFTEX5: XFER    SFTEX6
        050014  060004                          .WORD   T$CODE
 6755
 6756 050016     103     110     101    PRPA:   .ASCIZ  /CHANGE PRINTING PARAMETERS/
 6757 050051     040     105     116    SOER:   .ASCIZ  / ENABLE SOFT ERROR REPORT PRINTING/
 6758 050114     040     040     105    SRER:   .ASCIZ  /  ENABLE READ SOFT ERRORS ONLY/
 6759                                            .EVEN
 6760
 6761 050154                            SFTEX6: XFER    SFTEX7
        050154  043004                          .WORD   T$CODE
 6762
 6763 050156     040     105     116    NOCL:   .ASCIZ  / ENABLE CLEAR STATS ON FATAL ERROR/
 6764 050221     040     105     116    PDMP:   .ASCIZ  / ENABLE VARIABLES DUMP ON ERROR/
 6765                                            .EVEN
 6766
 6767 050262                            SFTEX7: XFER    SFTEX8
        050262  060004                          .WORD   T$CODE
 6768
 6769 050264     103     124     101    TSPA:   .ASCIZ  /CHANGE TEST PARAMETERS/
 6770 050313     040     104     101    PATE:   .ASCIZ  / DATA PATTERN/
 6771 050331     040     105     116    T5CP:   .ASCIZ  / ENABLE DATA COMPARES IN TEST 5/
 6772 050371     040     103     110    CHGF:   .ASCIZ  / CHANGE COMMAND SEQUENCE/
 6773                                            .EVEN
 6774
 6775 050422                            SFTEX8: XFER    SFTEX9
        050422  043004                          .WORD   T$CODE
 6776
 6777 050424     040     040     104    DPAT:   .ASCIZ  /  DATA PATTERN/
 6778 050443     040     040     111    ICNT:   .ASCIZ  /  ITEM COUNT (BYTE,RECORD,OBJECT)/
 6779 050505     040     040     111    ITER:   .ASCIZ  /  ITERATION COUNT/
 6780                                            .EVEN
 6781
 6782 050530                            SFTEX9: XFER    SFTEXT
        050530  026004                          .WORD   T$CODE
 6783
 6784 050532     103     115     104    CMD1:   .ASCIZ  "CMD/1"
 6785 050540     103     115     104    CMD2:   .ASCIZ  "CMD/2"
```

```
SOFTWARE PARAMETER CODING SECTION

6786 050546  103  115  104  CMD3:   .ASCIZ  "CMD/3"
6787 050554  103  115  104  CMD4:   .ASCIZ  "CMD/4"
6788 050562  103  115  104  CMD5:   .ASCIZ  "CMD/5"
6789 050570  103  115  104  CMD6:   .ASCIZ  "CMD/6"
6790 050576  103  115  104  CMD7:   .ASCIZ  "CMD/7"
6791                         .EVEN
6792
6793 050604                  SFTEXT:
6794 050604                          ENDSFT
                                     .EVEN
     050604                  L10025:
6795
6802
6803 050604                          ENDMOD
6804
6805 050604                                  .BLKW   4
6806 050614                  RDBUF::         .BLKW   10000   ;READ BUFFER
6807 070614                  WRTBUF::        .BLKW   10000   ;WRITE BUFFER
6808
6809 110614                  PATCH::         .BLKW   50      ;PATCH SPACE
6810
6811 110734                          LASTAD
                                     .EVEN
     110734  110750                .WORD T$FREE
     110736  000004                .WORD T$SIZE
     110740                  L$LAST::
6812
6813 110740                          BGNSETUP        1       ;NUMBER OF P TABLES
6814 110740                          BGNPTAB
     110740  000000                .WORD   0
     110742  000002                .WORD   L10030-./2-1
     110744                  L10026:
6815 110744  174500                        .WORD   174500  ;IP ADDRESS
6816 110746  000000                        .WORD   0       ;UNIT NUMBER
6817 110750                          ENDPTAB
     110750                  L10030:
6818 110750                          ENDSETUP
6819         000001          .END
```

Symbol table

```
ABO    = 000200 G   BUFOFF= 000012 G   CMPT    011116 G   C$GPRI= 000040     DROPUN  040012 G
ABOER    011531     BYPASS  020527 G   CMR     = 000031 G   C$INIT= 000011     DRSPB0  004216 G
ABOR     025622     BYTADD  003624 G   CMSTSV  010734 G   C$INLP= 000020     DRSPB1  005256 G
ABORT  = 000004 G   BYTCNT  022130 G   CNTEL   012622       C$MANI= 000050     DRSPB2  006316 G
ABOT     011046 G   BYTES   003574 G   CNTER   011725       C$MAP = 000102     DRSPB3  007356 G
ACC    = 000040 G   BYTSAV  003746 G   CNTERL  011326       C$MEM = 000031     DRVEL   013146
ACCESS   024744     CCTSAV  003542 G   CNTFLG  010746 G     C$MMU = 000103     DRVER   011746
ACR    = 000041 G   CDRECV  024064 G   CNTHI   010740 G     C$MSG = 000023     DRVT    011156 G
ADJUST   027420     CDRENO  010476 G   CNTT    011146 G     C$OPNR= 000034     DSPSTP= 000004 G
ADR    = 000020 G   CDREN1  010562 G   CNUSAV= 000014 G     C$OPNW= 000104     DSRNG0  010412 G
ALLPAT= 000200 G    CDREN2  010646 G   COLSAV= 000016 G     C$PNTB= 000014     DSRNG1  010476 G
ARETRY   003564 G   CDREN3  010732 G   COMEXI  026140       C$PNTF= 000017     DSRNG2  010562 G
ASSEMB= 000010      CDSRG0  010456 G   CONID = 177777 G     C$PNTS= 000016     DSRNG3  010646 G
AUTCNT   003566 G   CDSRG1  010542 G   CONTPA  002223 G     C$PNTX= 000015     DUMP    020344 G
AVALAB   025266     CDSRG2  010626 G   CONT1   047154       C$PUTB= 000072     DUMPKT  003522 G
AVB    = 000001 G   CDSRG3  010712 G   CONT2   047350       C$PUTW= 000073     DUMP1   020425 G
AVL    = 000130 G   CF.ATN= 000200 G   CORDMP  034022       C$QIO = 000377     DUMP2   020466 G
AVLB   = 000040 G   CF.MSC= 000100 G   COREL   012766       C$RDBU= 000007     ECCBC = 000036 G
AVLER    011562     CF.OTH= 000040 G   COUNT   003546 G     C$REFG= 000047     ECCDC = 000026 G
AVLT     011066 G   CF.THS= 000020 G   COUNTS  020177 G     C$REL = 000077     ECCFLG= 000100 G
AVU    = 000134 G   CHGF    050371       CPRIEX  014252       C$RESE= 000033     ECCTC = 000030 G
BADEL    012647     CHGFLG  002237 G   CRD     = 177776 G   C$REVI= 000004     ECLK    047422
BADER    011674     CHODAT  024724       CRDLIM  010755 G   C$RFLA= 000021     EDCEXT  032142
BADERL   011336     CLOCK   002216 G   CTPA    047602       C$RPT = 000025     EDLEXT  032470
BITO   = 000001 G   CLREOT  033550 G   C$AU    = 000052     C$SEFG= 000046     EF.CON= 000036 G
BIT00  = 000001 G   CLSDRV  023616 G   C$AUTO= 000061       C$SPRI= 000041     EF.EOT= 000010 G
BIT01  = 000002 G   CMD     = 000000 G   C$BRK = 000022     C$SVEC= 000037     EF.LOG= 000040 G
BIT02  = 000004 G   CMDASC  011366 G   C$BSEG= 000004       C$TOME= 000076     EF.NEW= 000035 G
BIT03  = 000010 G   CMDBF1  003752 G   C$BSUB= 000002       DATBL   003650 G   EF.PWR= 000034 G
BIT04  = 000020 G   CMDBF2  004022 G   C$CLCK= 000062       DATPAT= 000001 G   EF.RES= 000037 G
BIT05  = 000040 G   CMDBF3  004072 G   C$CLEA= 000012       DAY     020122 G   EF.SEX= 000020 G
BIT06  = 000100 G   CMDBF4  004142 G   C$CLOS= 000035       DAYS    003750 G   EF.STA= 000040 G
BIT07  = 000200 G   CMDBLD  021764 G   C$CLP1= 000006       DCBEND  004216 G   ENDPAT= 000010 G
BIT08  = 000400 G   CMDCNT  003536 G   C$CPBF= 000074       DCBSTP= 000050 G   EOT   = 000004 G
BIT09  = 001000 G   CMDER   011506       C$CPME= 000075     DCB3SP= 000170 G   EOTBIT= 000002 G
BIT1   = 000002 G   CMDLST= 000001 G   C$CVEC= 000036       DCMDBF  003756 G   EOTPR = 000010 G
BIT10  = 002000 G   CMDONE= 000100 G   C$DCLN= 000044       DCMPER  012540     ERASE   025210
BIT11  = 004000 G   CMDSAV  003744 G   C$DODU= 000051       DCMPT   011276 G   ERASGP  025246
BIT12  = 010000 G   CMDSEQ= 000006 G   C$DRPT= 000024       DENS    047745     ERG   = 000120 G
BIT13  = 020000 G   CMDSSV= 000012 G   C$DU  = 000053       DENSIT  002226 G   ERI   = 000113 G
BIT14  = 040000 G   CMDT    011036 G   C$EDIT= 000000       DEVERR  013176 G   ERLGER  014424 G
BIT15  = 100000 G   CMDTBL  024544       C$ERDF= 000055     DEVEXT  014302     ERL00   017137 G
BIT2   = 000004 G   CMD1    050532       C$ERHR= 000056     DEVFAT= 000001 G   ERL01   017174 G
BIT3   = 000010 G   CMD2    050540       C$ERRO= 000060     DFPTBL  002210 G   ERL02   017272 G
BIT4   = 000020 G   CMD3    050546       C$ERSF= 000054     DIAGMC= 000000     ERL03   017356 G
BIT5   = 000040 G   CMD4    050554       C$ERSO= 000057     DMPFLG  002233 G   ERL04   017445 G
BIT6   = 000100 G   CMD5    050562       C$ESCA= 000010     DPAT    050424     ERL05   017531 G
BIT7   = 000200 G   CMD6    050570       C$ESEG= 000005     DQCMD   030460 G   ERL06   017620 G
BIT8   = 000400 G   CMD7    050576       C$ESUB= 000003     DRBENO  005256 G   ERL07   017704 G
BIT9   = 001000 G   CMLSER  012244       C$ETST= 000001     DRBEN1  006316 G   ERL08   017734 G
BOE    = 000400 G   CMMDSQ  021366 G   C$EXIT= 000032     DRBEN2  007356 G   ERR   = 100000 G
BOTER    012002     CMP     = 000030 G   C$FREQ= 000101     DRBEN3  010416 G   ERRBLK  013174 G
BOTT     011176 G   CMPDAT  032732 G   C$FRME= 000100     DRBSTP= 000104 G   ERRDEC  031122 G
BRCNT    003554 G   CMPEL   013073       C$GETB= 000026     DRERFL= 000010 G   ERRDEI  031724 G
BUFADR   003600 G   CMPER   012515       C$GETW= 000027     DRINUS  003526 G   ERRDEL  032264 G
BUFDMP   034412     CMPERR  003622 G   C$GMAN= 000043     DROP  = 000010 G   ERREXT  031744
BUFDSC   026056     CMPPRI  014112       C$GPHR= 000042     DROPIT= 000200 G   ERRLOG= 040000 G
```

Symbol table

```
ERRMSG  013172 G      F$BGN = 000040        GWRBY3= 000140 G      IONORM= 000000 G      LUN3    003216 G
ERRNBR  013170 G      F$CLEA= 000007        GWRBY4= 000142 G      IOPDRE= 000003 G      L$ACP   002110 G
ERRTLY  032624 G      F$DU  = 000016        G$CNTO= 000200        IOSTAT  010732 G      L$APT   002036 G
ERRTYP  013166 G      F$END = 000041        G$DELM= 000372        IOTIME= 000004 G      L$AU    040276 G
ERR00   015730 G      F$HARD= 000004        G$DISP= 000003        ISR   = 000100 G      L$AUT   002070 G
ERR01   016000 G      F$HW  = 000013        G$EXCP= 000400        ISTART  037444        L$AUTO  037756 G
ERR02   016056 G      F$INIT= 000006        G$HILI= 000002        ITER    050505        L$CCP   002106 G
ERR03   016135 G      F$JMP = 000050        G$LOLI= 000001        ITERS   003576 G      L$CLEA  037760 G
ERR04   016213 G      F$MOD = 000000        G$NO  = 000000        ITMCNT= 000002 G      L$CO    002032 G
ERR05   016272 G      F$MSG = 000011        G$OFFS= 000400        ITMOFF= 000002 G      L$DEPO  002011 G
ERR06   016351 G      F$PROT= 000021        G$OFSI= 000376        ITRCNT= 000004 G      L$DESC  002142 G
ERR07   016421 G      F$PWR = 000017        G$PRMA= 000001        IVSER   012166        L$DESP  002076 G
ERR08   016461 G      F$RPT = 000012        G$PRMD= 000002        IVST1   011076 G      L$DEVP  002060 G
ERR09   016516 G      F$SEG = 000003        G$PRML= 000000        IVST2   011216 G      L$DISP  002124 G
ERR10   016557 G      F$SOFT= 000005        G$RADA= 000140        IVST3   011266 G      L$DLY   002116 G
ERR11   016611 G      F$SRV = 000010        G$RADB= 000000        IXE   = 004000 G      L$DTP   002040 G
ERR12   016637 G      F$SUB = 000002        G$RADD= 000040        I$AU  = 000041        L$DTYP  002034 G
ERR13   016675 G      F$SW  = 000014        G$RADL= 000120        I$AUTO= 000041        L$DU    040270 G
ERR14   016726 G      F$TEST= 000001        G$RADO= 000020        I$CLN = 000041        L$DUT   002072 G
ERR15   016754 G      GCMDST  025654        G$XFER= 000004        I$DU  = 000041        L$DVTY  002200 G
ERR16   017024 G      GCRDRP= 000072 G      G$YES = 000010        I$HRD = 000041        L$EF    002052 G
ERR17   017051 G      GCS   = 000210 G      HARD  = 000002 G      I$INIT= 000041        L$ENVI  002044 G
ERR18   017110 G      GCSCFL= 000020 G      HDATER  011634        I$MOD = 000041        L$ERFT  013166 G
ERS   = 000110 G      GCSEXT  027572        HDATT   011126 G      I$MSG = 000041        L$ETF   002102 G
ERTLY   032024        GCSHDL  027006 G      HDLEXT  021040        I$PROT= 000040        L$EXP3  002046 G
EVENT   003736 G      GCSREF  010736 G      HELP  = 000000        I$PTAB= 000041        L$EXP4  002064 G
EVL   = 000004 G      GCSRFL= 000040 G      HIADDR= 000002 G      I$PWR = 000041        L$EXP5  002066 G
EV.COR= 000150 G      GDCERR= 000062 G      HIBYTE= 177777 G      I$RPT = 000041        L$HARD  046500 G
EV.CTO= 000052 G      GHRDRD= 000054 G      HNDLRP  003556 G      I$SEG = 000041        L$HIME  002120 G
EV.DST= 000050 G      GHRDUA= 000056 G      HOE   = 100000 G      I$SETU= 000041        L$HPCP  002016 G
EV.HER= 000213 G      GHRDWR= 000052 G      HOUR    047453        I$SFT = 000041        L$HPTP  002022 G
EV.IDS= 000152 G      GMEDER= 000060 G      HOURS   002217 G      I$SRV = 000041        L$HW    002210 G
EV.LGP= 000010 G      GNOERR= 000074 G      HSTER   012216        I$SUB = 000041        L$ICP   002104 G
EV.SER= 000153 G      GO    = 000001 G      HSTIMO= 000000 G      I$TST = 000041        L$INIT  036644 G
EV.SRI= 000113 G      GOTHRD= 000066 G      HSTT    011136 G      J$JMP = 000167        L$LADP  002026 G
EV.SRT= 000053 G      GOTHUA= 000070 G      HUNGER  012277        KWCSR = 177546 G      L$LAST  110740 G
EV.URE= 000350 G      GOTHWR= 000064 G      IBE   = 010000 G      KWHDL   020702 G      L$LOAD  002100 G
EXC1A2  027576        GO1     040412        ICNT    050443        LEDB  = 000001 G      L$LUN   002074 G
EXC2A3  027620        GO2     042734        IDONE   026756        LEDER   012136        L$MREV  002050 G
EXC3A4  027642        GO3     043554        IDU   = 000040 G      LEDT    011256 G      L$NAME  002000 G
EXC4A1  027664        GO4     044414        IER   = 020000 G      LEOTFL= 000030 G      L$PRIO  002042 G
EXIT    026162        GO5     045046        ILCMD   026044        LF.CON= 000100 G      L$PROT  020640 G
EXTCLN  040004        GO6     045422        ILLCMD= 000007 G      LF.SNR= 000001 G      L$PRT   002112 G
EXTINT  037720        GO7     046224        ILOOP   026620        LF.SUC= 000200 G      L$REPP  002062 G
EX3REW  044142        GRDBY1= 000144 G      IMM   = 000200 G      LGPEL   013122        L$REV   002010 G
E$END = 002100        GRDBY2= 000146 G      IMMBIT= 000003 G      LGSTAT  030520 G      L$RPT   034536 G
E$LOAD= 000035        GRDBY3= 000150 G      INFORM  003734 G      LINE    020524 G      L$SOFT  046566 G
FAIL  = 000020 G      GRDBY4= 000152 G      INIT    025616        LOBYTE= 177776 G      L$SPC   002056 G
FLAG  = 040000 G      GSFTRD= 000050 G      INITER  012431        LOE   = 040000 G      L$SPCP  002020 G
FMTER   011762        GSFTWR= 000046 G      INITIT  037724 G      LOOP    026610        L$SPTP  002024 G
FMTT    011166 G      GSTERD= 000042 G      INT   = 000170 G      LOOPS   010744 G      L$STA   002030 G
FM.BAD= 000001 G      GSTEUA= 000044 G      INTDON= 000001 G      LOT   = 000010 G      L$SW    002216 G
FM.CNT= 000000 G      GSTEWR= 000040 G      INTERR= 000006 G      LUNFLG= 000026 G      L$TEST  002114 G
FM.TPE= 000005 G      GUNSTA  025714        INTTBL  026776        LUNSTP= 000224 G      L$TIML  002014 G
FORMAT  003732 G      GUS   = 000220 G      IOERTB  010756 G      LUNO    002322 G      L$UNIT  002012 G
F$AU  = 000015        GWRBY1= 000134 G      IOHUNG= 000002 G      LUN1    002546 G      L.BADR= 000030 G
F$AUTO= 000020        GWRBY2= 000136 G      IOICRD= 020000 G      LUN2    002772 G      L.CHVR= 000025 G
```

Symbol table

```
L.CNTI= 000014 G   MAXBUF= 020000 G   ONL    = 000140 G   PCFLAG  003674 G   P.CTPM= 000034 G
L.CNTO= 000061 G   MAXITR= 003720 G   ONLB   = 000100 G   PCKSIZ  010750 G   P.DVPM= 000034 G
L.CNT1= 000062 G   MD.ALL= 000002 G   ONLINE  025324       PCMDBF  003442 G   P.FLGS= 000011 G
L.CNT2= 000063 G   MD.CMP= 040000 G   OP.ABO= 000001 G   PDCERR= 000120 G   P.FMEM= 000044 G
L.CRF = 000000 G   MD.CSE= 020000 G   OP.ACC= 000020 G   PDMP    050221       P.FORM= 000040 G
L.CSVR= 000024 G   MD.DLE= 000200 G   OP.ACP= 000102 G   PDRECV  026352 G   P.HTMO= 000020 G
L.DFLG= 000054 G   MD.EXC= 000040 G   OP.AVA= 000100 G   PEDRP  = 000130 G   P.MEDI= 000034 G
L.DRVC= 000053 G   MD.IMM= 000100 G   OP.AVL= 000010 G   PHRDRD= 000112 G   P.MLUN= 000014 G
L.DRVS= 000064 G   MD.NXU= 000001 G   OP.CMP= 000040 G   PHRDUA= 000114 G   P.MOD = 000012 G
L.EVNT= 000012 G   MD.OBC= 000004 G   OP.DAP= 000013 G   PHRDWR= 000110 G   P.MXWR= 000044 G
L.FMVR= 000050 G   MD.REV= 000010 G   OP.END= 000200 G   PKPRNT  015416       P.NREC= 000050 G
L.FLGS= 000011 G   MD.RWD= 000002 G   OP.ERG= 000026 G   PMEDER= 000116 G   P.OPCD= 000010 G
L.FMT = 000010 G   MD.SEC= 001000 G   OP.ERS= 000022 G   PNOERR= 000132 G   P.OTRF= 000014 G
L.FMTD= 000042 G   MD.SER= 000400 G   OP.GCS= 000002 G   PNT    = 001000 G   P.POS = 000034 G
L.FSVR= 000051 G   MD.SPD= 000001 G   OP.GUS= 000003 G   POLER   012076       P.RCSK= 000014 G
L.GPCT= 000044 G   MD.SWP= 000004 G   OP.ONL= 000011 G   PQLT    011236 G   P.REDD= 000014 G
L.LBLK= 000060 G   MD.UNL= 000020 G   OP.RD  = 000041 G   PORTER  012317       P.SPED= 000042 G
L.LVL = 000042 G   MINBUF= 000024 G   OP.REP= 000045 G   POTHRD= 000124 G   P.STS = 000012 G
L.MLUN= 000026 G   MINITR= 000144 G   OP.SCC= 000004 G   POTHUA= 000126 G   P.TIME= 000024 G
L.OPFL= 000070 G   MINLIM  010754 G   OP.SUC= 000012 G   POTHWR= 000122 G   P.TMGC= 000020 G
L.PBLK= 000056 G   MINT    047535       OP.WR = 000042 G   PRDBY1= 000164 G   P.TMSK= 000020 G
L.PSTN= 000044 G   MINUTE  002220 G   OP.WTM= 000044 G   PRDBY2= 000166 G   P.TRBC= 000040 G
L.RTRY= 000043 G   MISSEQ= 000005 G   OWN    = 100000 G   PRDBY3= 000170 G   P.UHVR= 000053 G
L.RWST= 000066 G   MRETRY  003560 G   O$APTS= 000000       PRDBY4= 000172 G   P.UNFL= 000016 G
L.SEQN= 000006 G   MSCPVR= 000000 G   O$AU  = 000000       PRI    = 002000 G   P.UNIT= 000004 G
L.STI = 000050 G   MSGEXT  015610       O$BGNR= 000001      PRIERR  032714 G   P.UNTI= 000024 G
L.STS = 000052 G   MSGLEN= 177774 G   O$BGNS= 000001      PRIPCK  014040       P.USVR= 000052 G
L.TRK = 000055 G   MSKTST  031712       O$DU  = 000001      PRIOO  = 000000 G   P.VRSN= 000014 G
L.UHVR= 000041 G   MTBLOV= 000040 G   O$ERRT= 000001      PRIO1  = 000040 G   QCMD    023144 G
L.UNIT= 000004 G   N      = 000004 G   O$GNSW= 000001      PRIO2  = 000100 G   RANGEN  023010 G
L.UNTI= 000030 G   NCLK    017763 G   O$POIN= 000001      PRIO3  = 000140 G   RANWRD  003604 G
L.USVR= 000040 G   NCLKFL= 000002 G   O$SETU= 000001      PRIO4  = 000200 G   RAN1    003606 G
L.VSER= 000024 G   NOCL    050156       PASCNT  003700 G   PRIO5  = 000240 G   RAN2    003610 G
L10000  002114     NOCLK   020646 G   PASS1   003702 G   PRIO6  = 000300 G   RAN3    003612 G
L10001  002322     NOCLR   002232 G   PATCH   110614 G   PRIO7  = 000340 G   RD     = 000010 G
L10002  004422     NOTALY= 000004 G   PATCLR  033772       PRNTPA  002227 G   RDBUF   050614 G
L10003  020636     NRDY   = 000002 G   PATE    050313       PRPA    050016       RDR    = 000011 G
L10005  020700     NUL    = 000000 G   PATERN  002235 G   PRTCLR  026476 G   RDRENO  010456 G
L10006  021040     NULL    024614       PATGN1  022430      PRTDRV  026172 G   RDREN1  010542 G
L10007  036642     NULPAT= 000000 G   PATGN2  022444      PRTEXT  032362       RDREN2  010626 G
L10010  037754     NUPASS  037056       PATGN3  022456      PRTINT  026556 G   RDREN3  010712 G
L10011  037756     NURESP= 100000 G   PATGN4  022516      PSFTRD= 000106 G   RDSOER  002231 G
L10012  040010     OBCTHD  023436 G   PATGN5  022532      PSFTWR= 000104 G   RDSRG0  010416 G
L10013  040274     OBJECT  003676 G   PATGN6  022552      PSTERD= 000100 G   RDSRG1  010502 G
L10014  040302     OBJFDH= 000036 G   PATGN7  022566      PSTEUA= 000102 G   RDSRG2  010566 G
L10015  042624     OBJFDL= 000034 G   PATSAV= 000024 G   PSTEWR= 000076 G   RDSRG3  010652 G
L10016  043444     OBOFFH= 000006 G   PATTBL  022412 G   PWRBY1= 000154 G   RDTB   = 000002 G
L10017  044304     OBOFFL= 000004 G   PAT1   = 000001 G   PWRBY2= 000156 G   RDTER    012050
L10020  044736     OUDFLG= 000020 G   PAT2   = 000002 G   PWRBY3= 000160 G   RDTT    011226 G
L10021  045312     OFLER   011545       PAT3   = 000003 G   PWRBY4= 000162 G   READ    024620
L10022  046114     OFLT    011056 G   PAT4   = 000004 G   P.BCNT= 000014 G   RECCNT  003726 G
L10023  046474     OLD1    027124       PAT5   = 000005 G   P.BUFF= 000020 G   RESP    003572 G
L10024  046564     OLD2    027204       PAT6   = 000006 G   P.CHVR= 000051 G   RESPON  003552 G
L10025  050604     OLD3    027264       PAT7   = 000007 G   P.CMST= 000020 G   RETDON  030230 G
L10026  110744     OLD4    027344       PCBEND= 003522 G   P.CNTF= 000016 G   RETFLG= 000200 G
L10030  110750     ONE    = 000001 G   PCBSTP= 000014 G   P.CRF  = 000000 G   RETRY   021232 G
MANCNT  003562 G   ONEFIL= 000001       PCB3SP= 000044 G   P.CSVR= 000050 G   RET1    032150 G
```

Symbol table

```
RET2     032216 G    SERREC   002225 G    ST.ABO= 000002 G    TKUNT    046541      T$$SOF= 010025
REVBIT= 000001 G     SEXB   = 000004 G    ST.AVL= 000004 G    TLYEXT   031106      T$$SRV= 010006
REW    = 000160 G    SEXCNT   003544 G    ST.BOT= 000015 G    TMB    = 000010 G    T$$SW = 010001
REWIND   025520      SEXER    012114      ST.CMD= 000001 G    TMCNT    003730 G    T$$TES= 010023
RLSER    012452      SEXT     011246 G    ST.CMP= 000007 G    TMER     012022      T1       040304 G
RLST     011306 G    SFPTBL   002216 G    ST.CNT= 000012 G    TMT      011206 G    T1EXIT   042306
RNDBYT= 000000 G     SFTEXT   050604      ST.DAT= 000010 G    TPASS1= 000400 G    T1LEOT   042346
RNDITR= 000000 G     SFTEX1   047152      ST.DIA= 000037 G    TPEEL    012572      T1ONL    042332
RNUSAV= 000020 G     SFTEX2   047346      ST.DRV= 000013 G    TPEERL   011346      T1RD1    042552
ROLSAV= 000022 G     SFTEX3   047420      ST.FNT= 000014 G    TSPA     050264      T1RD2    042560
RSP8F0   004212 G    SFTEX4   047600      ST.HST= 000011 G    TSTGCR   020562 G    T1RD3    042566
RSP8F1   005252 G    SFTEX5   050014      ST.LED= 000023 G    TSTMSK   003616 G    T1RD4    042574
RSP8F2   006312 G    SFTEX6   050154      ST.MFE= 000005 G    TSTPE    020610 G    T1RD5    042602
RSP8F3   007352 G    SFTEX7   050262      ST.MSK= 000037 G    TSTSTP= 000006 G    T1RD6    042610
RSPCNT   003540 G    SFTEX8   050422      ST.OFL= 000003 G    TSTSUC   042324      T1REW    042340
RSPHDL   027706 G    SFTEX9   050530      ST.ONL= 000400 G    T$ARGC= 000004      T1SKD    042500
RS1    = 001233 G    SKD    = 000062 G    ST.POL= 000021 G    T$CODE= 026004      T1SKP    042362
RS2    = 007622 G    SKP    = 000060 G    ST.RDT= 000020 G    T$ERRN= 000000      T1SKP1   042456
RS3    = 000000 G    SKPTMK   025032      ST.SEX= 000022 G    T$EXCP= 000000      T1SKP2   042464
RTSPR1   042616      SKR    = 000061 G    ST.SUB= 000040 G    T$FLAG= 000041      T1SKR    042370
RTYEL    012733      SLTUSE= 000010 G     ST.SUC= 000000 G    T$FREE= 110750      T1SKR1   042426
RUNJAM   032500 G    SOER     050051      ST.TM = 000016 G    T$GMAN= 000000      T1SPC1   042376
RWI    = 000163 G    SOERRP   002230 G    ST.WPR= 000006 G    T$HILI= 065000      T1SPC2   042404
R10      003714 G    SOFT   = 000003 G    SUBCNT   003602 G    T$LAST= 000001      T1SPC3   042412
R11      003716 G    SOFTER   003570 G    SUBITR   023246 G    T$LOLI= 000000      T1SP01   042420
R12      003720 G    SPC    = 000050 G    SUBSEC   002222 G    T$LSYM= 010000      T1SP02   042434
R13      003722 G    SPCASC   011412      SUC    = 000150 G    T$LTNO= 000007      T1SP03   042442
R3SAVE   003740 G    SPCOBJ   025106      SUCCES= 000001 G    T$NEST= 177777      T1SPR1   042450
R4SAVE   003742 G    SPCREC   024764      SUNCHR   025422      T$NSO = 000000      T1SPR2   042472
R8       003710 G    SPO    = 000070 G    SUPRES   026110      T$NS1 = 000005      T1WR1    042506
R9       003712 G    SPR    = 000071 G    SUW    = 000155 G    T$PCNT= 000000      T1WR2    042514
SAERR    010752 G    SRER     050114      SVCGBL= 000000      T$PTAB= 010027      T1WR3    042522
SAVDIF   003614 G    START    036704      SVCINS= 000000      T$PTHV= 000001      T1WR4    042530
SCC    = 000230 G    START1   040304      SVCSUB= 000000      T$PTNU= 000001      T1WR5    042536
SCD    = 000052 G    START2   042626      SVCTAG= 000000      T$SAVL= 177777      T1WR6    042544
SCHED    021042 G    START3   043446      SVCTST= 000000      T$SEGL= 177777      T1WTM    042354
SCNTCH   025742      START4   044306      SYSFAT= 000000 G    T$SIZE= 000004      T2       042626 G
SCR    = 000051 G    START5   044740      S$LSYM= 010000      T$SUBN= 000000      T2END  = 004716 G
SDATER   011654      START6   045314      S1     = 004000 G    T$TAGL= 177777      T2EXIT   043356
SDATT    011316 G    START7   046116      TALLY    030570      T$TAGN= 010001      T2LEOT   043414
SDSAVE   033714 G    STATER   012476      TBLEND= 003650 G    T$TEMP= 000000      T2RD     043422
SDSTUP   033696 G    STATUS= 000004 G     TCNTFL= 000004 G    T$TEST= 000007      T2REW    043400
SECOND   002221 G    STAT01   035770      TEMP     003550 G    T$TSTM= 177[illegible]      T2SKD    043430
SECRNS   003724 G    STAT02   036036      TESTPA   002234 G    T$TSTS= 000[illegible]      T2SPO    043436
SED1   = 000174 G    STAT03   036061      TF.BLK= 000010 G    T$$AU = 010014      T2WRT    043406
SED2   = 000176 G    STAT04   036103      TF.GCR= 000004 G    T$$AUT= 010011      T3       043446 G
SED3   = 000200 G    STAT05   036157      TF.PE = 000002 G    T$$CLE= 010012      T3EXIT   044224
SEED1  = 000202 G    STAT06   036215      TF.800= 000001 G    T$$DAT= 010030      T3RD     044270
SEED2  = 000204 G    STAT07   036256      TIME     020037 G    T$$DU = 010013      T3REW    044246
SEED3  = 000206 G    STAT08   036314      TIMER    010742 G    T$$HAR= 010024      T3SPO    044276
SELDAT   022246 G    STAT09   036360      TIMERR   012343      T$$HW = 010000      T3WRT    044254
SELREC   022712 G    STAT10   036424      TKERR    026750      T$$INI= 010010      T3WTM    044262
SEQER    012400 G    STAT11   036462      TKINIT= 111400 G    T$$MSG= 010003      T4       044306 G
SERC     047637      STAT12   036534      TKIP   = 000000 G    T$$PC = 000001      T4EXIT   044702
SERCOR   002224 G    STAT13   036606      TKIPAD   046524      T$$PRO= 010004      T4REW    044714
SEREXC= 000002 G     STFPCK   024342 G    TKSA   = 000002 G    T$$PTA= 010027      T4WRT    044722
SERR     047703      STINIT   036644 G    TKUNIT= 000004 G    T$$RPT= 010007      T4WTM    044730
```

Symbol table

```
T5      044740 G        T7BRFL= 000001 G        UCDSRG= 000220 G        UNKERL  011356          WPRER   011607
T5CMP   002236 G        T7CMD1  002240          UDROP   003704 G        UNLBIT= 000004 G        WPRT    011106 G
T5CP    050331          T7CMD2  002246          UEOT    003706 G        UNTEOT  020253 G        WR    = 000020 G
T5EXIT  045264          T7CMD3  002254          UF.CMR= 000001 G        UNTLOT  020307 G        WRITE   024662
T5RD    045304          T7CMD4  002262          UF.CMW= 000002 G        UNTSTP= 000002 G        WRKMSK  003620 G
T5REW   045276          T7CMD5  002270          UF.RMV= 000200 G        URBEND= 000212 G        WRTBUF  070614 G
T6      045314 G        T7CMD6  002276          UF.VSS= 000040 G        URDEND= 000216 G        WTAPMK  025162
T6EXIT  046034          T7CMD7  002304          UF.VSU= 020000 G        URDSRG= 000214 G        WTM   = 000100 G
T6RD    046100          T7END   002312          UF.WPH= 020000 G        UREEL   013022          XFERST= 000010 G
T6REW   046056          T7EXIT  046464          UF.WPS= 010000 G        URSPBF= 000210 G        X$ALWA= 000000
T6SKD   046106          T7TBL   002232 G        UNDROP= 000032 G        UWEEL   013046          X$FALS= 000040
T6WRT   046064          UAM   = 000200 G        UNJAM   033420 G        WPRB  = 000020 G        X$OFFS= 000400
T6WTM   046072          UCDEND= 000222 G        UNKEL   012675          WPRBIT= 000005 G        X$TRUE= 000020
T7      046116 G

. ABS.  110750    000   (RW,I,GBL,ABS,OVR)
        000000    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 329
Work  file writes: 327
Size of work file: 35360 Words  ( 139 Pages)
Size of core pool: 19714 Words  ( 75 Pages)
Operating  system: RSX-11M/PLUS (Under VAX/VMS)

Elapsed time: 00:22:47.64
CZTU1A.BIN,CZTU1A.LST/-SP=SVC40R.MLB/ML,CZTU1A
```