

TS11

TS11 DATA RELIAB  
CZTSHD0

COPYRIGHT (c) 1978-84  
AH-E455D-MC  
FICHE 01 OF 01

JUL 1984  
digital  
Made In USA

This image shows a microfiche card with a grid of frames. The frames contain data, likely in a tabular format, but the text is extremely faint and difficult to read. The card is dark blue with a grid of small, light-colored frames. The data within the frames appears to be organized in rows and columns, with some frames containing what looks like a header or title. The overall appearance is that of a standard microfiche card used for data storage and retrieval.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39

.REF1 \

IDENTIFICATION

PRODUCT CODE: AC E454D MC

PRODUCT NAME: CZTSHD0 T511 DATA RELIAB

PRODUCT DATE: 15 MARCH 1984

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: J. MITT

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE OR EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1978, 1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DEC	DECTAPE	

40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91

USER DOCUMENTATION

USER DOCUMENTATION TABLE OF CONTENTS

GLOSSARY

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

- 1.1.1 FUNCTIONAL DESCRIPTION
- 1.1.2 STRUCTURE OF PROGRAM
- 1.1.3 MEMORY MAP
- 1.1.4 DIAGNOSTIC INFORMATION
  - 1.1.4.1 SCOPE
  - 1.1.4.2 ERROR RECOVERY
  - 1.1.4.3 WRITE ERROR RECOVERY
    - 1.1.4.3.1 MEDIA/OPERATIONAL SELECTIVE WRITE-ERROR RECOVERY
    - 1.1.4.3.2 OPERATIONAL WRITE ERROR
  - 1.1.4.4 DIAGNOSTIC TIMING ADJUSTMENT

1.2 SYSTEM REQUIREMENTS

- 1.2.1 HARDWARE REQUIREMENTS
- 1.2.2 SOFTWARE REQUIREMENTS

1.3 RELATED DOCUMENTS AND STANDARDS

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

1.5 ASSUMPTIONS

1.6 DIAGNOSTIC HISTORY

2.0 OPERATING INSTRUCTIONS

2.1 HARDWARE PARAMETERS

2.2 SOFTWARE PARAMETERS

- 2.2.1 T504 COMMAND LIST
- 2.2.2 DATA PATTERNS

92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104

- 2.3 EXAMPLES OF SOFTWARE PARAMETER DIALOGUE
  - 2.3.1 BASIC FUNCTION AND DATA RELIABILITY WITH ALL ERROR REPORTING ENABLED
  - 2.3.2 SCOPE LOOP SET UP IN BASIC FUNCTIONS
  - 2.3.3 SCOPE LOOP SET UP IN DATA RELIABILITY
  
- 2.4 EXECUTION TIMES
  - 2.4.1 SYSTEM CONFIGURATION
  - 2.4.2 TEST EXECUTION TIMES



105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144

3.0 ERROR INFORMATION

3.1 ERROR REPORTING

- 3.1.1 ERROR #1 COMMAND PACKET ADDRESS IS NOT ON A M
- 3.1.2 ERROR #2 TSO4 NOT READY
- 3.1.3 ERROR #3 NO RESPONSE ERRORS
- 3.1.4 ERROR #4 NO INTERRUPT ERROR
- 3.1.5 SPECIAL CONDITION ERRORS
  - 3.1.5.1 ERROR #5 TCC0, UNDEFINED SPECIAL COND
  - 3.1.5.2 ERROR #6 TCC1, ATTENTION CONDITION
  - 3.1.5.3 ERROR #7 TCC2, TAPE STATUS ALERT
  - 3.1.5.4 ERROR #8 TCC3, FUNCTION REJECT
  - 3.1.5.5 ERROR #9 - TCC4, RECOVERABLE ERROR
  - 3.1.5.6 ERROR #10 TCC5, RECOVERABLE ERROR
  - 3.1.5.7 ERROR #11 TCC6, UNRECOVERABLE ERROR
  - 3.1.5.8 ERROR #12- TCC7, FATAL SUBSYSTEM ERROR
- 3.1.6 ERROR #13 RFC NON ZERO ERROR
- 3.1.7 ERROR #14 - RETRY LIMIT EXCEEDED
- 3.1.8 ERROR #15 TOO MANY INTERRUPTS
- 3.1.9 ERROR #16 CAPSTAN RUNAWAY
- 3.1.10 ERROR #17 DATA COMPARE ERRORS

3.2 ERROR HALTS

4.0 PERFORMANCE REPORT

5.0 TEST SUMMARIES

- 5.1 TEST 1 BASIC FUNCTIONS
- 5.2 TEST 2 DATA RELIABILITY
- 5.3 TEST 3 WRITE COMPATABILITY/WRITE UTILITY
- 5.4 TEST 4 READ COMPATABILITY/READ UTILITY
- 5.5 TEST 5 EXECUTE OPERATOR SELECTED COMMAND SEQUENCE

145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156

6.0 DEVICE INFORMATION

- 6.1 GENERAL
- 6.2 UNIBUS INTERFACE SPECIFICATIONS
- 6.3 BIT DEFINITIONS FOR TS11/TS04 REGISTERS
  - 6.3.1 TS11/TS04 REGISTER SUMMARY
  - 6.3.2 TS11 STATUS REGISTER (TSSR)
  - 6.3.3 EXTENDED STATUS REGISTER (XSTAT0)
  - 6.3.4 EXTENDED STATUS REGISTER 1 (XSTAT1)
  - 6.3.5 EXTENDED STATUS REGISTER 2 (XSTAT2)
  - 6.3.6 EXTENDED STATUS REGISTER 3 (XSTAT3)

157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194

GLOSSARY

ACT	AUTOMATED COMPUTER TEST SYSTEM
APT	AUTOMATED PRODUCT TEST SYSTEM
BYTE/RECORD/FILE COUNT BRF	IS STORED IN THE 4TH WORD OF THE COMMAND PACKET AND IT'S USE BY THE TS04 DEPENDS ON THE TYPE OF COMMAND.
CMD	TS04 COMMAND (SEE 2.3.14.1 FOR LIST OF COMMANDS)
COMMAND PACKET CMDPK	FOUR WORD PACKET IN THE CPU MEMORY WHICH CONTAINS ALL INFORMATION NEEDED BY THE TS04 TO EXECUTE A COMMAND.
EXTENDED STATUS	FOUR WORDS OF TS04 STATUS WHICH ARE TRANSFERRED AS PART OF THE MESSAGE PACKET AT THE COMPLETION OF A COMMAND.
MESSAGE PACKET	SEVEN WORD PACKET IN THE CPU MEMORY INTO WHICH THE TS04 STORES STATUS AT THE COMPLETION OF A COMMAND.
PC	PROGRAM COUNTER
PSW	PROCESSOR STATUS WORD
RESIDUAL FRAME COUNT RFC	THIS COUNT IS PART OF THE MESSAGE PACKET AND CONTAINS THE NUMBER OF BYTES/RECORDS /FILES REMAINING TO BE PROCESSED AT THE COMPLETION OF A COMMAND.
SPECIAL CONDITION SPEC COND	TS04 BIT1 WHEN SET, INDICATES THAT THE LAST COMMAND DID NOT COMPLETE WITH- OUT INCIDFNT.
TERMINATION CLASS CODE	THREE BIT CODE IN THE TSSR WHICH INDI

195	TCC	CATES THE TYPE OF COMMAND TERMINATION.
196		
197	TSBA	TAPE SYSTEM BUS ADDRESS REGISTER.
198		
199	TSDB	TAPE SYSTEM DATA BUFFER REGISTER.
200		
201	TSSR	TAPE SYSTEM STATUS REGISTER.
202		
203	XST0	EXTENDED STATUS REGISTER 0
204		
205	XST1	EXTENDED STATUS REGISTER 1
206		
207	XST2	EXTENDED STATUS REGISTER 2
208		
209	XST3	EXTENDED STATUS REGISTER 3
210		
211	XXDP.	XXDP. IS A 'CATCH ALL' NAME FOR A GROUP OF PDP 1
212		DIAGNOSTIC PACKAGES AVAILABLE ON MULTIMEDIA.
213		



214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

1.1.1 FUNCTIONAL DESCRIPTION

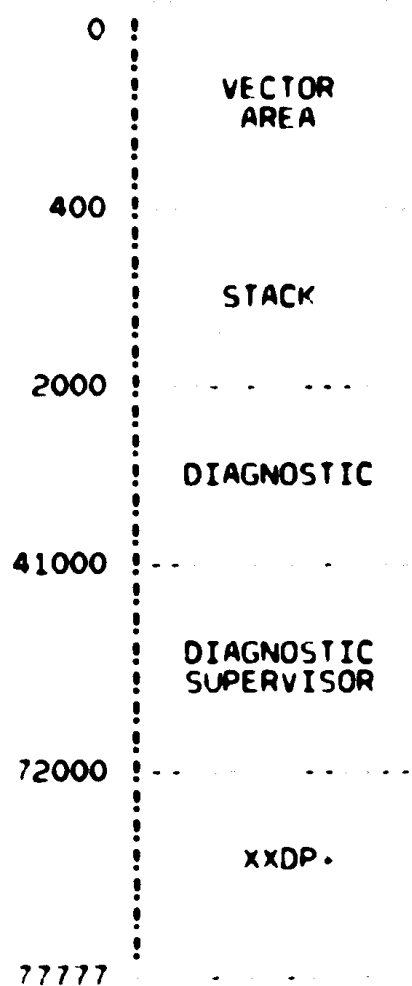
THIS PROGRAM CAN BE USED AS A BASIC FUNCTION TEST, A DATA RELIABILITY TEST, A COMPATABILITY TEST, OR TO EXECUTE A SEQUENCE OF OPERATOR SELECTED COMMANDS.

1.1.2 STRUCTURE OF PROGRAM

THIS DIAGNOSTIC IS A SINGLE PROGRAM FROM THE STANDPOINT OF THE DIAGNOSTIC USER, BUT IT CONTAINS A CONTROL MODULE RELEASED INDEPENDENTLY AS A DIAGNOSTIC SUPERVISOR.

1.1.3 MEMORY MAP

237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279



FREE MEMO SPACE FOR WR/RD BFRS OR OTHER PUROSES  
IS ALLOCATED BY THE SUPERVISOR ON REQUEST OR CHOOSEN  
BY PROGRAMMER TO RESIDE BETWEEN THE DIAG AND THE  
SUPERVISOR.

280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335

1.1.4 DIAGNOSTIC INFORMATION

1.1.4.1 SCOPE

THIS DIAGNOSTIC CAN TEST UP TO 4 UNITS SIMULTANEOUSLY. THE 4 UNITS ARE ASSIGNED LOGICAL UNIT NUMBERS 0 3 BY THE DIAGNOSTIC.

THERE ARE 5 TESTS IN THIS PROGRAM:

- TEST 1 BASIC FUNCTIONS.
- TEST 2 DATA RELIABILITY.
- TEST 3 - WRITE COMPATABILITY/WRITE UTILITY.
- TEST 4 - READ COMPATABILITY/READ UTILITY.
- TEST 5 - OPERATOR SELECTED SEQUENCE UTILITY.

1.1.4.2 ERROR RECOVERY

ERROR RECOVERY IS PERFORMED ON READ, WRITE AND WRITE TAPE MARK ERRORS UNLESS RECOVERY IS INHIBITED BY THE OPERATOR. THE READ FORWARD/READ REVERSE RETRY LIMIT IS 16 (8 IN THE SAME DIRECTION AND 8 IN THE OPPOSITE DIRECTION). FOR MORE INFORMATION ON ERROR RECOVER PROCEDURES, SEE SECTION 3.0 (ERROR REPORTING).

1.1.4.3 WRITE ERROR RECOVERY

THERE ARE 2 DISTINCT, SELECTABLE WRITE ERROR RECOVERY ALGORITHMS:  
1. MEDIA/OPERATIONAL SELECTIVE ALGORITHM  
2. OPERATIONAL ALGORITHM

BY DEFAULT THE DIAGNOSTIC SELECTS THE FIRST ALGORITHM TO DISCERN MEDIA RELATED WRITE ERRORS FROM OPERATIONAL ONES.

TO SELECT THE SECOND ALGORITHM:

- ANSWER 'Y' TO CHANGE SW (L) ?
- ANSWER 'N' TO BAD TAPE SPOT DETECTION (L) Y ?

WHEN ERROR RECOVERY IS INHIBITED, THE LATTER QUESTION IS NOT ASKED AND BOTH ALGORITHMS ARE BYPASSED.

1.1.4.3.1 MEDIA/OPERATIONAL SELECTIVE WRITE ERROR RECOVERY ALGORITHM

SCOPE

THE ALGORITHM DISCERNs MEDIA RELATED WRITE ERRORS FROM OPERATIONAL ONES.

ALGORITHM

A WRITE RETRY SUBROUTINE IS CALLED BY THE RECOVERABLE ERROR SUBROUTINE ENTERED UPON DETECTION OF A WRITE RECOVERABLE ERROR. THE WRITE RETRY SUBROUTINE REWRITES RECORD IN SAME SPOT ON TAPE; REPEAT 4 TIMES. IF ALL 4 REPEATS ARE GOOD, RECORD IS CONSIDERED AS RECOVERED AND A RECOVERABLE WRITE ERROR IS LOGGED AT THAT RECORD NUMBER. IF ANY OF THE 4 REPEAT FAILS, ERASE BAD RECORD, LOGG SUSPECTED

336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391

BAD SPOT AT THAT RECORD NUMBER, RETRY AGAIN 3 INCHES FURTHER DOWN TAPE.  
RETRY 4 TIMES, UP TO 4 REPEATS EACH.  
IF RECORD CANNOT BE WRITTEN WITHOUT RECOVERABLE ERROR AFTER 4 RETRIES,  
ERASE RECORD, REPORT RETRY FAILED ON BAD SPOT.  
THE RECOVERABLE ERROR SUBROUTINE THEN CONTINUES TO CALL THE WRITE  
RETRY SUBROUTINE, WHICH REISSUES THE GROUP OF 4 RETRIES,  
UNTIL THE RECORD IS RECOVERED OR 20 BAD SPOTS HAVE BEEN LOGGED .

TWENTY (20) BAD SPOTS MAXIMUM ARE ALLOWED PER TAPE PASS.  
WHEN 20 BAD SPOTS HAVE BEEN LOGGED, ON SAME RECORD NUMBER OR NOT,  
TAPE IS CONSIDERED DEFECTIVE: A BAD TAPE OVERFLOW MESSAGE IS PRINTED  
AND UNIT IS REWOUND, THEN DROPPED.

DURING THE RECOVERY PROCESS, IT IS NECESSARY TO PERFORM SEVERAL TAPE  
POSITION OPERATIONS: SPACE REVERSE, ERASE. IF A POSITION ERROR STATUS  
IS DETECTED DURING THOSE OPERATIONS, THEN THE RECOVERY ATTEMPT IS ABORTE  
AN APPROPRIATE UNRECOVERABLE MESSAGE IS PRINTED AND UNIT IS DROPPED.

ALL BADLY WRITTEN RECORDS FLAGGED WITH RECOVERABLE ERRORS ARE ERASED  
UNTIL RECOVERED, INCLUDING THE RECORD AT THE 20TH BAD SPOT,  
SO THAT ALL RECORDS LEFT ON TAPE ARE GOOD WRITTEN RECORDS.  
BAD SPOTS ARE ERASED, WITH ERASE GAPS FROM 3 TO 12 INCHES PER RETRY GRO  
UP TO 20 FEET OF ERASE GAP COULD RESULT WHEN RETRYING TO RECOVER  
A SINGLE RECORD, IF NO BAD SPOT WERE PREVIOUSLY DETECTED.  
THAT LONG STRETCH OF BAD TAPE WOULD THEN BE FLAGGED WITH 20  
BAD SPOTS AT SAME RECORD NUMBER AND THE TAPE CONSIDERED DEFECTIVE.

#### EAD SPOTS REPORTS

IF THE PRINT OF RECOVERABLE ERRORS IS ENABLED, THE BAD SPOTS ON TAPE ARE  
IDENTIFIED AS THEY ARE DETECTED. SINCE THE BAD RECORDS ARE ERASED UNTIL  
THE BAD SPOTS ACTUALLY PRECEDES THE RECORD NUMBER THAT IDENTIFIES THEM.  
THE NUMBER OF REPEATS AND RETRIES ATTEMPTED IS PRINTED, FROM WHICH THE  
LENGTH OF ERASE GAPS CAN BE DETERMINED: APPROXIMATELY 3 INCHES PER RETR

THE STATISTICAL REPORT PRINTED AT THE END OF TEST 2 OR UPON A 'PRINT' RE  
CONTAINS A SUMMARY OF THE BAD SPOTS LOGGED ON THE CURRENT TAPE PASS.  
IN THAT REPORT, ALL COUNTS ARE CUMULATIVE FROM PASS TO PASS, EXCEPT FOR  
THE NUMBER OF BAD SPOTS: IT RELATES TO A "TAPE PASS" ONLY.  
FOR THIS PURPOSE, A "TAPE PASS" IS A WRITE PASS FROM BOT TO EOT, OR FROM  
BOT TO WHERE THE DIAGNOSTIC IS HALTED BEFORE REACHING EOT.  
A PASS IS DEFINED BY THE SUPERVISOR AS A RUN THROUGH ALL THE TESTS REQUE  
ON ALL UNITS SELECTED. THOSE PASSES ARE IDENTIFIED AS "PASS" AND "EOP".

THE NUMBER OF WRITE RETRIES, CUMULATIVE FROM PASS TO PASS, IS A GLOBAL  
COUNT OF HOW MANY TIMES THE GROUP OF 4 RETRIES HAS BEEN CALLED.

THE NUMBER OF WRITE RECOVERABLE ERRORS EXCLUDES BAD TAPE SPOTS  
AND REFLECTS THE SPECIFICATIONS OF THE HARDWARE UNDER TEST.  
PER TAPE PASS, THE NUMBER OF WRITE RETRIES EQUALS THE SUM OF THE  
NUMBER OF RECOVERABLE WRITE ERRORS AND BAD SPOTS, MOST OF THE TIME.

TO CLEAR CUMULATIVE COUNTS, ANSWER 'Y' TO: CLEAR COUNTERS (L) Y ?.  
BAD TAPE SPOTS COUNT IS CLEARED WHEN WRITING FROM BOT.

IF TEST 2 IS HALTED, THEN RESTARTED OR CONTINUED, THE RECORD COUNT

ML

392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447

IS RESET TO ZERO AND THE BAD SPOT ID SHALL FOLLOW THAT RESET COUNT.

SINCE ALL WRITTEN RECORDS ARE KNOWN GOOD, THE READ ERRORS CAN BE ATTRIBUTED TO TRANSIENT NOISE, TRANSIENT ELECTRICAL MALFUNCTIONS, OR CONTAMINANTS ON TAPE AS OPPOSED TO TAPE DEFECTS.

THE SAME RECORDS MUST BE WRITTEN FROM TAPE PASS TO TAPE PASS FOR THE BAD SPOTS ID TO REMAIN CONSISTENT IN THOSE TAPE PASSES.

EXAMPLE OF A TAPE PASS PRINTS:

CZTSH SFT ERR 00009 ON UNIT 00 TST 002 SUB 000 PC: 012100  
RECOVERABLE ERROR  
WRT CMD FAILED - UNIT 0 PASS: 1 RECORD: 6  
PREVIOUS CMD WAS WRT  
CMDPKT TSBA RFC TSSR TCC  
100205 002406 000000 100210 4  
026600  
000000  
003107  
XST0 XST1 XST2 XST3  
000350 000002 100400 000000  
SUSPECT BAD SPOT AFTER 1 RETRY, 2 REPEAT  
SUSPECT BAD SPOT AFTER 2 RETRY, 1 REPEAT  
SUSPECT BAD SPOT AFTER 3 RETRY, 1 REPEAT  
SUSPECT BAD SPOT AFTER 4 RETRY, 3 REPEAT  
RETRY FAILED ON BAD SPOT...ERASED!  
SUSPECT BAD SPOT AFTER 1 RETRY, 1 REPEAT  
SUSPECT BAD SPOT AFTER 2 RETRY, 1 REPEAT

CZTSH SFT ERR 00009 ON UNIT 00 TST 002 SUB 000 PC: 012100  
RECOVERABLE ERROR  
WRT CMD FAILED UNIT 0 PASS: 1 RECORD:10210  
PREVIOUS CMD WAS WRT  
CMDPKT TSBA RFC TSSR TCC  
100205 002406 000000 100210 4  
026600  
000000  
004000  
XST0 XST1 XST2 XST3  
000350 000002 100010 000000  
RECOVERED ON RETRY # 1  
↑C  
DR>PRI

UNIT 0 PASS: 1 RECORD:10210  
BYTES WRITTEN 0,272,279,691  
BYTES READ REV 0,301,123,654  
BYTES READ REV 0,301,120,381  
RECOVERABLE ERRORS WRT RDR RDF  
UNRECOVERABLE ERRORS 0 0 0  
WRITE RETRIES 3

2 BAD SPOTS THIS TAPE PASS PRECEDING RECORD #:

SPEC COND    6    6  
              HARD    FATAL    COMPARE  
              2        0        0        0  
DR>

## THIS EXAMPLE SHOWS:

RECORD 6 RECOVERED ON 2ND RETRY GROUP  
THE 2 BAD SPOTS RESIDE IN A 18 INCH ERASE GAP BETWEEN RECORDS 5  
RECORD 10210 RECOVERED ON 1ST RETRY OF 4 GOOD REPEATS  
3 WRITE GROUP RETRIES ATTEMPTED, RESULTING IN:  
  1 RECOVERABLE WRT ERR FROM RECORD 10210  
  2 BAD SPOTS BETWEEN RECORDS 5 AND 6

## 1.1.4.3.2 OPERATIONAL WRITE ERROR RECOVERY ALGORITHM

WHEN THIS ALGORITHM IS SELECTED, THE TS11 WRITE RETRY COMMAND IS ISSUED UP TO 16 TIMES OR UNTIL RECORD IS RECOVERED, ON A WRITE RECOVERABLE ERROR. THE WRITE RETRY COMMAND CONSISTS OF A SPACE REVERSE OVER THE BAD RECORD, THEN AN ERASE OF 3 INCHES OF TAPE AND REWRITE OF THE RECORD. THAT COMPOSITE COMMAND DOES NOT ALLOW TO DETECT BAD SPOTS ON TAPE. THEREFORE NO BAD TAPE SPOTS STATUS IS PRINTED.

IF RECORD CANNOT BE RECOVERED AFTER 16 WRITE RETRY COMMANDS, A RETRY LIMIT EXCEEDED IS FLAGGED AND UNIT IS DROPPED.

## 1.1.4.4 DIAGNOSTIC TIMING ADJUSTMENT

A NUMBER OF SUPERVISOR TIMING DELAYS MACROS, KNOWN AS WATCH DOG DELAYS, ARE CALLED BY THE DIAGNOSTIC TO WAIT FOR VARIOUS COMMANDS COMPLETION. THESE DELAYS ARE NOT CALIBRATED AND SIMPLY EXPANDS INTO AN INLINE NESTED LOOP PAIR. THE COUNT FOR THE OUTER LOOP COMES FROM THE VARIABLE ARGUMENT SUPPLIED BY THE DELAY CALLS. THE COUNT FOR THE INNER LOOP COMES FROM THE FIXED "HEADER ELEMENT 'L\$DLY". AS THE DIAGNOSTIC IS RUN ON DIFFERENT CPU'S, THESE DELAYS WILL VARY IN LENGTH WITH MEMORY SPEED.

IF TIME-OUT OCCURS WHEN NO APPARENT MALFUNCTIONS IN THE TAPE UNIT IS EVIDENT, ALL TIMINGS OF THE DIAGNOSTIC MAY BE ADJUSTED TO MATCH MEMORY SPEED AND NOT RESULT IN TIME OUTS, BY PATCHING THAT FIXED DELAY ELEMENT "L\$DLY".

A PRESET COUNT OF 500 RESIDES AT 'L\$DLY' IN LOCATION 2116 OF THE 'HEADER' SECTION.

## 1.2 SYSTEM REQUIREMENTS

## 1.2.1 HARDWARE REQUIREMENTS

448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503

508  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515

PDP-11 PROCESSOR WITH 16K OR MORE OF MEMORY  
CONSOLE DEVICE (LA30,LA36,VT50,ETC.)  
PROGRAM LOAD DEVICE

1.2.2 SOFTWARE REQUIREMENTS

DIAGNOSTIC SUPERVISOR



514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546

1.3 RELATED DOCUMENTS AND STANDARDS  
.....

XXDP. USERS MANUAL MD-11-CMQUS  
DIAGNOSTIC SUPERVISOR PROGRAM LISTING  
PDP-11 DIAGNOSTIC SUPERVISOR INTERFACE SPECIFICATION.  
PDP-11 DIAGNOSTIC SUPERVISOR PROGRAMMER'S GUIDE.  
TS11/TS04 PROGRAMMING SPECIFICATION.  
TS11/TS04 ENGINEERING SPECIFICATION.  
TS11/TS04 COMMAND PACKET SPECIFICATION.

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES  
.....

ORDER OF MOST CPU DIAGNOSTIC USAGE:

- 1) CONTROL LOGIC PROGRAM ALL TESTS.
- 2) DATA RELIABILITY PROGRAM:
  - A) BASIC FUNCTION TEST.
  - B) DATA RELIABILITY TEST.

1.5 ASSUMPTIONS  
.....

THE HARDWARE OTHER THAN THE SUBSYSTEM BEING TESTED IS ASSUMED TO WORK PROPERLY. FALSE ERRORS MAY BE REPORTED IF THE PROCESSOR, MEMORY, ETC., DO NOT FUNCTION PROPERLY.

1.6 DIAGNOSTIC HISTORY

547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602

- REVISION A - OCT 1978
  - ORIGINAL RELEASE
  
- REVISION B FEB 1979
  - CORRECTED END OF TAPE PROBLEMS IN TESTS 3 5.
  - CHANGED DEFAULT VECTOR ADDRESS FROM 150 TO 224.
  - DECREASED MAXIMUM RECORD LENGTH FROM 4096 TO 2048 BYTES.
  
- REVISION B - AUG 1979
  - DO NOT PRINT RECOVERABLE ERRORS UNLESS REQUESTED BY OPERATOR
  - WARN OPERATOR OF UNIT(S) BEING NOT READY OR OFF-LINE.
  - DROP UNIT(S) LEFT NOT READY OR OFF-LINE FOR 3.5 MINUTES.
  - IMPROVE BEHAVIOR AT EOT
    - IN TEST 2, FREEZE UNITS REACHING EOT UNTIL OTHERS
    - CATCH-UP INSTEAD OF ALLOWING THEM TO SHUTTLE AT EOT
    - WHEN ALL UNITS REACH EOT, WRITE ONE RECORD BEYOND EOT.
    - READ REV THAT EXTRA RECORD TO POSITION TAPE
    - SO THAT THE NEXT COMMAND REQUESTED CAN BE EXECUTED.
    - THAT EXTRA RECORD SHALL LEAVE A CLEAN IRG GAP AND A VALID
    - RECORD TO READ WHEN SHORTER READ STOP DISTANCE MIGHT CAUSE
    - UNIT TO FLAG EOT ON THAT EXTRA RECORD INSTEAD OF THE
    - PREVIOUS ONE. THIS SHOULD ELIMINATE MANY READ ERRORS AT
    - EOT AND TAPES RUNNING OFF THE WHEELS.
  - WRITE RECORD COUNT ON TAPE.
  - PRINT RECORD COUNT READ FROM TAPE IN READ ERROR PRINTS TO
  - INDICATE IF POSITION WAS LOST.

\* CAUTION \*

INTERPRET THAT 'RECORD READ' COUNT WITH CAUTION.  
IF VERY DIFFERENT FROM RECORD COUNT TRACKED BY THE DIAGNOST  
POSITION IS NOT NECESSARELY LOST. ERRORS IN READING THAT  
RECORD MIGHT HAVE CAUSED RECORD COUNT TO BE ERRONEOUSLY  
READ FROM TAPE.  
IN TEST 2, IF DIAGNOSTIC IS RESTARTED OR CONTINUED, RECORD  
IS RESET TO ZERO ALTHOUGH TAPE WAS NOT REWOUND. THIS IS  
NECESSARY BECAUSE THERE IS NO ACCURATE WAY TO DETERMINE  
ON WHAT RECORD COUNT OF WHAT UNIT THE DIAGNOSTIC WAS HALTED  
BEFORE RESTARTING OR CONTINUING.  
IT IS SUGGESTED THAT A "PRINT" BE REQUESTED WHEN HALTING DI  
TO GET A PRINT OF THE RECORD COUNT WHEN HALTED.

VERIFY RECORD OF 4000 BYTES INSTEAD OF 22 BYTES.  
WHEN COMPARING DATA, CHECK AND PRINT IF NO DATA WAS READ  
OR RECORD WAS LONGER THAN EXPECTED.  
FREEZE TSSR REG WHEN A COMMAND IS COMPLETED TO AVOID DIFFERE  
BETWEEN TSSR AND TCC FETCHED AT DIFFERENT TIMES.  
WHEN DROPPING A UNIT, FLAG SECOND PRINT OF EXTENDED STATUS  
THE RESULT OF A GET STATUS COMMAND.  
WAIT FOR SSR UP BEFORE PRINTING THAT STATUS.  
ADJUST "PASS" COUNT OF DIAG TO MATCH 'EOP' PASS COUNT OF SUP  
INCREASE NUMBER OF SELECTABLE COMMANDS IN TEST 5 FROM  
4 TO 7. DEFAULT COMMAND 6 IS NOW REWIND.  
CONVERT DIAG TO REV C OF SUPERVISOR.

E?

603  
604  
605  
606  
607  
608  
609  
610  
611  
612

ADD SEVERAL SECTIONS:  
PROTECT TABLE  
AUTO-DROP CODE  
HARD CODED PARAMETER TABLE  
REVISION C OCT 79  
ADD MEDIA/OPERATIONAL SELECTIVE WRITE ERROR RECOVERY ALGORI  
TO DETECT BAD SPOTS ON TAPE.  
REVISION D MARCH 84  
- FIX ERROR ROUTINES SO THAT DATA COMPARE ERRORS IN TEST 2  
DO NOT CAUSE OTHER PROBLEMS.

613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651

## 2.0 OPERATING INSTRUCTIONS

FOR OPERATING INSTRUCTIONS, PLEASE SEE CHAPTER 5 OF XXDP OPERATOR'S MANUAL.

## 2.1 HARDWARE PARAMETERS

ON A "N" RESPONSE TO "CHANGE HW?", THE DIAG SHALL RUN ASSUMING ONE UNIT AT TSSR = 172522 WITH A VECTOR = 224.

ON A "Y" RESPONSE TO "CHANGE HW?" QUESTION, THEN THE FOLLOWING QUESTIONS WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.

TSSR ADDRESS (172522) ?

VECTOR (224) ?

THE VALIDITY OF THESE PARAMETERS CAN BE CHECKED BEFORE RUNNING THE TESTS BY SETTING THE FLAG "ADR" ON A STA, RES OR CON COMMAND. THE SO CALLED AUTO DROP CODE SHALL THEN BE EXECUTED AFTER THE INIT CODE AND BEFORE THE HARDWARE TESTS ARE RUN. THAT CODE FIRST TESTS THE ADDRESS OF THE TSSR(S). IF NO RESPONSE, IT DROPS THE UNIT(S) IMMEDIATELY WITH THE FOLLOWING MESSAGE:

BUS TRAP AT XXXXXX ( XXXXXX = TSSR AD )  
INTERFACE BAD OR NOT SET TO ABOVE AD.

ON A RESPONSE FROM THE INTERFACE, THE UNITS THAT ARE NOT READY OR NOT ON-LINE ARE DROPPED IMMEDIATELY. THE HARDWARE TESTS SHALL THEN BE RUN ON RESPONDING UNITS.

IF THE "ADR" FLAG IS NOT SET, THE READY AND OFF-LINE STATUS OF THE UNITS ARE CHECKED. A MESSAGE SHALL BE PRINTED EVERY SC OFTEN TO WARN THE OPERATOR OF UNITS BEING NOT READY OR OFF LINE. THESE UNITS SHALL BE DROPPED AFTER A REASONABLE AMOUNT OF TIME (3 MIN ON A 11'70).

2.2 SOFTWARE PARAMETERS

THE FOLLOWING QUESTIONS ARE ASKED IF REQUESTED ON A START, RESTART,  
OR CONTINUE. THEY ALLOW FLEXABILITY IN THE WAY THE PROGRAM BEHAVES.

CLEAR COUNTERS (L) Y ?

RESET RANDOM VARIABLES (L) N ?

PRINT RECOVERABLE ERRORS (L) N ?

HALT AFTER EACH CMD (L) N ?

INHIBIT RECOVERY (L) N ?

BAD TAPE SPOT DETECTION (L) Y ?

DISABLE INTERRUPTS (L) N ?

INHIBIT RFC ERROR REPORTS (L) N ?

CHANGE CMD SEQUENCE (L) N ?

NOTE: THIS QUESTION SHOULD BE ANSWERED (N) UNLESS AN  
OPERATOR SELECTED SEQUENCE IS TO BE EXECUTED.  
IF THIS QUESTION WAS ANSWERED (N), NO MORE  
QUESTIONS WILL BE ASKED. IF THIS QUESTION WAS  
ANSWERED Y, THE FOLLOWING QUESTIONS MUST BE  
ANSWERED OR DEFAULTED WITH A <CR> ONLY:

CHARACTERISTICS CODE (D) 40 ?	(0,20,40,200) (OCTAL)
CMD/2 (D) 13 ?	(1-27) (DECIMAL)
BRF COUNT (D) 1 ?	(1-2K) (DECIMAL)
# OF OPERATIONS (D) 1 ?	(1-32K) (DECIMAL)
PATTERN (D) 7 ?	(0-8) (DECIMAL)
CMD/3 (D) 4 ?	(1-27) (DECIMAL)
BRF COUNT (D) 2048 ?	(1-2K) (DECIMAL)
# OF OPERATIONS (D) 32000 ?	(1-32K) (DECIMAL)
PATTERN (D) 7 ?	(0-8) (DECIMAL)
CMD/4 (D) 3 ?	(1-27) (DECIMAL)
BRF COUNT (D) 2048 ?	(1-2K) (DECIMAL)
# OF OPERATIONS (D) 32000 ?	(1-32K) (DECIMAL)
PATTERN (D) 7 ?	(0-8) (DECIMAL)
CMD/5 (D) 2 ?	(1-27) (DECIMAL)
BRF COUNT (D) 2048 ?	(1-2K) (DECIMAL)
# OF OPERATIONS (D) 32000 ?	(1-32K) (DECIMAL)
PATTERN (D) 7 ?	(0-8) (DECIMAL)
CMD/6 (D) 13 ?	(1-27) (DECIMAL)
BRF COUNT (D) 1 ?	(1-2K) (DECIMAL)
# OF OPERATIONS (D) 1 ?	(1-32K) (DECIMAL)
PATTERN (D) 7 ?	(0-8) (DECIMAL)
CMD/7 (D) 27 ?	(1-27) (DECIMAL)
BRF COUNT (D) 2048 ?	(1-2K) (DECIMAL)
# OF OPERATIONS (D) 32000 ?	(1-32K) (DECIMAL)
PATTERN (D) 7 ?	(0-8) (DECIMAL)

652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707

708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720

CMD/8 (D) 27. ? (1-27) (DECIMAL)  
BRF COUNT (D) 2048 ? (1-2K) (DECIMAL)  
# OF OPERATIONS (D) 32000 ? (1-32K) (DECIMAL)  
PATTERN (D) 7 ? (0-8) (DECIMAL)

NOTE: THE PROGRAM AUTOMATICALLY INSERTS AN CHARACTERISTIC 40 AS THE FIRST COMMAND IN THE SEQUENCE TABLE. IF A DIFFERENT CHARACTERISTIC IS DESIRED, THE OPERATOR SHOULD ENTER THAT CHARACTERISTIC CODE. A TOTAL OF 7 COMMANDS MAY BE ENTERED IN ADDITION TO THE SET CHARACTERISTICS COMMAND. IF THE OPERATOR WISHES TO USE LESS THAN 7 COMMANDS, AN END COMMAND MUST BE ENTERED AND THEN A CONTROL Z (^Z) CAN BE ENTERED TO TERMINATE SOFTWARE DIAL

2.2.1 COMMAND LIST FOR USE IN SOFTWARE DIALOGUE.

	CODE	COMMAND	DESCRIPTION
721	1	DRI	DRIVE INITIATE.
722	2	RDF	READ FORWARD.
723	3	RDR	READ REVERSE.
724	4	WRT	WRITE.
725	5	WTV	WRITE/VERIFY. IE. WRITE N RECORDS; READ REVERSE AND CHEC
726			N RECORDS OF DATA; READ FORWARD AND CHECK N RECORDS.
727	6	SRF	SPACE RECORDS FORWARD.
728	7	SRR	SPACE RECORDS REVERSE.
729	8	RNR	READ NEXT REVERSE, IE. SPACE FWD, READ REV.
730	9	RNF	READ NEXT FORWARD, IE. READ FWD, SPACE REV.
731	10	RPF	READ PREVIOUS FWD, IE. SPACE REV, READ FWD.
732	11	RPR	READ PREVIOUS REV, IE. READ REV, SPACE FWD.
733	12	WRR	WRITE RETRY.
734	13	RWD	REWIND.
735	14	MBR	MESSAGE BUFFER RELEASE.
736	15	WTM	WRITE TAPE MARK.
737	16	WTR	WRITE TAPE MARK RETRY.
738	17	SFF	SPACE FILES FORWARD.
739	18	SFR	SPACE FILES REVERSE.
740	19	GES	GET EXTENDED STATUS.
741	20	ERS	ERASE 3 INCHES OF TAPE.
742	21	UNL	UNLOAD.
743	22	CLN	CLEAN TAPE
744	23	SCM	SET DEVICE CHARACTERISTIC. WHERE BRF=200, 40, 20, 0.
745			200 = ENABLE SKIP TAPE MARKS STOP (STOP AT LOGICAL EOT)
746			40 = ENABLE ATTENTION INTERRUPTS.
747			20 = ENABLE MESSAGE BUFFER RELEASE INTERRUPTS.
748			SEE TS11/TS04 PROGRAMMING SPECIFICATION FOR DESCRIPTION.
749	24	DIA	DIAGNOSTICS. SEE TS11/TS04 PROGRAMMING SPECIFICATION
750			FOR DESCRIPTION. ODT MUST BE USED TO LOAD DIAGNOSTIC DA
751			INTO THE WRITE BUFFER BEFORE THIS CMD IS ISSUED.
752	25	JMP	JUMP TO THE NTH COMMAND IN THE COMMAND SEQUENCE
753			TABLE, WHERE N IS DEFINED IN THE BRF FIELD.
754			THE NUMBER OF JUMPS IS ENTERED IN THE # OF OPERATIONS FI
755	26	DLY	DELAY "N" MILLISECONDS WHERE N IS DEFINED IN
756			THE # OF OPERATIONS.
757	27	END	END OF COMMAND SEQUENCE.

2.2.2 DATA PATTERN LIST FOR USE IN SOFTWARE DIALOGUE.

	PATTERN #	DESCRIPTION.
764	0	INCREMENTING PATTERN. 0 377.
765	1	ALL "1"'S PATTERN.
766	2	ALL "0"'S PATTERN.
767	3	"1" BIT WALKING FROM R TO L IN A FIELD OF "0" S.
768	4	"0" BIT WALKING FROM R TO L IF A FIELD OF "1" S.
769	5	ALTERNATING "1" AND "0" BITS WITH ALTERNATE BYTES COMPL.
770	6	ALTERNATING BYTES OF 000 AND 377.
771	7	RANDOM DATA PATTERN.
772	8	NO PATTERN GENERATION.



777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796

2.3 EXAMPLES OF SOFTWARE DIALOGUE

2.3.1 BASIC FUNCTION AND DATA RELIABILITY WITH ALL ERROR REPORTING ENABLED

- A) RECEIVE PROMPT (DR>)
- B) ENTER STA/TES:1-2<CR>
- C) ANSWER HARDWARE QUESTIONS.
- D) PROCEED WITH THE FOLLOWING DIALOGUE:

CHANGE SW (L) ?	Y<CR>
CLEAR COUNTERS (L) N ?	Y<CR>
RESET RANDOM VARIABLES (L) N ?	N<CR>
PRINT RECOVERABLE ERRORS (L) N ?	Y<CR>
HALT AFTER EACH CMD (L) N ?	N<CR>
INHIBIT RECOVERY (I) N ?	N<CR>
BAD TAPE SPOT DETECTION (L) Y ?	Y<CR>
DISABLE INTERRUPTS (L) N ?	N<CR>
INHIBIT RFC ERROR REPORT (L) N ?	N<CR>
CHANGE CMD SEQUENCE (L) N ?	N<CR>

797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843

2.3.2 TO SET UP A SCOPE LOOP FOR A FAILURE IN BASIC FUNCTIONS.

- A) RECEIVE PROMPT (DR>)
- B) ENTER STA/TES:1/FLA:LOE:IER:ISR:IDU<CR>
- C) ANSWER HARDWARE QUESTIONS.
- D) PROCEED WITH THE FOLLOWING DIALOGUE:
  - CHANGE SW (L) ? Y<CR>
  - CLEAR COUNTERS (L) N ? Y<CR>
  - RESET RANDOM VARIABLES (L) N ? N<CR>
  - PRINT RECOVERABLE ERRORS (L) N ? N<CR>
  - HALT AFTER EACH CMD (L) N ? N<CR>
  - INHIBIT RECOVERY (L) N ? N<CR>
  - BAD TAPE SPOT DETECTION (L) Y ? N<CR>
  - DISABLE INTERRUPTS (L) N ? N<CR>
  - INHIBIT RFC ERROR REPORT (L) N ? Y<CR>
  - CHANGE CMD SEQUENCE (L) N ? N<CR>

2.3.3 TO SET UP A SCOPE LOOP FOR A FAILURE IN DATA RELIABILITY

- A) RECEIVE PROMPT (DR>)
- B) ENTER STA/TES:5/FLA:IER:ISR:IDU/EOP:1000<CR>
- C) ANSWER HARDWARE QUESTIONS.
- D) PROCEED WITH THE FOLLOWING DIALOGUE:
  - CHANGE SW (L) ? Y<CR>
  - CLEAR COUNTERS (L) N ? Y<CR>
  - RESET RANDOM VARIABLES (L) N ? N<CR>
  - PRINT RECOVERABLE ERRORS (L) N ? N<CR>
  - HALT AFTER EACH CMD (L) N ? N<CR>
  - INHIBIT RECOVERY (L) N ? N<CR>
  - BAD TAPE SPOT DETECTION (L) Y ? N<CR>
  - DISABLE INTERRUPTS (L) N ? Y<CR>
  - INHIBIT RFC ERROR REPORT (L) N ? Y<CR>
  - CHANGE CMD SEQUENCE (L) N ? Y<CR>
  - CHARACTERISTIC CODE (D) 40 ? 40<CR>
  - CMD/2 (D) 5 ? 13<CR> (REWIND) (COULD
  - BRF COUNT (D) 2048 ? 1<CR>
  - # OF OPERATIONS (D) 10 ? 1<CR>
  - PATTERN (D) 7 ? 1<CR>
  - CMD/3 (D) 5 ? 4<CR> (WRITE) (COULD B
  - BRF (D) 2048 ? 1000<CR>
  - # OF OPERATIONS (D) 10 ? 10000<CR>
  - PATTERN (D) 7 ? 1<CR>
  - CMD/4 (D) 5 ? 27<CR> (END) (COULD B
  - BRF (D) 2048 ? <+2>

844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868

2.4 EXECUTION TIMES

2.4.1 SYSTEM CONFIGURATION

PDP11/34  
MOS MEMORY  
LA36  
TS11/TS04

2.4.2 TEST EXECUTION TIMES

- TEST 1 - BASIC FUNCTIONS - 30 SECONDS PER PASS.
- TEST 2 - DATA RELIABILITY - 45 MINUTES PER PASS.
- TEST 3 - WRITE COMPATABILITY - 20 MINUTES PER PASS.
- TEST 4 - READ COMPATABILITY - 20 MINUTES PER PASS.
- TEST 5 - OPERATOR SELECTED SEQUENCE - DEPENDS ON SEQUENCE SELECTED.

NOTE: ALL EXECUTION TIMES ARE SHOWN FOR ONE UNIT OEPRATION.  
APPROXIMATELY 10% WILL BE ADDED TO ALL EXECUTION TIMES  
FOR EACH ADDITIONAL UNIT.

M2

869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
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

3.0 ERROR INFORMATION  
-----

3.1 ERROR REPORTING  
-----

ALL ERROR REPORTS EXCEPT FOR ERRORS #1 AND #17 INCLUDE A DUMP OF THE FOLLOWING INFORMATION:

ERROR #, TEST #, SUBTEST #, PROGRAM COUNTER, UNIT #, COMMAND, PREVIOUS COMMAND, PASS COUNT, # OF RECORDS FROM BOT, RECORD READ COUNT, THE COMMAND PACKET, TSSR, TCC, TSBA, RFC, AND THE EXTENDED STATUS REGISTERS (SEE 2.3.14.1 FOR LIST OF COMMANDS).

STANDARD ERROR REPORT FORMAT:

```
CZTSH SFT ERR XXXXX TST XXX SUB XXX PC: XXXXXX
(ASCII ERROR MESSAGE)
XXX CMD FAILED - UNIT X PASS: XXXXX RECORD: XXXXX
PREVIOUS CMD WAS XXX * RECORD READ: XXXXX *
CMDPKT TSBA RFC TSSR TCC
XXXXXX XXXXXX XXXXXX XXXXXX X
XXXXXX
XXXXXX
XXXXXX
XST0 XST1 XST2 XST3
XXXXXX XXXXXX XXXXXX XXXXXX
```

\* CAUTION \*

INTERPRET THAT "RECORD READ" COUNT WITH CAUTION. IF VERY DIFFERENT FROM RECORD COUNT TRACKED BY THE DIAGNOST POSITION IS NOT NECESSARELY LOST. ERRORS IN READING THAT RECORD MIGHT HAVE CAUSED RECORD COUNT TO BE ERRONEOUSLY READ FROM TAPE. IN TEST 2, IF DIAGNOSTIC IS RESTARTED OR CONTINUED, RECORD IS RESET TO ZERO ALTHOUGH TAPE WAS NOT REWOUND. THIS IS NECESSARY BECAUSE THERE IS NO ACCURATE WAY TO DETERMINE ON WHAT RECORD COUNT OF WHAT UNIT THE DIAGNOSTIC WAS HALTED BEFORE RESTARTING OR CONTINUING. IT IS SUGGESTED THAT A 'PRINT' BE REQUESTED WHEN HALTING DI TO GET A PRINT OF THE RECORD COUNT WHEN HALTED.

EXAMPLE OF AN ERROR REPORT:

```
CZTSH SFT ERR 00009 1ST 002 SUB 000 PC: 010606
RECOVERABLE ERROR
WRT CMD FAILED - UNIT 2 PASS: 2 RECORD: 254
PREVIOUS CMD WAS WRT
CMDPKT TSBA RFC TSSR TCC
100005 002324 000000 100210 4
051766
000000
```

000371  
XST0 XST1 XST2 XST3  
000350 000002 100004 000000

925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980

3.1.1 ERROR #1 COMMAND PACKET ADDRESS NOT ON A MODULO 4 BOUNDARY:

IF THIS ERROR IS REPORTED, THE PROGRAM DID NOT LOAD PROPERLY. THIS IS A SYSTEM FATAL ERROR AND THE PROGRAM MUST BE RELOADED TO CORRECT IT.

3.1.2 ERROR #2 TS04 NOT READY:

BEFORE ANY COMMAND IS ISSUED TO THE TS04, THE SUBSYSTEM READY BIT IN THE TSS4 IS CHECKED. IF THE SSR IS NOT SET, THE PROGRAM REPORTS THE NOT READY ERROR. THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST SEQUENCE UNLESS THE IDU OPTION IS USED.

3.1.3 ERROR #3 NO RESPONSE ERROR:

ONCE THE TSDB IS LOADED, THE TS04 HAS ONE MILLISECOND TO RESPOND OR THE PROGRAM REPORTS A NO RESPONSE ERROR. THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST SEQUENCE UNLESS THE IDU OPTION IS USED.

3.1.4 ERROR #4 - NO INTERRUPT ERROR:

COMMAND WAS ISSUED AND NO INTERRUPT RECEIVED. THE PROGRAM REPORTS THAT NO INTERRUPT OCCURRED. THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST CYCLE UNLESS THE IDU OPTION IS USED.

3.1.5 SPECIAL CONDITION ERRORS:

IF, DURING EXECUTION, AN INCIDENT OCCURS FORCING THE TSSR SPECIAL CONDITION BIT TO SET, THE PROGRAM WILL SELECT ONE OF 8 ERROR HANDLING ROUTINES, DEPENDING ON THE TERMINATION CLASS CODE.

THE TERMINATION CLASS CODES IN THE TSSR ARE PROCESSED AS FOLLOWS WHEN SPECIAL CONDITION IS SET:

3.1.5.1 ERROR #5 - TERMINATION CLASS CODE 0, UNDEFINED SPECIAL CONDITION

THE ERROR IS REPORTED, A HARD ERROR IS LOGGED AND THE PROGRAM PROCEEDS NORMALLY.

3.1.5.2 ERROR #6 TERMINATION CLASS CODE 1, ATTENTION CONDITION

981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036

THIS TCC INDICATES THAT THE DRIVE HAS UNDERGONE A STATUS CHANGE SUCH AS GOING OFFLINE OR COMING ONLINE. THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST CYCLE UNLESS THE IDU OPTION IS USED.

3.1.5.3 ERROR #7 TERMINATION CLASS CODE 2, TAPE STATUS ALERT

A STATUS CONDITION HAS BEEN ENCOUNTERED THAT MAY HAVE SIGNIFICANCE TO THE PROGRAM. BITS OF INTEREST INCLUDE TMK, RLS, LET, RLL, EOT. ACTION TAKEN DEPENDS ON THE TEST BEING EXECUTED. IF THE CONDITION IS UNEXPECTED, THE ERROR IS REPORTED AND A HARD ERROR IS LOGGED. THE PROGRAM PROCEEDS NORMALLY.

3.1.5.4 ERROR #8 TERMINATION CLASS CODE 3, FUNCTION REJECT

THE SPECIFIED FUNCTION WAS NOT INITIATED. BITS OF INTEREST ARE RMR, OFL, VCK, BOT, ILC, WLE, ILA, AND NBA. THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST CYCLE UNLESS THE IDU OPTION IS USED.

3.1.5.5 ERROR #9 - TERMINATION CLASS CODE 4, RECOVERABLE ERROR

TAPE POSITION IS ONE RECORD BEYOND WHAT ITS POSITION WAS WHEN THE FUNCTION WAS INITIATED. RECOVERY PROCEDURE IS TO LOG THE ERROR AND ISSUE THE APPROPRIATE RETRY COMMAND. IF RETRY LIMIT IS REACHED BEFORE THE ERROR IS RECOVERED, RETRY LIMIT EXCEEDED IS REPORTED AS DESCRIBED IN ERROR #14 BELOW.

3.1.5.6 ERROR #10 TERMINATION CLASS CODE 5, RECOVERABLE ERROR

TAPE POSITION HAS NOT CHANGED. RECOVERY PROCEDURE IS TO LOG THE ERROR AND RE ISSUE THE ORIGINAL COMMAND. IF RETRY LIMIT IS REACHED BEFORE THE ERROR IS RECOVERED, RETRY LIMIT EXCEEDED IS REPORTED AS DESCRIBED IN ERROR #14 BELOW.

3.1.5.7 ERROR #11 - TERMINATION CLASS CODE 6, UNRECOVERABLE ERROR

TAPE POSITION HAS BEEN LOST. THE ONLY VALID RECOVERY PROCEDURE IS TO REWIND AND START OVER AT BOT UNLESS THE TAPE HAS LABELS OR SEQUENCE NUMBERS. IF DENSITY CHECK IS SET THIS DIAGNOSTIC WILL REWIND AND RETRY THE COMMAND. OTHERWISE THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST CYCLE UNLESS THE IDU OPTION IS USED.

3.1.5.8 ERROR #12 TERMINATION CLASS CODE 7, FATAL SUBSYSTEM ERROR

THE SUBSYSTEM IS INCAPABLE OF PROPERLY PERFORMING COMMANDS OR AT LEAST ITS INTEGRITY IS SERIOUSLY QUESTIONABLE. REFER TO THE FATAL CLASS CODE FIELD IN THE TSSR REGISTER FOR ADDITIONAL INFORMATION ON THE TYPE OF FATAL ERROR. THE DEVICE WILL BE DROPPED FROM THE TEST CYCLE UNLESS THE IDU OPTION IS USED.

1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087

3.1.6 ERROR #13 RFC NON ZERO ERROR:

IF, AFTER EXECUTION, THE RESIDUAL FRAME COUNT IS NON ZERO, THE ERROR IS REPORTED AND A HARD ERROR IS LOGGED. THE PROGRAM THEN PROCEEDS NORMALLY. THE REPORTING AND LOGGING OF THESE ERRORS IS OPTIONAL.

3.1.7 ERROR #14 RETRY LIMIT EXCEEDED:

ON A WRITE COMMAND THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST CYCLE UNLESS THE IDU OPTION IS USED.

ON A READ COMMAND THIS ERROR IS LOGGED AS A HARD ERROR AND THE PROGRAM PROCEEDS NORMALLY.

3.1.8 ERROR #15 - TOO MANY INTERRUPTS:

IF MORE THAN ONE INTERRUPT OCCURS PER COMMAND, THIS ERROR IS REPORTED. THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST CYCLE UNLESS THE IDU OPTION IS USED.

3.1.9 ERROR #16 CAPSTAN RUNAWAY:

CAPSTAN DID NOT STOP WITHIN ACCEPTABLE WINDOW AFTER LAST COMMAND. THE PROGRAM WILL ISSUE A GET STATUS COMMAND BEFORE REPORTING THE ERROR SO THAT THE DEAD TRACK FIELD IN EXTENDED STATUS REGISTER 2 WILL CONTAIN THE TACH COUNT WHEN THE TAPE STOPPED. THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST CYCLE UNLESS THE IDU OPTION IS USED.

3.1.10 ERROR #17 DATA COMPARE ERROR:

IF A DATA VALIDATION ERROR OCCURS DURING A WRITE/VERIFY COMMAND, THE PROGRAM PRINTS WHAT THE DATA SHOULD HAVE BEEN AND WHAT THE DATA WAS, AND PRINTS THE BYTE AND RECORD NUMBER THE ERROR OCCURRED ON. ONLY THE FIRST 10 BYTES IN ERROR PER RECORD ARE PRINTED. THE TOTAL # OF BYTES IN ERROR PER RECORD IS ALSO PRINTED. A HARD ERROR IS LOGGED AND THE PROGRAM PROCEEDS NORMALLY.

3.2 ERROR HALTS

ERROR HALTS ARE SUPPORTED PER DESCRIBED IN THE PREVIOUS SECTION WITH /FLAG:MOE. THERE ARE NO OTHER HALTS.



1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143

4.0 PERFORMANCE REPORT

UNIT X PASS:XXXXX RECORD:XXXXX  
BYTES WRITTEN XXX,XXX,XXX,XXX  
BYTES READ REV XXX,XXX,XXX,XXX  
BYTES READ FWD XXX,XXX,XXX,XXX  
RECOVERABLE ERRORS WRT RDR RDF  
UNRECOVERABLE ERRORS XXXXX XXXXX XXXXX  
SPEC COND HARD FATAL COMPARE  
XXXXX XXXXX XXXXX XXXXX

5.0 TEST SUMMARIES

5.1 TEST 1 -

BASIC FUNCTIONS.

EXECUTES AND VERIFIES CORRECT COMPLETION OF ALL TS04 F/JN

- SUBTEST 1 SET CHAR, DRIVE INIT, GET STATUS.
  - SET CHARACTERISTIC 200.
  - DRIVE INITIATE.
  - SET CHARACTERISTIC 20.
  - GET STATUS
  - SET CHARACTERISTIC 40.
  - PRINT TS04 MICROCODE LEVEL (PASS 1 ONL
- SUBTEST 2 REWIND.
  - REWIND.
  - REWIND AT BOT.
- SUBTEST 3 WRITE/VERIFY.
  - WRITE/VERIFY PATTERN 1.
  - WRITE/VERIFY PATTERN 2.
  - WRITE/VERIFY PATTERN 3.
  - WRITE/VERIFY PATTERN 4.
  - WRITE/VERIFY PATTERN 5.
  - WRITE/VERIFY PATTERN 6.
  - WRITE/VERIFY PATTERN 0.
- SUBTEST 4 WRITE TAPE MARK, ERASE.
  - WRITE TAPE MARK.
  - WRITE 10 RECORDS
  - ERASE 10 TIMES
  - WRITE TAPE MARK.
  - WRITE TAPE MARK RETRY.
- SUBTEST 5 SPACE FILES.
  - SPACE 2 FILES REVERSE.
  - SPACE 2 FILES FORWARD.

1144		* SPACE 2 FILES REVERSE.
1145		* SPACE 2 FILES FORWARD.
1146		
1147		
1148		
1149		
1150		
1151		
1152		
1153		
1154		
1155		
1156		
1157		
1158		
1159		
1160		
1161		
1162		
1163		
1164		
1165		
1166		
1167		
1168		
1169		
1170		
1171		
1172		
1173		
1174		
1175		
1176		
1177		
1178		
1179		
1180		
1181		
1182		
1183		
1184		
1185		
1186		
1187		
1188		
1189		
1190		
1191		

	SUBTEST 6	SPACE RECORDS.
		* REWIND.
		* SPACE 7 RECORDS FORWARD.
		* SPACE 7 RECORDS REVERSE.
		* SPACE 7 RECORDS FORWARD.
		* SPACE 7 RECORDS REVERSE.
	SUBTEST 7	WRITE RETRY.
		* REWIND.
		* WRITE DATA.
		* WRITE RETRY.
	SUBTEST 8	READ REV RETRY.
		* READ REVERSE.
		* READ NEXT REVERSE.
		* READ NEXT FORWARD.
	SUBTEST 9	READ FWD RETRY.
		* READ FORWARD.
		* READ PREVIOUS FORWARD.
		* READ PREVIOUS REVERSE.
	SUBTEST 10	CLEAN.
		* CLEAN.
		* REWIND.
	SUBTEST 11	WRITE/VERIFY SWAPPED DATA BYTES.
		* WRITE/VERIFY EVEN LENGTH (RECORD 1).
		* WRITE/VERIFY ODD LENGTH (RECORD 2).
		* SET DATA BYTE SWAP.
		* WRITE/VERIFY EVEN LENGTH (RECORD 3).
		* WRITE/VERIFY ODD LENGTH (RECORD 4).
		* CLEAR DATA BYTE SWAP.
	SUBTEST 12	READ SWAPPED DATA BYTES.
		* READ REV RECORD 4.
		* READ REV RECORD 3.
		* SET DATA BYTE SWAP.
		* READ REV RECORD 2.
		* READ REV RECORD 1.
		* READ FWD RECORD 1.
		* READ FWD RECORD 2.
		* CLEAR DATA BYTE SWAP.
		* READ FWD RECORD 3.
		* READ FWD RECORD 4.

1192	5.2	TEST 2	DATA RELIABILITY.
1193			
1194			1. THE TAPE IS INITIATED WITH THE FOLLOWING COMMANDS:
1195			SET CHARACTERISTIC 40
1196			REWIND
1197			WRITE/VERIFY 31 RECORDS OF RANDOM LENGTH AND DAT
1198			2. WRITE AND READ COMMANDS ARE SELECTED AT RANDOM AND
1199			EXECUTED A RANDOM NUMBER OF TIMES WITH RANDOM
1200			LENGTHS AND RANDOM PATTERN UNTIL END OF TAPE IS REA
1201			3. AT THE END OF EACH PASS, A REWIND COMMAND IS ISSUED
1202			A PERFORMANCE REPORT IS PRINTED.
1203			
1204			NOTE: IF A RESTART COMMAND IS USED TO INITIATE
1205			TEST 1, THE INITIAL REWIND COMMAND IS NO
1206			
1207			
1208	5.3	TEST 3	WRITE COMPATABILITY/WRITE UTILITY.
1209			
1210			REWINDS AND WRITES RECORDS OF RANDOM LENGTHS
1211			AND RANDOM DATA FROM BOT TO EOT.
1212			
1213			
1214	5.4	TEST 4	READ COMPATABILITY/READ UTILITY.
1215			
1216			REWINDS AND READS ENTIRE TAPE, FORWARD AND REVERSE.
1217			
1218			
1219	5.5	TEST 5	EXFCUTE OPERATOR SELECTED COMMAND SEQUENCE.
1220			
1221			THE SEQUENCE OF COMMANDS ENTERED BY THE OPERATOR
1222			IS EXECUTED. IF NO COMMANDS WERE ENTERED, A
1223			DEFAULT SEQUENCE OF REWIND/WRITE/READ REV/READ FWD/REWIN
1224			OF ENTIRE TAPE IS EXECUTED WITH RANDOM PATTERN
1225			AND RECORD LENGTH OF 2048 BITES.

1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279

## 6.0 DEVICE INFORMATION TABLES

### 6.1 GENERAL

THE TS04 TAPE SUBSYSTEM CONSISTS OF A TS11 UNIBUS TO SERIAL BUS CONTROLLER CONNECTED TO A TS04 DRIVE. FROM A SOFTWARE VIEWPOINT THIS CONFIGURATION IS UNIQUE (FOR A UNIBUS DEVICE) IN A NUMBER OF WAYS:

- A. ONLY ONE REGISTER MAY BE WRITTEN - TSDB (TAPE SYSTEM DATA BUFFER).
- B. TWO REGISTERS MAY BE READ TSSR AND TSBA (TAPE SYSTEM STATUS REGISTER AND TAPE SYSTEM BUS ADDRESS REGISTER).
- C. COMMANDS ARE NOT WRITTEN TO THE DRIVE; RATHER, COMMAND POINTERS ARE WRITTEN WHICH POINT TO COMMAND PACKETS SOMEWHERE IN CPU MEMORY. THE COMMAND POINTER IS USED BY THE TS04 SUBSYSTEM TO FETCH THE WORD(S) WITHIN THE COMMAND PACKET. THE WORDS WITHIN THE COMMAND PACKET ARE:
  1. COMMAND WORD
  2. LOW ORDER BUFFER ADDRESS
  3. HIGH ORDER BUFFER ADDRESS
  4. BYTE COUNT
- D. THE TSSR CONTAINS ALL THE INFORMATION WHICH WILL BE NECESSARY TO DETERMINE WHETHER:
  1. THE DRIVE IS READY TO ACCEPT ANOTHER COMMAND.
  2. THE PREVIOUS COMMAND WAS EXECUTED WITHOUT ERROR.
 IF EITHER OF THE ABOVE CONDITIONS IS UNTRUE AT "JOB DONE" OR "COMMAND INITIATION" TIME, IT MAY BE NECESSARY TO GET THE EXTENDED STATUS REGISTERS TO DETERMINE WHAT ACTION IS TO BE TAKEN AND/OR LOG THE ERROR INFORMATION.
- E. EXTENDED STATUS REGISTERS ARE NOT READ DIRECTLY FROM DRIVE REGISTERS; RATHER, A "GET STATUS" COMMAND IS ISSUED WHICH WILL CAUSE THE TS04 TO TRANSFER EXTENDED STATUS INFORMATION TO THE MEMORY AREA POINTED TO BY THE BUFFER ADDRESS OF THE "GET STATUS" COMMAND. THERE ARE FOUR EXTENDED STATUS REGISTERS. SEE 6.3.
- F. THE TSDB MUST BE WRITTEN WITH A DATO INSTRUCTION TO PROPERLY WRITE THE COMMAND POINTER. A DATOB WILL CAUSE A MAINTENANCE FUNCTION. A DATO TO THE TSSR WILL CAUSE SUBSYSTEM INIT.
- G. COMMAND PACKETS MUST RESIDE ON DIVIDE BY FOUR MEMORY BOUNDARIES (AS OPPOSED TO DIVIDE BY 2 OR WORD BOUNDARIES).

1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298

6.2 UNIBUS INTERFACE SPECIFICATIONS  
-----

TS11/ TS04 -----	INT. VECTOR ---	UNIBUS ADDRESS --	REGISTER --
FIRST	224	772520 772522	TSBA/TSDB TSSR
SECOND	154	772524 772526	TSBA/TSDB TSSR
THIRD	160	772530 772532	TSBA/TSDB TSSR
FOURTH	164	772534 772536	TSBA/TSDB TSSR

6.3 BIT DEFINITIONS FOR TS11/TS04 REGISTERS

6.3.1 TS11/TS04 REGISTER SUMMARY

1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	
TSBA	A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	
TSDB	P15	P14	P13	P12	P11	P10	P09	P08	P07	P06	P05	P04	P03	P02	P01	
TSSR	SC	UPE	SPE	RMR	NXM	NBA	A17	A16	SSR	OFL	FC1	FC0	TC2	TC1	TC0	
XST0	TMK	RLS	LET	RLI	WLE	NEF	ILC	ILA	MOT	ONL	IE	VCK	PED	WLK	BO	
XST1	DLT		COR	CRS	TIG	DBF	SCK		IPR	SYN	IPO	IED	POS	POL	UN	
XST2					NZO				DRP		ITM	LCO	NZN	LRC	CR	
XST3								DTP	DT7	DT6	DT5	DT4	DT3	DT2	DT1	
	MICRO DIAGNOSTIC ERROR CODE								LMX	OPI		REV	CRF	DCK	NOI	LX

TERMINATION CLASS CODES (TSSR TC0-TC2):

- 0 = NORMAL TERMINATION
- 1 = ATTENTION CONDITION
- 2 = TAPE STATUS ALERT
- 3 = FUNCTION REJECT
- 4 = RECOVERABLE ERROR - TAPE POSITION = ONE RECORD DOWN TAPE FROM START OF FUNCTION
- 5 = RECOVERABLE ERROR - TAPE NOT MOVED
- 6 = UNRECOVERABLE ERROR - TAPE POSITION LOST
- 7 = FATAL CONTROLLER ERROR

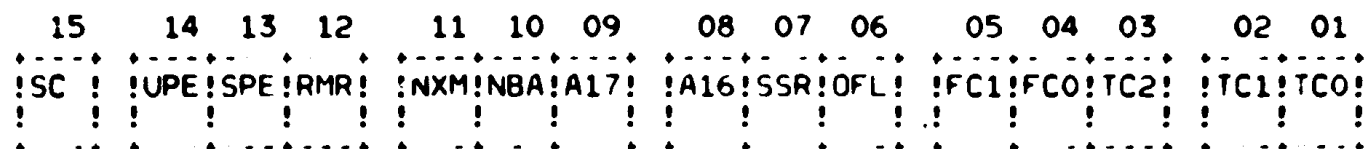
FATAL CLASS CODES (TSSR FC0-FC1):

- 0 = MICRO DIAGNOSTIC FAILURE (DISPLAYED IN TS04 OPERATOR PANEL AND
- 1 = I/O SEQUENCER CROM PARITY ERROR.
- 2 = MICROPROCESSOR CROM PARITY ERROR. SILO PARITY ERROR. SERIAL BUS PARITY ERROR DETECTED AT TS11 (SPE). SERIAL BUS PARITY ERROR DETECTED AT TS04 (BPE). FATAL ERROR HALTS 1750 1777 IN TS04 PROGRAM COUNTER DISPLAY.
- 3 = LOSS OF AC POWER HAS BEEN DETECTED.

1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409

6.3.2 TS11 STATUS REGISTER (TSSR)

UNIBUS ADDRESS + 2 - READ ONLY



BIT	NAME	TCC	DEFINITION
15	SC	S	SPECIAL CONDITION. WHEN SET, INDICATES THAT THE LAST COMMAND DID NOT COMPLETE WITHOUT INCIDENT. SPECIFICALLY, EITHER AN ERROR WAS DETECTED OR AN EXCEPTION CONDITION OCCURRED. EXCEPTION CONDITIONS CAN BE TAPE MARKS ON READ COMMANDS, REVERSE MOTION AND AT BOT, EOT WHILE WRITING, ETC. MAY ALSO BE SET BY THE ERROR BITS CONTAINED IN THE TSSR REGISTER: UPE, SPE, RMR, AND NXM. THE TERMINATION CLASS BITS ARE SOMETHING OTHER THAN 0 (UNLESS RMR IS THE ONLY ERROR - SEE RM
14	UPE	4/5	UNIBUS PARITY ERROR. SET BY THE TS11 WHEN IT DETECTS A PARITY ERROR ON THE UNIBUS DATA WHEN TRANSFERRING TO OR FROM THE CPU'S MEMORY.
13	SPE	7	SERIAL BUS PARITY ERROR. THIS BIT IS SET BY THE TS11 WHEN IT DETECTS A SERIAL BUS PARITY ERROR ON DATA RECEIVED FROM THE TS04.
12	RMR	S	REGISTER MODIFICATION REFUSED. SET BY THE TS11 WHEN A COMMAND POINTER IS LOADED INTO TSOB AND SUB-SYSTEM READY (SSR) IS NOT SET. NOTE THAT THIS BIT CAUSES SPECIAL CONDITION BUT NO TERMINATION CLASS (IN FACT, THE TS04 NEVER SEES THIS ERROR) BECAUSE ON A SYSTEM WITH NO BUGS, THIS BIT MAY COME UP ON AN ATTENTION MESSAGE. IF ATTNS ARE NOT ENABLED, THIS BIT COMING UP IS AN INDICATION OF EITHER A FATAL CONTROLLER ERROR OR A SOFTWARE BUG.
11	NXM	4/5	NON-EXISTENT MEMORY. SET BY THE TS11 WHEN TRYING TO TRANSFER TO OR FROM A MEMORY LOCATION WHICH DOES NOT EXIST. MAY OCCUR WHEN FETCHING THE COMMAND PACKET, FETCHING OR STORING DATA, OR STORING THE MESSAGE PACKET.
10	NBA	S	NEED BUFFER ADDRESS. WHEN SET, INDICATES THAT THE TS04 NEEDS A MESSAGE BUFFER ADDRESS. THIS



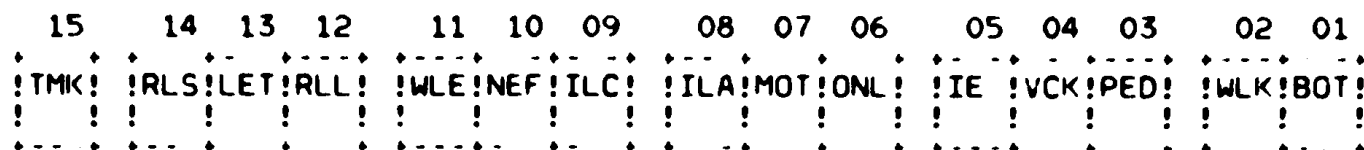
K3

1410				BIT IS CLEARED DURING THE SET CHARACTERISTICS COMMAND (IF A GOOD ADDRESS WAS GIVEN).
1411				
1412				
1413	09	A17	S	BUS ADDRESS BIT 17. A17 AND A16 (BIT 08) TRACK THE VALUES OF BITS 17 AND 16 OF THE TSBA REGISTER.
1414				
1415				
1416				
1417				
1418	08	A16	S	BUS ADDRESS BIT 16. SEE A17 (BIT 09).
1419				
1420	07	SSR	S	SUB SYSTEM READY. WHEN SET, INDICATES THAT THE TS11/TS04 SUBSYSTEM IS NOT BUSY AND IS READY TO ACCEPT A NEW COMMAND POINTER.
1421				
1422				
1423				
1424	06	OFL	S,1,3	OFF-LINE. WHEN SET, INDICATES THAT THE TS04 IS OFF LINE AND UNAVAILABLE FOR ANY TAPE MOTION COMMANDS. THIS BIT CAN CAUSE A TERMINATION CLASS OF 1 (ON ATTN INTERRUPT) OR 3 (RESULTS IN NEF).
1425				
1426				
1427				
1428				
1429	05	FC1	7	FATAL TERMINATION CLASS 01. FC1 AND FC0 (BIT 04) ARE USED TO INDICATE THE TYPE OF FATAL ERROR WHICH HAS OCCURRED ON THE TS04. THESE BITS ARE VALID ONLY WHEN SC IS SET AND THE TERMINATION CLASS CODE BITS ARE ALL SET (111).
1430				
1431				
1432				
1433				
1434				
1435	04	FC0	7	FATAL TERMINATION CLASS 00. SEE FC1 (BIT 05).
1436				
1437	03	TC2	S	TERMINATION CLASS BIT 02. THIS BIT, ALONG WITH THE TC1 AND TC0 BITS, ACT AS AN OFFSET VALUE WHENEVER AN ERROR OR EXCEPTION CONDITION OCCURS ON A COMMAND. EACH OF THE EIGHT POSSIBLE VALUES OF THIS FIELD REPRESENT A PARTICULAR CLASS OF ERRORS OR EXCEPTIONS. THE CONDITIONS IN EACH CLASS HAVE SIMILAR SIGNIFICANCE AND, AS APPLICABLE, RECOVERY PROCEDURES. THE CODE PROVIDED IN THIS FIELD IS EXPECTED TO BE UTILIZED AS AN OFFSET INTO A DISPATCH TABLE FOR HANDLING OF THE CONDITION.
1438				
1439				
1440				
1441				
1442				
1443				
1444				
1445				
1446				
1447				
1448				
1449	02	TC1	S	TERMINATION CLASS BIT 01. SEE TC2 (BIT 03).
1450				
1451	01	TC0	S	TERMINATION CLASS BIT 00. SEE TC2 (BIT 03).
1452				
1453	00			NOT USED.
1454				
1455				
1456				
1457				
1458				
1459				
1460				

UNIBUS ADDRESS + 2 WRITE ONLY  
 SUBSYSTEM INITIALIZE

1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516

6.3.3 EXTENDED STATUS REGISTER 0 (XSTAT0)



BIT	NAME	TCC	DEFINITION
15	TMK	5,2	TAPE MARK DETECTED. SET WHENEVER A TAPE MARK WAS DETECTED DURING A READ, SPACE, OR SKIP COMMAND AND AS A RESULT OF THE WRITE TAPE MARK OR WITE TAPE MARK RETRY COMMANDS.
14	RLS	2	RECORD LENGTH SHORT. THIS BIT INDICATES THAT EITHER THE RECORD'S LENGTH WAS SHORTER THAN THE BYTE COUNT ON READ OPERATIONS, A SPACE RECORD OPERATION ENCOUNTERED A TAPE MARK OR BOT BEFORE THE POSITION COUNT WAS EXHAUSTED, OR A SKIP TAPE MARKS COMMAND WAS TERMINATED BY ENCOUNTERING BOT OR A DOUBLE TAPE MARK (IF THAT OPERATIONAL MODE IS ENABLED, SEE LET) PRIOR TO EXHAUSTING THE POSITION COUNTER.
13	LET	2	LOGICAL END OF TAPE. SET ONLY ON THE SKIP TAPE MARKS COMMAND WHEN EITHER TWO CONTIGUOUS TAPE MARKS ARE DETECTED OR WHEN MOVING OFF OF BOT AND THE FIRST RECORD ENCOUNTERED IS A TAPE MARK. THE SETTING OF THIS BIT WILL NOT OCCUR UNLESS THIS MODE OF TERMINATION IS ENABLED THROUGH USE OF THE SET CHARACTERISTICS COMMAND.
12	RLL	2	RECORD LENGTH ONG. WHEN SET, THIS BIT INDICATES THAT THE RECORD READ WAS LONGER THAN THE BYTE COUNT SPECIFIED.
11	WLE	3,6	WRITE LOCK ERROR. WHEN SET, INDICATES THAT A WRITE OPERATION WAS ISSUED BUT THE MOUNTED TAPE DID NOT CONTAIN A WRITE ENABLE RING OR THE WRT LOCK SWITCH ACTIVATED DURING THE OPERATION.
10	NEF	3	NON-EXECUTABLE FUNCTION. WHEN SET, INDICATES THAT THE COMMAND COULD NOT BE EXECUTED DUE TO ONE OF THE FOLLOWING CONDITIONS:  THE COMMAND SPECIFIED REVERSE TAPE DIRECTION BUT THE TAPE WAS ALREADY POSITIONED AT BOT. THE ISSUING OF ANY COMMAND, EXCEPT REWIND.

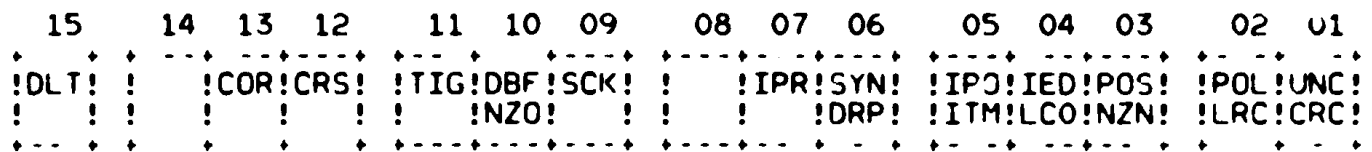
1517					
1518					
1519					
1520					UNLOAD, OR A COMMAND WITH THE CLEAR VOLUME
1521					CHECK (CVC) BIT SET, WHEN THE VOLUME CHECK
1522					BIT IS SET.
1523					- ANY COMMAND, EXCEPT GET STATUS OR DRIVE
1524					INITIALIZE, WHEN THE TSO4 IS OFF-LINE.
1525					- ANY WRITE COMMAND WHEN THE TAPE DOES NOT
1526					CONTAIN A WRITE ENABLE RING (WRITE LOCK
1527					STATUS - WLS).
1528					
1529					
1530					
1531					
1532					
1533					
1534					
1535					
1536					
1537					
1538					
1539					
1540					
1541					
1542					
1543					
1544					
1545					
1546					
1547					
1548					
1549					
1550					
1551					
1552					
1553					
1554					
1555					
1556					
1557					
1558					
1559					
1560					
1561					
1562					
1563					
1564					
1565					
1566					
1567					
1568					
1569					
1570					
1571					

09	ILC	3	ILLEGAL COMMAND. SET WHEN A COMMAND IS ISSUED AND EITHER ITS COMMAND FIELD OR ITS COMMAND MODE FIELD CONTAINS CODES WHICH ARE NOT SUPPORTED BY THE TSO4.
08	ILA	3	ILLEGAL ADDRESS. (MORE THAN 18 BITS OR ODD WHEN AN EVEN ADDRESS IS REQUIRED.)
07	MOT	S	TAPE IS MOVING.
06	ONL	S	ON LINE. WHEN SET, INDICATES THAT THE TSO4 IS ON-LINE AND OPERABLE.
05	IE	S	INTERRUPT ENABLE. REFLECTS THE STATE OF THE INTERRUPT ENABLE BIT SUPPLIED ON THE LAST COMMAND.
04	VCK	S	VOLUME CHECK. WHEN SET, INDICATES THAT THE DRIVE HAS BEEN EITHER POWERED DOWN OR TURNED OFF-LINE. CLEARED BY THE CLEAR VOLUME CHECK (CVC) BIT IN THE COMMAND HEADER WORD. THIS BIT CAN CAUSE A TERMINATION CLASS OF 3.
03	PED	S	PHASE ENCODED DRIVE. WHEN SET, INDICATES THAT THE TSO4 IS CAPABLE OF READING AND WRITING ONLY 1600 BPI PHASE ENCODED DATA. WHEN RESET, INDICATES THAT THE TSO4 HAS ONLY 800 BPI NRZI DATA CAPABILITIES.
02	WLK	S,3	WRITE LOCKED. WHEN SET, INDICATES THAT THE MOUNTED REEL OF TAPE DOES NOT HAVE A WRITE-ENABLE RING INSTALLED. THE TAPE IS, THEREFORE, WRITE PROTECTED.
01	BOT	S,3	BEGINNING OF TAPE. WHEN SET, INDICATES THAT THE TAPE IS POSITIONED AT THE LOAD POINT AS DENOTED BY THE BOT REFLECTIVE STRIP ON THE TAPE.
00	EOT	S,2	END OF TAPE. THIS BIT IS SET WHENEVER THE TAPE IS POSITIONED AT OR BEYOND THE END OF TAPE REFLECTIVE STRIP. DOES NOT RESET UNTIL THE TAPE PASSES OVER THE REFLECTIVE STRIP IN THE REVERSE DIRECTION UNDER PROGRAM CONTROL.

1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627

6.3.4 EXTENDED STATUS REGISTER 1 (XSTAT1)

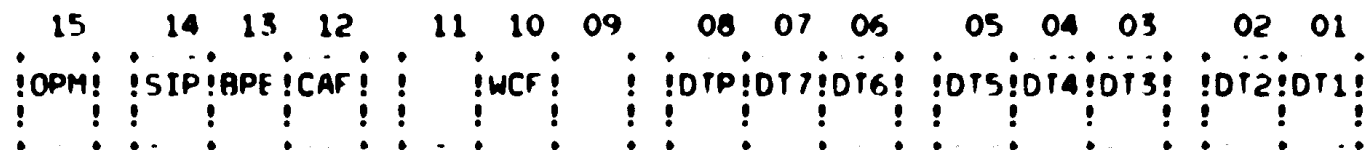


BIT	NAME	TCC	DEFINITION
15	DLT	4	DATA LATE. SET WHEN THE I/O SILO IS FULL ON A READ OR EMPTY ON A WRITE. THESE CONDITIONS OCCUR WHENEVER THE UNIBUS LATENCY EXCEEDS THE DATA TRANSFER RATE OF THE TS04.
14	-	-	NOT USED.
13	COR	5	CORRECTABLE DATA. IN PHASE ENCODED MODE, A CORRECTABLE DATA ERROR HAS BEEN ENCOUNTERED.
12	CRS	4	CREASE DETECTED. FOR NRZI, ALL DATA TRACKS DROPPED OUT FOR MORE THAN THREE CHARACTER TIMES BUT FOR LESS THAN .1 INCHES OF TAPE. FOR PE, EIGHT OUT OF NINE DATA TRACKS WENT DEAD FOR LESS THAN .1 INCHES BEFORE A VALID POSTAMBLE WAS DETECTED.
11	TIG	4	TRASH IN THE GAP. NON-ERASED DATA WAS DETECTED IN A GAP DURING A READ, WRITE, WRITE TAPE MARK, OR ERASE COMMAND.
10	DBF	4	DESKEW BUFFER FAIL. ONE OF THE DESKEW BUFFERS FAILED TO ASSERT "OUTPUT READY" WITHIN 20 MICROSECONDS AFTER BEING ENABLED. THE DEAD TRACK BITS WILL INDICATE ON WHICH TRACKS THIS FAILURE OCCURRED. THIS ERROR IS PROBABLY A RESULT OF A BROKEN FOR MATTER.
	NZO	4	NRZ FIFO OVERRUN.
09	SCK	4	SPEED CHECK. TAPE SPEED WAS OFF BY MORE THAN 5% DURING A WRITE DATA OPERATION. NOTE THAT SPEED AVERAGED OVER 8 TICKS AND THE AVERAGE MUST BE OFF 5% TO CAUSE THIS ERROR.
06			NOT USED.
07	IPR	S,4	INVALID PREAMBLE. SET ON A PE DRIVE IF THE PREAMBLE APPEARS TO BE SHORTER THAN 36 CHARACTERS OR LONGER THAN 44 CHARACTERS. ALSO

1628				SET IF THE PREAMBLE IS INCORRECTLY ENCODED BEYOND THE FIFTEENTH CHARACTER IN READ OR THE TENTH CHARACTER IN READ AFTER WRITE.
1629				
1630				
1631				
1632	06	SYN	4	SYNCH FAILURE. SET ON A PE DRIVE IF THE FORMATTER WAS UNABLE TO ACHIEVE SYNCHRONIZATION IN THE PREAMBLE.
1633				
1634				
1635				
1636		DRP	4	NRZ RECORD DROPPED A CHARACTER (THE NEXT CHARACTER WAS TO BE CONSIDERED CRC).
1637				
1638				
1639	05	IPO	5,4	INVALID POSTAMBLE. SET ON A PE DRIVE DURING READ OR WRITE IF ANY OF THE FIRST 39 CHARACTERS OF THE POSTAMBLE ARE NOT READ CORRECTLY.
1640				
1641				
1642				
1643		ITM	5,4	ILLEGAL TAPE MARK FOR NRZ.
1644				
1645	04	IED	4	INVALID END DATA. FOR PE, EIGHT OUT OF NINE TRACKS WENT DEAD BEFORE THE POSTAMBLE WAS DETECTED.
1646				
1647				
1648		LRO	4	FOR NRZI, DATA WAS NOT DETECTED IN EITHER THE LRCC OR CRCC WINDOWS. (LRC WAS ZERO)
1649				
1650				
1651	03	POS	5,4	POSTAMBLE SHORT. SET ON PE DRIVES DURING A READ OR WRITE WHEN LESS THAN 38 ALL ZEROES CHARACTERS ARE READ FOLLOWING THE ALL ONES CHARACTER.
1652				
1653				
1654				
1655				
1656		NZN	5,4	NRZ NOISE RECORD (FEWER THAN 13(10) FRAMES).
1657				
1658	02	POL	4	POSTAMBLE LONG. SET ON PE DRIVES DURING READ OR WRITE OPERATIONS WHEN THE POSTAMBLE EXCEEDS 42 CHARACTERS.
1659				
1660				
1661				
1662		LRC	4	LRC ERROR. SET ON NRZI DRIVES WHEN THE LRCC CHARACTER WAS FOUND IN ERROR.
1663				
1664				
1665	01	UNC	4	UNCORRECTABLE DATA. SET ON PE DRIVES WHEN A PARITY ERROR OCCURRED WITHOUT A CORRESPONDING DEAD TRACK INDICATION.
1666				
1667				
1668				
1669		CRC	4	CRC ERROR. SET ON NRZI DRIVES WHEN THE CRC CHARACTER WAS FOUND TO BE IN ERROR.
1670				
1671				
1672	00	MTE	4	MULTI-TRACK ERROR. SET ON PE DRIVES WHEN MORE THAN ONE DEAD TRACK OCCURRED IN THE PREAMBLE OR IN THE DATA FIELD.
1673				
1674				
1675				
1676		VPE	4	VERTICAL PARITY ERROR. SET ON NRZI DRIVES WHEN A CHARACTER DID NOT CONTAIN AN ODD NUMBER OF ONE BITS.
1677				
1678				

C4

6.3.5 EXTENDED STATUS REGISTER 2 (XSTAT2)



1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734

BIT	NAME	TCC	DEFINITION
15	OPM	S	OPERATION IN PROGRESS. (TAPE MOVING)
14	SIP	7,F2	SILO PARITY ERROR. CAUSES FATAL CLASS 2 BECAUSE THE ERROR MIGHT HAVE OCCURRED DURING THE TRANSMISSION OF THE MESSAGE PACKET.
13	BPE	7,F2	SERIAL BUS PARITY ERROR AT DRIVE. SET BY THE TSO4 WHEN A PARITY ERROR IS DETECTED ON DATA TRANSMITTED FROM THE TSI1 TO THE TSO4. CAUSES FATAL CLASS 2 BECAUSE THE ERROR MIGHT HAVE OCCURRED DURING THE TRANSMISSION OF THE MESSAGE PACKET.
12	CAF	7	CAPSTAN ACCELERATION FAIL. AFTER ACCELERATING TAPE FOR .2 INCHES, THE TAPE SPEED WAS CHECKED AND FOUND TO BE OUT OF TOLERANCE BY MORE THAN 10%.
11			NOT USED.
10	WCF	7	THE WRITE BOARD IS NOT EMPTYING THE I/O SILO AT THE PROPER RATE. THIS ERROR CAN BE THE RESULT OF THE WRITE BOARD CLOCK NOT BEING TURNED ON (BROKEN HARDWARE).
09			NOT USED.
08	DTP	S	DEAD TRACK PARITY. THE BITS DTP THROUGH DTD INDICATE WHICH TRACK(S) WENT DEAD, IF ANY, DURING THE LAST DATA TRANSFER OPERATION. IF DFSKEW BUFFER FAIL (DBF) IS SET, THESE BITS INDICATE WHICH CHANNEL FAILED.
07	DT7	S	DEAD TRACK 7. SEE DTP.
06	DT6	S	DEAD TRACK 6. SEE DTP.
05	DT5	S	DEAD TRACK 5. SEE DTP.
04	DT4	S	DEAD TRACK 4. SEE DTP.
03	DT3	S	DEAD TRACK 3. SEE DTP.

1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744

02 DT2 S DEAD TRACK 2. SEE DTP.  
01 DT1 S DEAD TRACK 1. SEE DTP.  
00 DT0 S DEAD TRACK 0. SEE DTP.

NOTE: ON A SET CHARACTERISTICS COMMAND, THE UCODE LEVEL IS RETURNED  
IN DT7 THRU DT0. ON A GET STATUS COMMAND, THE RESIDUAL CAPSTAN  
TICK COUNT (INTERNALLY R7) IS RETURNED IN DT7 THRU DT0.





1801  
 1802  
 1803  
 1804  
 1805  
 1806  
 1807  
 1808  
 1809  
 1810  
 1811  
 1812  
 1813  
 1814  
 1815  
 1816  
 1817  
 1818  
 1819  
 1820  
 1821  
 1822  
 1823  
 1824  
 1825  
 1826  
 1827  
 1828  
 1829  
 1830  
 1831  
 1832  
 1833  
 1834  
 1835  
 1836  
 1837  
 1838  
 1839  
 1840  
 1841  
 1842  
 1843  
 1844  
 1845  
 1846  
 1847  
 1848  
 1849  
 1850  
 1851  
 1852  
 1853  
 1854  
 1855  
 1856

PE: AT LEAST 24 CHARACTERS IN A ROW THAT DO NOT QUALIFY AS A TAPE MARK OR A DATA PREAMBLE.

01 LXS 5 LIMIT EXCEEDED STATICALLY. THIS BIT IS SET ANY TIME THE LIMIT SWITCHES ARE EXCEEDED. THIS BIT CAN ONLY BE CLEARED BY MANUALLY LOADING THE TAPE.

00 RIB 2 REVERSE INTO BOT. A READ, SPACE, OR SKIP COMMAND ALREADY IN PROGRESS HAS ENCOUNTERED THE BOT MARKER WHEN MOVING TAPE IN THE REVERSE DIRECTION. TAPE MOTION WILL BE HALTED AT BOT.

MICRO DIAGNOSTIC ERROR CODES

FOLLOWING IS A LIST OF THE ERRORS WHICH ARE DISPLAYED IN THE MICRO DIAGNOSTIC ERROR CODE (XSTAT3 BITS 15 - 08) AND ALSO IN THE LIGHTS ON THE TSO4 CONTROL PANEL, DUE TO FAILURES ON THE CAPSTAN BOARD, I/O BOARDS, WRITE BOARD, READ BOARD, OR FORMATTER BOARD. THE MICRO WILL BE IN A TIGHT LOOP IN THE DISPM PROGRAM, WAITING FOR OPERATOR OR CPU INTERVENTION WHILE THE ERROR IS BEING DISPLAYED IN THE CONSOLE LIGHTS. IT IS APPARENT THAT AN ERROR IS BEING DISPLAYED IF THE "UOK" LIGHT IS NOT LIGHTED, THE PROCESSOR IS NOT STOPPED, AND AN OCTAL NUMBER (100-377) IS BEING DISPLAYED IN THE LIGHTS. TO SCOPE LOOP THESE TESTS, ENTER MAINTENANCE MODE (ON-LINE SWITCH TO "OFF" POSITION, MAINTENANCE SWITCH UP, PRESS RESET), ENTER THE OFF-LINE TEST NUMBER (SEE SCOPE LOOP COLUMN BELOW) IN THE OPERATOR CONSOLE LIGHTS (ENTER ONE, WITH LEFT MOST SWITCH, ENTER ZEROES WITH RIGHT-MOST SWITCH), AND PRESS ON LINE BUTTON. TEST WILL LOOP UNTIL ON-LINE SWITCH IS RETURNED TO OFF LINE POSITION. ERRORS WILL BE DISPLAYED CONTINUOUSLY.

ERROR PROGRAM (DISPLAY)	ERROR DESCRIPTION	LIKELY MODULE	SCOPE LOOP
337 OPERATIONAL CODE	CAPSTAN RUNAWAY ERROR (M3.RNY). CAPSTAN DIDN'T STOP WITHIN ACCEPTABLE WINDOW AFTER LAST COMMAND.		
100 IOTSM	BASIC I/O MICRO FAILURE (PARITY ERROR, IOATN, HANDSHAKING, AND DATA WINDOW TEST BETWEEN THE I/O AND MAIN MICROS.  NOTE: CAN ALSO BE CAUSED BY THE SERIAL BUS .SMIN (SHIFT IN) STUCK ASSERTED.	M8967	14
101 IOTSM	ERROR IN I/O CONTROL REGISTER TEST	M8966 M8967	15

1857						
1858	102	IOTSM	FAILURE OF FRAME COUNTER TEST	M8966	15	
1859						
1860	103	IOISM	FAILURE OF I/O SILO NON PARITY ERROR	M8966	16	
1861			DATA TEST OR THE WRITE FLAG.	M8963		
1862						
1863	104	IOTSM	FAILURE OF I/O SILO PARITY ERROR	M8966	17	
1864			TEST OR DATA LATE TEST.			
1865						
1866	105	IOTSM	FAILURE OF SHIFT LOOP WITH ZEROES.	M8965	20	
1867						
1868	106	IOTSM	FAILURE OF SHIFT LOOP WITH ONES.	M8965	21	
1869						
1870	107	IOTSM	FAILURE OF SHIFT LENGTH MUX.	M8965	22	
1871						
1872	110	IOTSM	FAILURE TO RECEIVE CORRECT OP-CODE	M8965	47	
1873			FROM TS11 WHEN WE SENT DATA OVER	TS11		
1874			THE SERIAL BUS.	MOTHER BD		
1875				SBUS CABLE		
1876						
1877	111	CATSM	FAILURE OF 1 KHZ CLOCK TEST.	G159	2	
1878			TSTS TAC SYNC FLOP AND ATTN, TOO.	CBUS CABLE		
1879				M8963		
1880						
1881	112	CATSM	LIGHT REGISTER CHANGED WHEN MOTION	G159	3.4	
1882			REGISTER WAS CLEARED.			
1883						
1884	113	CATSM	FWD OR MVG BITS WRONG AFTER 1 TICK	G159	3.4	
1885			OF SIMULATED COMMAND AND TACH PULSES.			
1886						
1887	114	CATSM	FAILURE OF SIMULATED CAPSTAN	G159	3.4	
1888			SPEED TEST. THE CAPSTAN SPEED			
1889			COUNTER WAS OUT OF RANGE WHEN			
1890			TAPE MOTION AT SPEED WAS			
1891			SIMULATED.			
1892						
1893	115	CATSM	FAILURE OF SIMULATED SLOW CAPSTAN	G159	3.4	
1894			TEST. SPEED COUNTER DID NOT LATCH			
1895			UP WITH MAX COUNT WHEN SLOW TACH			
1896			TICKS WERE SIMULATED.			
1897						
1898	116	CATSM	FAILURE OF SIMULATED CAPSTAN DECEL	G159	3.4	
1899			TEST. COUNTER NOT ZERO FOR FORWARD			
1900			OR 377 FOR REVERSE WHILE DECELERATING.			
1901			OR MVG BIT NOT 1.			
1902						
1903	117	CATSM	FAILURE OF MOVING FLOP TO GO TO ZERO	G159	3.4	
1904			AFTER STOPPING (DIRECTION REVERSAL			
1905			FOR ONE TACH TICK).			
1906						
1907	120	PETSM	FAILURE OF WRITE BOARD TO TURN ON	M8929	23	
1908			AND EMPTY THE SILO. OR DATA LATE	M8966		
1909			BIT DOESN'T WORK.			
1910						
1911	121	PETSM	FAILURE OF WRITE BOARD TO EMPTY	M8929	23	
1912			SILO AT CORRECT SPEED.			

1913						
1914						
1915						
1916						
1917						
1918						
1919						
1920						
1921						
1922						
1923						
1924						
1925						
1926						
1927						
1928						
	124	PETSM	FORMATTER FLAG DOESN'T WORK ON THE M8922.	M8922	24	
	125	PETSM	FORMATTER SILO FILLING AND DATA ERROR	M8922 M8923 M8924	24	
	126	PETSM	PEAK SHIFT TEST ERROR	M8922 M8923 M8924	25	
	127	PETSM	FORMATTER TABLE LOOKUP ROM CHECKSUM TEST ERROR	M8922 M8923 M8924	26	
	\					

1929			.TITLE PROGRAM HEADER AND TABLES		
1930			.SBTTL PROGRAM HEADER		
1931					
1932			.ENABL ABS,AMA		
1933	002000	002000	" 2000		
1934	002000		BGNMOD		
1935					
1936			;		
1937			;		
1938			THE PROGRAM HEADER IS THE INTERFACE BETWEEN		
1939			THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.		
1940			;		
1941	002000		POINTER BGNRPT,BGNSW,BGNSFT,BGNAU,BGNDU,BGNSETUP		
1942					
1943					
1944	002000		HEADER CZTSH,D,0,5000,1,#INTPRI		
1945	002000		L\$NAME:: ;DIAGNOSTIC NAME		
1946	002000	103			.ASCII /C/
1947	002001	132			.ASCII /Z/
1948	002002	124			.ASCII /T/
1949	002003	123			.ASCII /S/
1950	002004	110			.ASCII /M/
1951	002005	000			.BYTE 0
1952	002006	000			.BYTE 0
1953	002007	000			.BYTE 0
1954	002010		L\$REV:: ;REVISION LEVEL		
1955	002010	104			.ASCII /D/
1956	002011		L\$DEPO:: ;0		
1957	002011	060			.ASCII /O/
1958	002012		L\$UNIT:: ;NUMBER OF UNITS		
1959	002012	000001			.WORD T\$PTMV
1960	002014		L\$TIML:: ;LONGEST TEST TIME		
1961	002014	005000			.WORD 5000
1962	002016		L\$HPCP:: ;POINTER TO H.W. QUES.		
1963	002016	025266			.WORD L\$HARD
1964	002020		L\$SPCP:: ;POINTER TO S.W. QUES.		
1965	002020	025340			.WORD L\$SOFT
1966	002022		L\$HPTP:: ;PTR. TO DEF. H.W. PTABLE		
1967	002022	002174			.WORD L\$HW
1968	002024		L\$SPTP:: ;PTR. TO S.W. PTABLE		
1969	002024	002202			.WORD L\$SW
1970	002026		L\$LADP:: ;DIAG. END ADDRESS		
1971	002026	026746			.WORD L\$LAST
1972	002030		L\$STA:: ;RESERVED FOR APT STATS		
1973	002030	000000			.WORD 0
1974	002032		L\$CO::		
1975	002032	000000			.WORD 0
1976	002034		L\$DTYP:: ;DIAGNOSTIC TYPE		
1977	002034	000001			.WORD 1
1978	002036		L\$APT:: ;APT EXPANSION		
1979	002036	000000			.WORD 0
1980	002040		L\$DTP:: ;PTR. TO DISPATCH TABLE		
1981	002040	002124			.WORD L\$DISPA
1982	002042		L\$PRIO:: ;DIAGNOSTIC RUN PRIORITY		
1983	002042	000340			.WORD #INTPRI
1984	002044		L\$ENVI:: ;FLAGS DESCRIBE HOW IT WAS SETUP		

J4

1985	002044	000000			.WORD	0
1986	002046		L\$EXP1::	;EXPANSION WORD		
1987	002046	000000			.WORD	0
1988	002050		L\$MREV::	;SVC REV AND EDIT #		
1989	002050	003			.BYTE	C\$REVISI
1990	002051	003			.BYTE	C\$EDIT
1991	002052		L\$EF::	;DIAG. EVENT FLAGS		
1992	002052	000000			.WORD	0
1993	002054	000000			.WORD	0
1994	002056		L\$SPC::			
1995	002056	000000			.WORD	0
1996	002060		L\$DEVP::	; POINTER TO DEVICE TYPE LIST		
1997	002060	002164			.WORD	L\$DVTYP
1998	002062		L\$REPP::	;PTR. TO REPORT CODE		
1999	002062	016150			.WORD	L\$RPT
2000	002064		L\$EXP4::			
2001	002064	000000			.WORD	0
2002	002066		L\$EXP5::			
2003	002066	000000			.WORD	0
2004	002070		L\$AUT::	;PTR. TO ADD UNIT CODE		
2005	002070	021770			.WORD	L\$AU
2006	002072		L\$DUT::	;PTR. TO DROP UNIT CODE		
2007	002072	021716			.WORD	L\$DU
2008	002074		L\$LUN::	;LUN FOR EXERCISERS TO FILL		
2009	002074	000000			.WORD	0
2010	002076		L\$DESP::	;POINTER TO DIAG. DESCRIPTION		
2011	002076	002136			.WORD	L\$DESC
2012	002100		L\$LOAD::	;GENERATE SPECIAL AUTOLOAD EMT		
2013	002100	104035			EMT	E\$LOAD
2014	002102		L\$ETP::	;POINTER TO ERRTBL		
2015	002102	000000			.WORD	0
2016	002104		L\$ICP::	;PTR. TO INIT CODE		
2017	002104	017704			.WORD	L\$INIT
2018	002106		L\$CCP::	;PTR. TO CLEAN-UP CODE		
2019	002106	021654			.WORD	L\$CLEAN
2020	002110		L\$ACP::	;PTR. TO AUTO CODE		
2021	002110	021232			.WORD	L\$AUTO
2022	002112		L\$PRT::	;PTR. TO PROTECT TABLE		
2023	002112	017676			.WORD	L\$PROT
2024	002114		L\$TEST::	;TEST NUMBER		
2025	002114	000000			.WORD	0
2026	002116		L\$DLY::	;DELAY COUNT		
2027	002116	000000			.WORD	0
2028	002120		L\$HIME::	;PTR. TO HIGH MEM		
2029	002120	000000			.WORD	0
2030						



2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083

002172  
002172 000002  
002174  
002174  
  
002174 172522  
002176 000224  
  
002200  
002200

.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 RUN TIME P TABLE.  
---

BGNHW DFPTBL

.WORD L10000-L

L\$HW::  
DFPTBL::

172522 ;TSSR ADDRESS.  
224 ;VECTOR ADDRESS.

ENDHW  
L10000:

```

2084 .SBTTL SOFTWARE P-TABLE
2085
2086 ;**
2087 ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
2088 ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
2089 ;--
2090
2091 002200          BGNSW  SFPTBL
2092 002200 000043
2093 002202
2094 002202
2095
2096 002202 001
2097 002203 000
2098 002204 000
2099 002205 000
2100 002206 001
2101 002207 000
2102 002210 000
2103 002211 000
2104 002212 000
2105 002213 000
2106 002214 000
2107 002215 000
2108 002216 000040

```

```

L$SW::
SFPTBL::
CLRFLG:: .BYTE 1 ;CLEAR COUNTERS FLAG.
RRANV:: .BYTE 0 ;RESET RANDOM VARIABLES EACH PASS FLAG.
HAE:: .BYTE 0 ;HALT AFTER EACH COMMAND FLAG.
ERCVR:: .BYTE 0 ;ENABLE RECOVERABLE ERROR PRINTS FLAG.
BADTSW:: .BYTE 1 ;BAD TAPE SWITCH TO REWRITE ON SAME SPOT & DETEC
          .BYTE 0 ;SPARE
DINT:: .BYTE 0 ;DISABLE INTERRUPTS FLAG.
IREC:: .BYTE 0 ;INHIBIT ERROR RECOVERY FLAG.
CHGFLG:: .BYTE 0 ;CHANGE CMD SEQ TABLE FLAG.
          .BYTE 0 ;SPARE.
PIRE:: .BYTE 0 ;INHIBIT RESIDUAL FRAMECOUNT ERROR REPORT FLAG.
          .BYTE 0 ;SPARE.
CHAR:: CH.EAI ;CHARACTERISTICS CODE (DEFAULT = 40).

```

```

          .WORD L10001

```



2109	002220	000015	CMDD::	.WORD	13.	;COMMAND 2 (DEFAULT = REWIND).
2110	002222	000001		.WORD	1	;BYTE COUNT
2111	002224	000001		.WORD	1	;NUMBER OF OPERATIONS
2112	002226	000007		.WORD	RANP	;PATTERN
2113	002230	000004		.WORD	4	;COMMAND 3 (DEFAULT = WRITE)
2114	002232	004000		.WORD	DATCNT	;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
2115	002234	076400		.WORD	32000.	;NUMBER OF OPERATIONS (DEFAULT = 32000).
2116	002236	000007		.WORD	RANP	;PATTERN (DEFAULT = RANDOM).
2117	002240	000003		.WORD	3	;COMMAND 4 (DEFAULT = READ REV).
2118	002242	004000		.WORD	DATCNT	;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
2119	002244	076400		.WORD	32000.	;NUMBER OF OPERATIONS (DEFAULT = 32,000).
2120	002246	000007		.WORD	RANP	;PATTERN (DEFAULT = RANDOM).
2121	002250	000002		.WORD	2	;COMMAND 5 (DEFAULT = READ FWD).
2122	002252	004000		.WORD	DATCNT	;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
2123	002254	076400		.WORD	32000.	;NUMBER OF OPERATIONS (DEFAULT = 32,000).
2124	002256	000007		.WORD	RANP	;PATTERN (DEFAULT = RANDOM).
2125	002260	000015		.WORD	13.	;COMMAND 6 (DEFAULT = REWIND).
2126	002262	000001		.WORD	1	;BYTE COUNT
2127	002264	000001		.WORD	1	;NUMBER OF OPERATIONS
2128	002266	000007		.WORD	RANP	;PATTERN
2129	002270	000033		.WORD	27.	;END OF CMD SEQ TABLE CODE (DEF) OR CMD 7
2130	002272	004000		.WORD	DATCNT	;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
2131	002274	076400		.WORD	32000.	;NUMBER OF OPERATIONS (DEFAULT = 32000).
2132	002276	000007		.WORD	RANP	;PATTERN (DEFAULT = RANDOM).
2133	002300	000033		.WORD	27.	;END OF CMD SEQ TABLE CODE (DEF) OR CMD 8
2134	002302	004000		.WORD	DATCNT	;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
2135	002304	076400		.WORD	32000.	;NUMBER OF OPERATIONS (DEFAULT = 32000).
2136	002306	000007		.WORD	RANP	;PATTERN (DEFAULT = RANDOM).
2137						
2138	002310				ENDSW	
2139	002310		.10001:			
2140						
2141	002310				ENDMOD	

```
2142  
2143  
2144 .TITLE GLOBAL AREAS  
2145 .SBTTL GLOBAL EQUATES SECTION  
2146 002310 BGNMOD  
2147  
2148 ;  
2149 ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT  
2150 ; ARE USED IN MORE THAN ONE TEST.  
2151 ;  
2152  
2153 002310 EQUALS  
2154 ;  
2155 ; BIT DEFINITIONS  
2156 ;  
2157 100000 BIT15.. 100000  
2158 040000 BIT14.. 40000  
2159 020000 BIT13.. 20000  
2160 010000 BIT12.. 10000  
2161 004000 BIT11.. 4000  
2162 002000 BIT10.. 2000  
2163 001000 BIT09.. 1000  
2164 000400 BIT08.. 400  
2165 000200 BIT07.. 200  
2166 000100 BIT06.. 100  
2167 000040 BIT05.. 40  
2168 000020 BIT04.. 20  
2169 000010 BIT03.. 10  
2170 000004 BIT02.. 4  
2171 000002 BIT01.. 2  
2172 000001 BIT00.. 1  
2173 ;  
2174 001000 BIT9.. BIT09  
2175 000400 BIT8.. BIT08  
2176 000200 BIT7.. BIT07  
2177 000100 BIT6.. BIT06  
2178 000040 BIT5.. BIT05  
2179 000020 BIT4.. BIT04  
2180 000010 BIT3.. BIT03  
2181 000004 BIT2.. BIT02  
2182 000002 BIT1.. BIT01  
2183 000001 BIT0.. BIT00  
2184 ;  
2185 ; EVENT FLAG DEFINITIONS  
2186 ; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION  
2187 ;  
2188 000040 EF.START.. 32. ; START COMMAND WAS ISSUED  
2189 000037 EF.RESTART.. 31. ; RESTART COMMAND WAS ISSUED  
2190 000036 EF.CONTINUE.. 30. ; CONTINUE COMMAND WAS ISSUED  
2191 000035 EF.NEW.. 29. ; A NEW PASS HAS BEEN STARTED  
2192 000034 EF.PWR.. 28. ; A POWER FAIL POWER UP OCCURRED  
2193 ;  
2194 ;  
2195 ; PRIORITY LEVEL DEFINITIONS  
2196 ;  
2197 000340 PRI07.. 340
```



()

```

2249                                     ;THE FOLLOWING ARE BIT DEFINITIONS FOR THE COMMAND WORD
2250
2251      100000      ACK.C**100000      ;ACKNOWLEDGE BIT
2252      040000      CVC.C**40000      ;CLEAR VOLUME CHECK.
2253      020000      OPP.C**20000      ;OPPOSITE BIT
2254      010000      SWB.C**10000      ;SWAP BYTE BIT
2255      004000      MOD.C3**4000      ;MODE BIT 3
2256      004000      BRP.C**4000      ;BYTE/RECORD/FILE COUNT FLAG BIT. NOT USED
2257                                     ;BY TSO4 BUT USED INTERNALLY BY THIS PROGRAM ONL
2258      002000      MOD.C2**2000      ;MODE BIT 2
2259      001000      MOD.C1**1000      ;MODE BIT 1
2260      000400      MOD.C0**400      ;MODE BIT 0
2261      000200      IE.C**200      ;INTERRUPT ENABLE
2262      000100      FMT.C1**100      ;FORMAT BIT 1
2263      000100      VFY.C**100      ;WRITE VERIFY FLAG BIT. INTERNAL USE ONLY.
2264                                     ;NOT USED BY TSO4.
2265      000040      FMT.C0**40      ;FORMAT BIT 0.
2266      000040      JMP.C**40      ;JUMP BIT TO DIRECT THIS PROGRAM TO JUMP TO
2267                                     ;A CERTAIN LOCATION IN THE COMMAND SEQUENCE
2268                                     ;TABLE. INTERNAL USE ONLY.
2269      000020      CMD.C4**20      ;COMMAND BIT 4
2270      000020      DLY.C**20      ;INSERT DELAY. INTERNAL USE ONLY.
2271      000010      CMD.C3**10      ;COMMAND BIT 3
2272      000004      CMD.C2**4      ;COMMAND BIT 2
2273      000002      CMD.C1**2      ;COMMAND BIT 1
2274      000001      CMD.C0**1      ;COMMAND BIT 0
2275
2276                                     ; BIT DEFINITIONS FOR DEVICE CHARACTERISTICS.
2277
2278      000200      CH.ESS**200      ;ENABLE SKIP TAPE MARKS STOP (STOP AT LOGICAL E0
2279      000040      CH.EAI**40      ;ENABLE ATTENTION INTERRUPTS.
2280      000020      CH.ERI**20      ;ENABLE MESSAGE BUFFER RELEASE INTERRUPTS.
2281      000040      DFTSCH**CH.EAI      ;DEFAULT CHARACTERISTICS CODE.
2282
2283                                     ;THE FOLLOWING INDICATES THE RELATIVE POSITIONS OF THE STATUS WORDS
2284                                     ;IN THE MESSAGE BUFFER.
2285
2286      000004      MS.RFC**4      ;RESIDUAL FRAME COUNT.
2287      000006      MS.XS0**6      ;EXT STATUS REG 0
2288      000010      MS.XS1**10      ;EXT STATUS REG 1
2289      000012      MS.XS2**12      ;EXT STATUS REG 2
2290      000014      MS.XS3**14      ;EXT STATUS REG 3
2291
2292                                     ;THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0.
2293
2294      100000      X0.TMK**100000      ;TAPE MARK.
2295      040000      X0.RLS**40000      ;RECORD LENGTH SHORT.
2296      020000      X0.LET**20000      ;LOGICAL EOT.
2297      010000      X0.RLL**10000      ;RECORD LENGTH LONG.
2298      000100      X0.ONL**100      ;ON LINE BIT.
2299      000002      X0.BOT**2      ;BOT BIT.
2300      000001      X0.EOT**1      ;EOT BIT.
2301
2302                                     ;THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2.
2303
2304      100000      X2.OPM**100000      ;OPERATION IN PROGRESS. TAPE MOVING

```

```

2305
2306                ;THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3.
2307
2308                000010                X3.DCK==10                ;DENSITY CHECK.
2309                157400                X3.RNY==157400            ;CAPSTAN RUNAWAY UDIAG ERROR CODE.
2310
2311                ;THE FOLLOWING DEFINITIONS SHOW THE RELATIVE POSITIONS OF THE COMMAND
2312                ;PACKET ENTRIES.
2313
2314                000000                CP.CMD==0                ;CMDPKT.0==TS04 COMMAND.
2315                000002                CP.ADL==2                ;CMDPKT.2==BUFFER ADDRESS LOW.
2316                000004                CP.ADH==4                ;CMDPKT.4==BUFFER ADDRESS HIGH.
2317                000006                CP.CNT==6                ;CKDPKT.6==BYTE/FILE/RECORD COUNT
2318
2319                ;
2320                ;      MISCELLANEOUS DEFINITIONS.
2321
2321                000340                INTPRI==PRI07            ;PRIORITY TO BE USED IN INTERRUPT STATE.
2322                002452                TSBA==TSDB              ;DATA BUFFER ADDRESS REGISTER.
2323                000010                SCHCNT==10              ;ARBITRARY BYTE LENGTH FOR CHARACTERISTIC
2324                ;BUFFER LENGTH. (EVEN #)
2325                000016                MSGCNT==16              ;MESSAGE BUFFER LENGTH IN BYTES. (EVEN #)
2326                003334                DIABLK==DATAWT          ;WRITE BUFFER ALSO USED FOR DIAG CMD.
2327                000020                DIACNT==20              ;DIAGNOSTIC COMMAND BUFFER EXTENT.
2328                004000                DATCNT==2048.           ;MAXIMUM RECORD LENGTH IN BYTES.
2329                ;THIS COUNT SHOULD BE A MULTIPLE OF 256 TO INSUR
2330                ;PROPER READ/WRITE BUFFER ALLOCATION BY THE SUPE
2331                000550                CNTLEN==CNTEND CNTBGN    ;LENGTH OF STATISTICAL COUNTER AREA.
2332                177740                RNOPSC==177740          ;RANDOM # OF OPERATIONS MASK.
2333                000007                RANP==7                ;CODE TO SELECT RANDOM PATTERN.
2334                000020                RRECL==16.              ;READ RECOVERY ATTEMPT LIMIT.
2335                000020                WRECL==16.              ;WRITE RECOVERY ATTEMPT LIMIT.
2336                153624                RANBC==153624           ;CONSTANT USED TO RESET RANDOM # GENERATOR BASE.
2337                032561                RANSC==32561           ;CONSTANT USED TO RESET RANDOM # SAVE LOCATION.
2338                177774                NINUSE==17 774         ;NOT IN USE CODE FOR DEVICE STATE TABLE.
2339                177740                NCMD.C==ACK.C!CVC.C!OPP.C!SWB.C!MOD.C3!MOD.C2!MOD.C1!MOD.CO!IE.C!FMT.C1!FMT.CO
2340                ;NOT "COMMAND" BITS.
2341
2342                ;THE FOLLOWING DEFINES THE COMMAND WORD FOR EACH TS04 COMMAND.
2343
2344                100013                DRI==  ACK.C!CMD.C3!CMD.C1!CMD.CO
2345                ;DRIVE INIT.
2346
2347                104001                RDF==  ACK.C!BRF.C!CMD.CO
2348                ;READ FORWARD
2349
2350                104401                RDR==  ACK.C!BRF.C!MOD.CO!CMD.CO
2351                ;READ REVERSE
2352
2353                104005                WRT==  ACK.C!BRF.C!CMD.CO!CMD.C2
2354                ;WRITE COMMAND
2355
2356                104105                WTV==  ACK.C!BRF.C!VFY.C!CMD.CO!CMD.C2
2357                ;WRITE VERIFY
2358
2359                104010                SRF =  ACK.C!BRF.C!CMD.C3
2360                ;SPACE RECORD FORWARD

```

GLOBAL AREAS  
CZTSMO.P11MAC111 30(1046)  
06 APR 84 08:4906 APR 84 08:51 PAGE 59  
GLOBAL EQUATES SECTION

SEQ 0057

2361			
2362	104410	SRR==	ACK.C!BRF.C!MOD.CO!CMD.C3
2363			;SPACE RECORD REVERSE
2364			
2365	105401	RNR=	ACK.C!BRF.C!MOD.C1!MOD.CO!CMD.CO
2366			;READ REV RETRY1 - REREAD NEXT REVERSE, IE. SPACE FWD, READ REVE
2367			
2368	125401	RNF==	ACK.C!BRF.C!OPP.C!MOD.C1!MOD.CO!CMD.CO
2369			;READ REV RETRY2 - REREAD NEXT FORWARD, IE. READ FORWARD, SPACE R
2370			
2371	105001	RPF=	ACK.C!BRF.C!MOD.C1!CMD.CO
2372			;READ FWD RETRY1 - REREAD PREVIOUS FORWARD, IE. SPACE REVERSE, R
2373			
2374	125001	RPR==	ACK.C!BRF.C!OPP.C!MOD.C1!CMD.CO
2375			;READ FWD RETRY2 REREAD PREVIOUS REVERSE, IF. READ REVERSE, 3P
2376			
2377	105005	WRR==	ACK.C!MOD.C1!BRF.C!CMD.C2!CMD.CO
2378			;WRITE RETRY
2379			
2380	102010	RWD==	ACK.C!MOD.C2!CMD.C3
2381			;REWIND COMMAND
2382			
2383	100012	MBR==	ACK.C!CMD.C3!CMD.C1
2384			;MESSAGE BUFFER RELEASE
2385			
2386	100011	WTM==	ACK.C!CMD.C3!CMD.CO
2387			;WRITE TAPE MARK.

2388				
2389	101011	WTR==	ACK.C!MOD.C1!CMD.C3!CMD.CO	
2390				;WRITE TAPE MARK RETRY.
2391				
2392	105010	SFF==	ACK.C!BRF.C!MOD.C1!CMD.C3	
2393				;SPACE FILE FORWARD
2394				
2395	105410	SFR==	ACK.C!BRF.C!MOD.CO!MOD.C1!CMD.C3	
2396				;SPACE FILE REVERSE
2397				
2398	100017	GES==	ACK.C!CMD.CO!CMD.C1!CMD.C2!CMD.C3	
2399				;GET EXTENDED STATUS
2400				
2401	100411	ERS==	ACK.C!MOD.CO!CMD.C3!CMD.CO	
2402				;ERASE 3 INCHES OF TAPE
2403				
2404	100412	UNL==	ACK.C!MOD.CO!CMD.C3!CMD.C1	
2405				;UNLOAD COMMAND
2406				
2407	101012	CLN==	ACK.C!MOD.C1!CMD.C3!CMD.C1	
2408				;ERASE TAPE.
2409				
2410	140004	SCH==	ACK.C!CVC.C!CMD.C2	;SET DEVICE CHARACTERISTICS.
2411				
2412	100006	DIA==	ACK.C!CMD.C2!CMD.C1	;DIAGNOSTICS.
2413				
2414	000040	JMP==	JMP.C	;JUMP TO 'N' TH COMMAND
2415				
2416	000020	DLY==	DLY.C	;DELAY 'N' MS.
2417				
2418	177777	END==	177777	;END OF COMMAND SEQUENCES

2419  
2420  
2421  
2422  
2423  
2424  
2425  
2426  
2427  
2428  
2429  
2430  
2431  
2432  
2433  
2434  
2435  
2436  
2437  
2438  
2439  
2440  
2441  
2442  
2443  
2444  
2445  
2446  
2447  
2448  
2449  
2450  
2451  
2452  
2453  
2454  
2455  
2456  
2457  
2458  
2459  
2460  
2461  
2462  
2463  
2464  
2465

002310 000000  
002312 000000  
002314 000000  
002316 000000  
  
002320 100017  
  
002324 100012  
  
002330 102010  
002332 000001  
  
002334 000007

```
.SBTTL GLOBAL DATA SECTION

;
;
; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
; IN MORE THAN ONE TEST.
;

; COMMAND PACKET.

      .3E177774      ;MUST BE ON MOD 4 BOUNDARY.
CNDPKT:: 0           ;1ST WORD IS TSO4 COMMAND.
          0           ;2ND WORD IS THE BUFFER LOW ADDRESS.
          0           ;3RD WORD IS THE BUFFER HIGH ADDRESS.
          0           ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.

; GET STATUS COMMAND PACKET.

      .3E177774      ;MUST BE ON MOD 4 BOUNDARY.
GSCPKN:: .WORD      GES

; MESSAGE BUFFER RELEASE COMMAND PACKET.

      .3E177774      ;MUST BE ON MOD 4 BOUNDARY.
BRCPKN:: .WORD      MBR

; REWIND COMMAND PACKET (USED IN ERROR RECOVERY ONLY)

      .3E177774      ;MUST BE ON A MODULE 4 BOUNDARY.
RWCPKN:: .WORD      RWD
          .WORD      1

; WORK AREA FOR ANALYSIS OF MESSAGE PACKET CONTENTS.

MSGPKT:: .BLKW      7           ;1ST WORD:: MESSAGE TYPE.
                               ;2ND WORD:: DATA FIELD LENGTH.
                               ;3RD WORD:: RESIDUAL FRAME COUNT.
                               ;4TH WORD:: XSTAT0
                               ;5TH WORD:: XSTAT1
                               ;6TH WORD:: XSTAT2
                               ;7TH WORD:: XSTAT3
```



```

2466 ; MESSAGE PACKETS.
2467
2468 002352 000007 MSGPK0:: .BLKW 7 ;MESSAGE PACKET FOR DEVICE #0
2469 002370 000007 MSGPK1:: .BLKW 7 ;MESSAGE PACKET FOR DEVICE #1
2470 002406 000007 MSGPK2:: .BLKW 7 ;MESSAGE PACKET FOR DEVICE #2
2471 002424 000007 MSGPK3:: .BLKW 7 ;MESSAGE PACKET FOR DEVICE #3
2472
2473 ; SET CHARACTERISTIC BLOCK.
2474
2475 002442 002352 SCHBK:: MSGPK0 ;1ST WORD:: MSGPKT ADDR LO(SET UP BY EXECUTE ROUT
2476 002444 000000 0 ;2ND WORD:: MSGPKT ADDR HI.
2477 002446 000016 MSGCNT ;3RD WORD:: MSG BUFFER LENGTH (BYTES)
2478 002450 000040 CH.EAI ;4TH WORD:: CHARACTERISTICS WORD(SET BY SETUP RO
2479
2480 ; TS04 REGISTER ADDRESSES.
2481
2482 002452 000004 TSDB:: .BLKW 4 ;TS04 DATA BUFFER ADDRESSES.
2483 002462 000004 TSSR:: .BLKW 4 ;TS04 STATUS REGISTER ADDRESSES.
2484 002472 000004 TSVCT:: .BLKW 4 ;TS04 VECTOR ADDRESSES.
2485
2486 ; ADDRESSES OF MESSAGE PACKETS.
2487
2488 002502 002352 MSGPKA:: MSGPK0 ;DEVICE 0.
2489 002504 002370 MSGPK1 ;DEVICE 1.
2490 002506 002406 MSGPK2 ;DEVICE 2.
2491 002510 002424 MSGPK3 ;DEVICE 3.
2492
2493 ; ADDRESSES OF INTERRUPT HANDLING ROUTINES.
2494
2495 002512 006316 TS4INT:: TS4INO ;DEVICE 0.
2496 002514 006324 TS4IN1 ;DEVICE 1.
2497 002516 006332 TS4IN2 ;DEVICE 2.
2498 002520 006340 TS4IN3 ;DEVICE 3.
2499
2500 ; TS04 CODE LEVELS, WILL BE STORED AFTER SCH CMD IN BASIC FUNCTION TEST
2501
2502 002522 000000 TS4CL:: 0 ;DEVICE 0
2503 002524 000000 0 ;DEVICE 1
2504 002526 000000 0 ;DEVICE 2
2505 002530 000000 0 ;DEVICE 3
2506
2507 ; UNIT NUMBERS OF ALL DEVICES BEING TESTED(1-4).
2508 ; WHEN DEVICE IS NOT IN USE, IT'S LOCATION WILL = 3.
2509 ; R5 WILL ALWAYS CONTAIN THE PRESENT LOGICAL UNIT NUMBER X 2.
2510

```

J5

2511 002532 177774  
2512 002534 177774  
2513 002536 177774  
2514 002540 177774  
2515 002542 177777  
2516  
2517  
2518  
2519  
2520  
2521 002544 002774  
2522 002546 003046  
2523 002550 003120  
2524 002552 003172

DEVTBL:: .WORD NINUSE  
.WORD NINUSE  
.WORD NINUSE  
.WORD NINUSE  
.WORD END

; BAD TAPE TABLE POINTER: USED BY WRITE RETRY ROUTINE  
; 'WRTY" TO LOG BAD TAPE SPOTS ON UNITS UNDER TEST

BTADDR:: BT0  
BT1  
BT2  
BT3

145

```

2525 ; COUNTER AREA.
2526
2527 CNTBGN*.
2528 002554 000020 WRBC:: .BLKW 20 ;BYTES WRITTEN.
2529 002614 000020 RRBC:: .BLKW 20 ;BYTES READ REV.
2530 002654 000020 RFBC:: .BLKW 20 ;BYTES READ FWD.
2531 002714 000004 WRREC:: .BLKW 4 ;RECOVERABLE WRITE ERRORS.
2532 002724 000004 WRUNR:: .BLKW 4 ;UNRECOVERABLE WRITE ERRORS.
2533 002734 000004 RRREC:: .BLKW 4 ;RECOVERABLE READ REV ERRORS.
2534 002744 000004 RRUNR:: .BLKW 4 ;UNRECOVERABLE READ REV ERRORS.
2535 002754 000004 RFREC:: .BLKW 4 ;RECOVERABLE READ FWD ERRORS.
2536 002764 000004 RFUNR:: .BLKW 4 ;UNRECOVERABLE READ FWD ERRORS.
2537 002774 000025 BT0:: .BLKW 21. ;UNIT 0 BAT TAPE SPOTS LOG
2538 003046 000025 BT1:: .BLKW 21. ;UNIT 1 BAT TAPE SPOTS LOG
2539 003120 000025 BT2:: .BLKW 21. ;UNIT 2 BAT TAPE SPOTS LOG
2540 003172 000025 BT3:: .BLKW 21. ;UNIT 3 BAT TAPE SPOTS LOG
2541 003244 000004 WRTYCT:: .BLKW 4 ;WRITE RETRY COUNTER
2542 003254 000004 PASCNT:: .BLKW 4 ;PASS COUNT.
2543 003264 000004 SCCNT:: .BLKW 4 ;SPECIAL CONDITION COUNT.
2544 003274 000004 VFYCNT:: .BLKW 4 ;COUNT OF TSO4 DATA COMPARE ERRORS.
2545 003304 000004 HRDCNT:: .BLKW 4 ;COUNT OF HARD ERRORS.
2546 003314 000004 FTLCNT:: .BLKW 4 ;COUNT OF FATAL ERRORS.
2547 003324 000004 CNTEND*. ;END OF STATISTICAL COUNTERS.
2548 003324 000004 RECCNT:: .BLKW 4 ;NUMBER OF RECORDS FROM BOT; CLEARED ON REWIND
2549 ; AND WHEN RESTARTING OR CONTINUING TEST 2.
2550
2551
2552 ; THE FOLLOWING ARE THE DEFINITIONS OF VARIABLES
2553 ; USED BY THE PROGRAM.
2554
2555 003334 000000 DATAW:: .WORD 0 ;WRITE BUFFER ADDRESS.
2556 003336 000000 DATARD:: .WORD 0 ;READ BUFFER ADDRESS.
2557 003340 000000 NCNT:: .WORD 0 ;STORAGE FOR VALUE OF N.
2558 003342 000000 NCNT1:: .WORD 0 ;TEMP STORAGE FOR VALUE OF N.
2559 003344 000000 BRFCNT:: .WORD 0 ;STORAGE FOR BPCR VALUE.
2560 003346 177777 CMDWRD:: .WORD END ;CONTAINS COMMAND WORD BEING EXECUTED PRESENTLY.
2561 003350 177777 CMDSAV:: .WORD END ;SAVE LOCATION FOR CMD WORD DURING ERROR RECOVER
2562 003352 177777 PCMDWD:: .WORD END ;CONTAINS PREVIOUS COMMAND WORD.
2563 003354 000000 CMDLG:: .WORD 0 ;CURRENT COMMAND LOGGING CODE.
2564 003356 000000 LENMSK:: .WORD 0 ;RANDOM WRITE LENGTH MASK, TO BE SET UP BY TESTS
2565 003360 153624 RANB:: .WORD 153624 ;RANDOM # GENERATOR BASE.
2566 003362 032561 RANS:: .WORD 32561 ;RANDOM # SAVE LOCATION.
2567 003364 000000 TIME1:: .WORD 0 ;TIME COUNT 1.
2568 003366 000000 TIME2:: .WORD 0 ;TIME COUNT 2.
2569 003370 000000 JLOOP:: .WORD 0 ;JMP COMMAND LOOP COUNT.
2570 003372 000000 JLOC:: .WORD 0 ;JMP COMMAND LOCATION COUNT.
2571 003374 000000 PATERN:: .WORD 0 ;PATTERN SELECT CODE.
2572 003376 000000 CTCC:: .WORD 0 ;CURRENT TERMINATION CLASS CODE.
2573 003400 000000 RSSAVE:: .WORD 0 ;LOCATION FOR SAVING CURRENT DEVICE POINTER.
2574 003402 000000 TSSREG:: .WORD 0 ;CURRENT STATUS REGISTER.

```

```

2575 ; ERROR FLAG AREA, THESE FLAGS ARE CLEARED DURING INITIALIZATION AND
2576 ; AFTER EACH COMMAND IS COMPLETED.
2577
2578 003404 BGNFLG=.
2579 003404 000000 RETRYC:: .WORD 0 ;# OF RECOVERY ATTEMPTS EXECUTED.
2580 003406 000 RPTCNT:: .BYTE 0 ;WRITE REPEAT ON SAME SPOT CNTR: 4 PER WRITE RETRY
2581 003407 000 WRTYFG:: .BYTE 0 ;WRITE RETRY ON SAME SPOT IN PROGRESS FLAG
2582 003410 000 WRTYER:: .BYTE 0 ;WRITE RETRY ON SAME SPOT ERROR FLAG
2583 003411 000 RECLOG:: .BYTE 0 ;RECORD COUNT HAS BEEN UPDATED FOR THIS RECORD.
2584 003412 000 ERLOG:: .BYTE 0 ;DATA BYTES AND ERRORS HAVE BEEN LOGGED FOR THIS
2585 003413 000 RWERR:: .BYTE 0 ;READ/WRITE ERROR HAS OCCURED.
2586 003414 000 UNREC:: .BYTE 0 ;UNRECOVERABLE ERROR HAS OCCURED.
2587 003415 000 ERRREC:: .BYTE 0 ;ERROR RECOVERY MODE.
2588 .EVEN
2589 003416 ENDERF=.
2590
2591 ; ADDITIONAL FLAGS, THESE FLAGS ARE CLEARED DURING INITIALIZATION.
2592
2593 003416 000004 INTFLG:: .BLKW 4 ;INTERRUPT OCCURRED FLAGS FOR EACH DEVICE.
2594 003426 000004 EOTFLG:: .BLKW 4 ;EOT/BOT FLAGS FOR EACH DEVICE (XSTATO).
2595 003436 000000 BTPT:: .WORD 0 ;BAD TAPE SPOT POINTER TO BTO BT3 VIA BTADDR
2596 003440 000 EXPBOT:: .BYTE 0 ;BOT IS EXPECTED, DO NOT ABORT ON BOT/FUNC RTI.
2597 003441 000 RANDOM:: .BYTE 0 ;RANDOM EVERYTHING FLAG
2598 003442 000 VFYFLG:: .BYTE 0 ;SET DURING WRITE/VERIFY COMMAND.
2599 003443 000 RPTFLG:: .BYTE 0 ;PERFORMANCE REPORT HAS BEEN REQUESTED.
2600 003444 000 SWBFLG:: .BYTE 0 ;ENABLES SWAP BYTE FUNCTION WHEN NOT EQUAL TO ZE
2601 003445 000 IRE:: .BYTE 0 ;INHIBIT RESIDUAL FRAME COUNT ERROR REPORT.
2602 003446 000 DROPED:: .BYTE 0 ;CURRENT UNIT HAS BEEN DROPPED
2603 003447 000 T1SWB:: .BYTE 0 ;TEST1 SWAP BYTES FLAG
2604 003450 000 ALLEOT:: .BYTE 0 ;ALL UNITS @ EOT FLAG
2605 003451 000 ERSFLG:: .BYTE 0 ;ERASE FLAG: DO ERASE AFTER A SPACE REV TO DELE
2606 ;BADLY WRITTEN RECORD. 1 TO 4 ERASES LEAVING
2607 ;A 3 TO 12 INCH GAP MAY RESULT.
2608 .EVEN
2609 003452 ENDFLG=.
2610
2611 ; ADDITIONAL FLAGS, THESE FLAGS ARE CLEARED ONLY AFTER BEING CHECKED.
2612
2613 003452 000 STAF LG:: .BYTE 0 ;START FLAG SET BY INIT CODE IF STARTING.
2614 003453 000 PWRFLG:: .BYTE 0 ;POWER FAILURE FLAG - SET ONLY DURING INIT.
2615 003454 000 TRAPD4:: .BYTE 0 ;TRAPED AT 4 FLAG
2616 003455 000 MISCFG:: .BYTE 0 ;MISCELLANEOUS FLAG
2617
2618 ; OPERATOR FLAG SETTINGS PASSED BY DIAG. SUPERVISOR IN A 16 BIT WORD
2619 ; SEE GLOBAL EQUATES SECTION FOR FLAG BIT LIST
2620
2621 003456 000000 OPFLAG:: .WORD 0 ;READ ONLY OPERATOR FLAG WORD
2622 .EVEN
  
```



```

2657                                     ;THE FOLLOWING IS THE TS04 COMMAND TABLE
2658
2659 003562 100013 CMTDTBL:: .WORD DRI ;DRIVE INIT.
2660 003564 104001 .WORD RDF ;READ FORWARD.
2661 003566 104401 .WORD RDR ;READ REVERSE.
2662 003570 104005 .WORD WRT ;WRITE
2663 003572 104105 .WORD WTV ;WRITE/VERIFY. (WRITE ALL RECORDS, RDR AND
2664 ;CHECK DATA ON ALL RECORDS, RDF AND
2665 ;CHECK DATA ON ALL RECORDS.)
2666 003574 104010 .WORD SRF ;SPACE "N" RECORDS FORWARD.
2667 003576 104410 .WORD SRR ;SPACE "N" RECORDS REVERSE.
2668 003600 105401 .WORD RNR ;READ NEXT REVERSE. I.E., SPACE FWD, READ REVERS
2669 003602 125401 .WORD RNF ;READ NEXT FORWARD, I.E., READ FORWARD, SPACE RE
2670 003604 105001 .WORD RPF ;READ PREVIOUS FORWARD. I.E., SPACE REVERSE, REA
2671 003606 125001 .WORD RPR ;READ PREVIOUS REVERSE. I.E., READ REVERSE, SPAC
2672 003610 105005 .WORD WRR ;WRITE RETRY.
2673 003612 102010 .WORD RWD ;REWIND.
2674 003614 100012 .WORD MBR ;MESSAGE BUFFER RELEASE
2675 003616 100011 .WORD WTM ;WRITE TAPE MARK
2676 003620 101011 .WORD WTR ;WRITE TAPE MARK RETRY.
2677 003622 105010 .WORD SFF ;SPACE "N" FILES FORWARD.
2678 003624 105410 .WORD SFR ;SPACE "N" FILES REVERSE.
2679 003626 100017 .WORD GES ;GET EXTENDED STATUS.
2680 003630 100411 .WORD ERS ;ERASE 3 INCHES OF TAPE.
2681 003632 100412 .WORD UNL ;REWIND AND UNLOAD.
2682 003634 101012 .WORD CLN ;CLEAR TAPE.
2683 003636 140004 .WORD SCH ;SET CHARACTERISTICS.
2684 003640 100006 .WORD DIA ;DIAGNOSTIC COMMAND.
2685 003642 000040 .WORD JMP ;JUMP TO THE NTH COMMAND IN THE SEQUENCE.
2686 003644 000020 .WORD DLY ;DELAY 'N' MS.
2687 003646 177777 .WORD END ;END OF COMMAND TABLE
2688

```

2689  
2690  
2691 003650 051104 111  
2692 003653 122 043104  
2693 003656 042122 122  
2694 003661 127 052122  
2695 003664 052127 126  
2696  
2697 003667 123 043122  
2698 003672 051123 122  
2699 003675 122 051116  
2700 003700 047122 106  
2701 003703 122 043120  
2702 003706 050122 122  
2703 003711 127 051122  
2704 003714 053522 104  
2705 003717 115 051102  
2706 003722 052127 115  
2707 003725 127 051124  
2708 003730 043123 106  
2709 003733 123 051106  
2710 003736 042507 123  
2711 003741 105 051522  
2712 003744 047125 114  
2713 003747 103 047114  
2714 003752 041523 110  
2715  
2716 003755 104 040511  
2717  
2718  
2719 003760 046512 120  
2720  
2721  
2722 003763 104 054514  
2723  
2724 003766 047105 104  
2725 003772  
2726  
2727  
2728

THE FOLLOWING TABLE CONTAINS THE ASCII FOR EACH COMMAND.

CMDASC::	.ASCII /DRI/	;DRIVE INIT.
	.ASCII /RDF/	;READ FORWARD.
	.ASCII /RDR/	;READ REVERSE.
	.ASCII /WRT/	;WRITE
	.ASCII /WTV/	;WRITE/VERIFY. (WRITE ALL RECORDS, RDR AND CHEC
		;ON ALL RECORDS, RDF AND CHECK DATA ON ALL RECOR
	.ASCII /SRF/	;SPACE "N" RECORDS FORWARD.
	.ASCII /SRR/	;SPACE "N" RECORDS REVERSE.
	.ASCII /RNR/	;READ NEXT REVERSE. I.E., SPACE FWD READ REVERSE
	.ASCII /RNF/	;READ NEXT FORWARD. I.E., READ FORWARD, SPACE RE
	.ASCII /RPF/	;READ PREVIOUS FORWARD. I.E., SPACE REVERSE, READ
	.ASCII /RPR/	;READ PREVIOUS REVERSE. I.E., READ REVERSE, SPACE
	.ASCII /WRR/	;WRITE RETRY.
	.ASCII /RWD/	;REWIND.
	.ASCII /MBR/	;MESSAGE BUFFER RELEASE
	.ASCII /WTM/	;WRITE TAPE MARK
	.ASCII /WTR/	;WRITE TAPE MARK RETRY.
	.ASCII /SFF/	;SPACE "N" FILES FORWARD.
	.ASCII /SFR/	;SPACE "N" FILES REVERSE.
	.ASCII /GES/	;GET EXTENDED STATUS.
	.ASCII /ERS/	;ERASE 3 INCHES OF TAPE.
	.ASCII /UNL/	;REWIND AND UNLOAD.
	.ASCII /CLN/	;CLEAN TAPE.
	.ASCII /SCH/	;SET CHARACTERISTICS. WHERE BRF=200, 40, 20, 0.
		;SEE TS11/TS04 PROGRAMMING SPECIFICATION FOR DES
	.ASCII /DIA/	;DIAGNOSTICS. SEE TS11/TS04 PROGRAMMING SPECIFI
		;FOR DESCRIPTION. ODT MUST BE USED TO LOAD DIAG
	.ASCII /JMP/	;INTO THE WRITE BUFFER BEFORE THIS CMD IS ISSUED
		;JUMP TO THE NTH COMMAND IN THE COMMAND
		;SEQUENCE TABLE, WHERE N IS DEFINED IN
	.ASCII /DLT/	;THE # OF OPERATIONS.
		;DELAY N MS, WHERE N IS DEFINED IN
	.ASCII /END/	;THE # OF OPERATIONS.
	.EVEN	;END OF COMMAND SEQUENCE.

0000  
0001  
0002  
0003  
0004  
0005  
0006  
0007  
0008  
0009  
0010  
0011  
0012  
0013  
0014  
0015  
0016  
0017  
0018  
0019  
0020  
0021  
0022  
0023  
0024  
0025  
0026  
0027  
0028  
0029  
0030  
0031  
0032  
0033  
0034  
0035  
0036  
0037  
0038  
0039  
0040  
0041  
0042

.SBTTL GLOBAL TEXT SECTION

\*\*\*  
; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,  
; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN  
; MORE THAN ONE TEST.  
;

; FORMAT STATEMENTS USED IN PRINT CALLS  
;

.NLIST BEX

003772 047045 040445 047125  
004042 054130 020130 046503  
004102 046503 020104 040520  
004164 040504 040524 041440  
004207 116 020117 051524  
004230 047125 042504 044506  
004254 043122 020103 047516  
004271 124 030523 020061  
004310 042522 051124 020131  
004335 125 044516 020124  
004353 106 047125 052103  
004373 106 052101 046101  
004421 116 020117 047111  
004436 040524 042520 051440  
004460 047524 020117 040515  
004504 040503 051520 040524  
004550 042522 047503 042526  
004572 047125 042522 047503  
004616 047045 040445 051104  
004645 045 022516 040501  
004677 045 022516 041101  
004746 042045 022464 020101  
005010 040445 047516 042040  
005031 045 051101 041505  
005073 045 051101 041505  
005127 045 052501 044516  
005157 045 043501 052105  
005213 045 000116  
005216 047045 051445 000067

CODELM:: .ASCIZ /#N#AUNIT #D1#A TS11 CODE LEVEL P#03#N#N/  
.EVEN  
HALTM:: .ASCIZ /XXX CMD - TYPE <CR> TO CONTINUE/  
CMOPKM:: .ASCIZ /CMD PACKET ADR NOT ON MODULO 4 BOUNDARY: RELOAD!/  
.EVEN  
WTVERM:: .ASCIZ /DATA COMPARE ERROR/  
TOERM:: .ASCIZ /NO TS11 RESPONSE/  
SCERM:: .ASCIZ /UNDEFINED SPEC COND/  
RFCERM:: .ASCIZ /RFC NON ZERO/  
NSSRM:: .ASCIZ /TS11 NOT READY/  
RLEXM:: .ASCIZ /RETRY LIMIT EXCEEDED/  
ATTNM:: .ASCIZ /UNIT OFF LINE/  
FUNRM:: .ASCIZ /FUNCTION REJECT/  
FATSM:: .ASCIZ /FATAL SUBSYSTEM ERROR/  
NOINTM:: .ASCIZ /NO INTERRUPT/  
TSAM:: .ASCIZ /TAPE STATUS ALERT/  
TOOMM:: .ASCIZ /TOO MANY INTERRUPTS/  
RNYM:: .ASCIZ /CAPSTAN RUNAWAY GET STATUS RESULTS:/  
RERM:: .ASCIZ /RECOVERABLE ERROR/  
URERM:: .ASCIZ /UNRECOVERABLE ERROR/  
DROPM:: .ASCIZ /#N#ADROPPED UNIT #D1#N/  
AUDRPM:: .ASCIZ /#N#AALL UNITS DROPPED#N#N/  
DTAER2:: .ASCIZ "#N#ABYTE:#D4#S2#AWAS:#B8#S2#AS/B:#B8#N"  
DTAER3:: .ASCIZ "#D4#A BYTES IN ERROR OUT OF #D4#N"  
DTAER4:: .ASCIZ /#ANO DATA READ#N/  
DTAER5:: .ASCIZ /#ARECORD TOO LONG: >#04#A BYTES#N/  
NURTY1:: .ASCIZ /#ARECOVERED ON RETRY #D2#N/  
OFLINM:: .ASCIZ /#AUNIT #D1#A OFF LINE#N/  
GETSTM:: .ASCIZ /#AGET STATUS CMD RESULTS:#N/  
CRLF:: .ASCIZ /#N/  
CRLFSP:: .ASCIZ /#N#S7/

.LIST BEX  
.EVEN



2744  
2745  
2746  
2747  
2748  
2749  
2750  
2751  
2752  
2753  
2754  
2755  
2756  
2757  
2758  
2759  
2760  
2761  
2762  
2763  
2764  
2765  
2766  
2767  
2768  
2769  
2770  
2771  
2772  
2773  
2774  
2775  
2776  
2777  
2778  
2779  
2780  
2781  
2782  
2783  
2784  
2785  
2786  
2787  
2788  
2789  
2790  
2791  
2792  
2793  
2794  
2795  
2796  
2797  
2798  
2799

005224  
005224  
005224  
005224 016546 003324  
005230 016546 003254  
005234 016546 002532  
005240 012746 005704  
005244 012746 000004  
005250 010600  
005252 104414  
005254 062706 000012  
005260  
005260 012746 005776  
005264 012746 000001  
005270 010600  
005272 104414  
005274 062706 000004  
005300  
005300 010237 006312  
005304  
005304 010337 003364  
005310  
005310 010437 003366  
005314 004737 006346  
005320  
005320 013702 006312  
005324  
005324 010337 006312  
005330  
005330 013703 003364  
005334  
005334 013704 003366  
005340  
005340 013746 006312  
005344 012746 006026  
005350 012746 000002  
005354 010600  
005356 104414  
005360 062706 000006  
005364  
005364 000167  
005366 000000  
  
005370  
005370  
005370 104423

```
.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..
;

DTAERM: BGNMSG DTAERM
        PRINTB @STAER1,DEVTL(R5),PASCNT(R5),RECCNT(R5)

        MOV RECCNT(R
        MOV PASCNT(R
        MOV DEVTL(R
        MOV @STAER1,
        MOV @4,(SP)
        MOV SP,R0
        TRAP C$PNTB
        ADD @12,SP

        PRINTB @STAER7

        MOV @STAER7,
        MOV @1,-(SP)
        MOV SP,R0
        TRAP C$PNTB
        ADD @4,SP

        LET RECD := R2 ;SAVE R2
        MOV R2,RECD
        LET TIME1 := R3 ;SAVE R3
        MOV R3,TIME1
        LET TIME2 := R4 ;SAVE R4
        MOV R4,TIME2

        JSR PC,RECTAP ;RETRIEVE RECORD READ
        LET R2 := RECD ;RESTORE R2
        MOV RECD,R
        LET RECD := R3 ;SAVE RECORD READ
        MOV R3,RECD
        LET R3 := TIME1 ;RESTORE R3
        MOV TIME1,R3
        LET R4 := TIME2 ;RESTORE R4
        MOV TIME2,R4

        PRINTB @STAER6,RECD ;PRINT RECORD READ

        MOV RECD,
        MOV @STAER6,
        MOV @2,(SP)
        MOV SP,R0
        TRAP C$PNTB
        ADD @6,SP

        EXIT MSG

        .EVEN

        ENDMSG

L10002:
        .WORD .JMP
        .WORD L10002

        TRAP C$MNO
```

2800								
2801	005372			BGNMSG	STAERM			
2802	005372			STAERM::				
2803	005372			PRINTB	#STAER1,DEVTBL(R5),PASCNT(R5),RECCNT(R5)			
2804	005372	016546	003324			MOV	RECCNT(R	
2805	005376	016546	003254			MOV	PASCNT(R	
2806	005402	016546	002532			MOV	DEVTBL(R	
2807	005406	012746	005704			MOV	#STAER1,	
2808	005412	012746	000004			MOV	#4,(SP)	
2809	005416	010600				MOV	SP,RO	
2810	005420	104414				TRAP	C\$PNTB	
2811	005422	062706	000012			ADD	#12,SP	
2812	005426			PRINTB	#STAER7			
2813	005426	012746	005776			MOV	#STAER7,	
2814	005432	012746	000001			MOV	#1,(SP)	
2815	005436	010600				MOV	SP,RO	
2816	005440	104414				TRAP	C\$PNTB	
2817	005442	062706	000004			ADD	#4,SP	
2818	005446			LET R2 :=	CMDPKT CLR.BY #177740			
2819	005446	013702	002310			MOV	CMDPKT,R	
2820	005452	042702	177740			BIC	#177740,	
2821	005456			LET R2 :=	R2 #1			
2822	005456	005302				DEC	R2	
2823	005460			IF R2 EQ	#0 THEN			
2824	005460	005702						
2825	005462	001016						
2826	005464	004737	006346					
2827	005470			JSR PC,RECTAP				
2828	005470	010337	006312	LET RECRED :=	R3			
2829	005474			PRINTB	#STAER6,RECRED			
2830	005474	013746	006312					
2831	005500	012746	006026					
2832	005504	012746	000002					
2833	005510	010600						
2834	005512	104414						
2835	005514	062706	000006					
2836	005520			ENDIF				
2837	005520							
2838	005520			PRINTB	#STAER2			
2839	005520	012746	006062					
2840	005524	012746	000001					
2841	005530	010600						
2842	005532	104415						
2843	005534	062706	000004					
2844	005540			PRINTX	#STAER3,CMDPKT,@TSDB(R5),MSGPKT*MS.RFC,TSSREG,CTCC			
2845	005540	013746	003376					
2846	005544	013746	003402					
2847	005550	013746	002340					
2848	005554	017546	002452					
2849	005560	013746	002310					
2850	005564	012746	006141					
2851	005570	012746	000006					
2852	005574	010600						
2853	005576	104415						
2854	005600	062706	000016					
2855	005604			PRINTX	#STAER4,CMDPKT*2,CMDPKT*4,CMDPKT*6			

2856	005604	013746	002316				MOV	CMDPKT+6
2857	005610	013746	002314				MOV	CMDPKT+4
2858	005614	013746	002312				MOV	CMDPKT+2
2859	005620	012746	006177				MOV	STAER4,
2860	005624	012746	000004				MOV	04, (SP)
2861	005630	010600					MOV	SP,RO
2862	005632	104415					TRAP	C\$PNTX
2863	005634	062706	000012				ADD	012,SP
2864	005640				PRINTX	STAER5,MSGPKT+MS.XS0,MSGPKT+MS.XS1,MSGPKT+MS.XS2,MSGPKT+MS.XS3		
2865	005640	013746	002350				MOV	MSGPKT+M
2866	005644	013746	002346				MOV	MSGPKT+M
2867	005650	013746	002344				MOV	MSGPKT+M
2868	005654	013746	002342				MOV	MSGPKT+M
2869	005660	012746	006217				MOV	STAER5,
2870	005664	012746	000005				MOV	05, -(SP)
2871	005670	010600					MOV	SP,RO
2872	005672	104415					TRAP	C\$PNTX
2873	005674	062706	000014				ADD	014,SP
2874	005700				EXIT	MSG		
2875	005700	000167					.WORD	J\$JMP
2876	005702	000410					.WORD	L10003-2
2877								
2878								
	005704	040445	054130	020130	STAER1:	.NLIST BEX		
						.ASCIZ /#AXXX CMD FAILED UNIT #D1#S3#APASS:#D5#S3#ARECORD:#D5#N/		
						.EVEN		
	005776	040445	051120	053105	STAER7:	.ASCIZ /#APREVIOUS CMD WAS XXX/		
	006026	051445	030461	040445	STAER6:	.ASCIZ /#S11#A* RECORD READ:#D5#A */		
	006062	047045	040445	046503	STAER2:	.ASCIZ /#N#ACMDPKT#S2#ATSBA#S4#ARFC#S5#ATSSR#S3#ATCC#N/		
	006141	045	033117	051445	STAER3:	.ASCIZ /#06#S2#06#S2#06#S2#06#S2#D1#N/		
	006177	045	033117	047045	STAER4:	.ASCII /#06#N/		
	006204	047445	022466	116		.ASCII /#06#N/		
	006211	045	033117	047045		.ASCIZ /#06#N/		
	006217	045	054101	052123	STAER5:	.ASCII /#AXST0#S4#AXST1#S4#AXST2#S4#AXST3#N/		
	006262	047445	022466	031123		.ASCIZ /#06#S2#06#S2#06#S2#06#N/		
						.LIST BEX		
						.EVEN		
2879					RECRED:	.WORD 0		;RECORD READ FROM TAPE
2880	006312	000000						
2881								
2882	006314					ENDMSG		
2883	006314				L10003:			
2884	006314	104423					TRAP	C\$MSG

```

2885 .SBTTL GLOBAL SUBROUTINES SECTION
2886
2887
2888 ;**
2889 ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
2890 ; THAT ARE USED IN MORE THAN ONE TEST.
2891 ;
2892 ;
2893 ; MODULES TO HANDLE TS04 INTERRUPTS.
2894
2894 006316 BGNSRV TS4IN0 ;DEVICE 0.
2895 006316 TS4IN0::
2896 006316 LET INTFLG := INTFLG + 01 ;SET INTERRUPT OCCURRED FLAG.
2897 006316 005237 003416 INC INTFLG
2898 006322 ENDSRV
2899 006322 L10004:
2900 006322 000002 RTI
2901
2902 006324 BGNSRV TS4IN1 ;DEVICE 1.
2903 006324 TS4IN1::
2904 006324 LET INTFLG+2 := INTFLG+2 + 01 ;SET INTERRUPT OCCURRED FLAG.
2905 006324 005237 003420 INC INTFLG+2
2906 006330 ENDSRV
2907 006330 L10005:
2908 006330 000002 RTI
2909
2910 006332 BGNSRV TS4IN2 ;DEVICE 2.
2911 006332 TS4IN2::
2912 006332 LET INTFLG+4 := INTFLG+4 + 01 ;SET INTERRUPT OCCURRED FLAG.
2913 006332 005237 003422 INC INTFLG+4
2914 006336 ENDSRV
2915 006336 L10006:
2916 006336 000002 RTI
2917
2918 006340 BGNSRV TS4IN3 ;DEVICE 3.
2919 006340 TS4IN3::
2920 006340 LET INTFLG+6 := INTFLG+6 + 01 ;SET INTERRUPT OCCURRED FLAG.
2921 006340 005237 003424 INC INTFLG+6
2922 006344 ENDSRV
2923 006344 L10007:
2924 006344 000002 RTI

```

H6

```

2925      ; SUBROUTINE TO RETRIEVE RECORD COUNT READ FROM TAPE FOR ERROR
2926      ; PRINTS.
2927      ; INPUTS:
2928      ; OUTPUTS: R3 = RECORD COUNT READ
2929      ; REGISTERS: R2, R3, R4
2930      ; CALLS:
2931
2932 006346      RECTAP::IF #MOD.CO SETIN CMDWRD THEN      ;READ REV FETCH
2933 006346 032737 000400 003346      BIT      #MOD.CO,
2934 006354 001430      BEQ      50001$
2935 006356      LET R2 := MSGPKT.MS.RFC + DATARD ;FIND LAST READ AD.
2936 006356 013702 002340      MOV      MSGPKT.M
2937 006362 063702 003336      ADD      DATARD,R
2938 006366      IF #BIT00 SETIN R2 THEN      ;ODD AD., REASSEMBLE
2939 006366 032702 000001      BIT      #BIT00,P
2940 006372 001417      BEQ      50002$
2941 006374      LET R2 := R2 + #1      ;REC COUNT STARTING
2942 006374 005202      INC      R2
2943 006376      LET R3 := (R2) CLR.BY #177400 ;WITH UPPER BYTE FETCH
2944 006376 111203      MOV      (R2),R3
2945 006400 142703 177400      BIC      #177400,
2946 006404      LET R3 := SWAP R3      ;
2947 006404 000303      SWAB      R3
2948 006406      LET R2 := R2 + #1      ;LOWER BYTE AD.
2949 006406 005302      DEC      R2
2950 006410      IFB SWBFLG NE #0 THEN
2951 006410 105737 003444      TSTB      SWBFLG
2952 006414 001401      BEQ      50003$
2953 006416      LET R2 := R2 - #1      ;LOWER BYTE AD. ON SWAP
2954 006416 005302      DEC      R2
2955 006420      ENDIF
2956 006420      LET R4 := (R2) CLR.BY #177400 ;FETCH LOWER BYTE
2957 006420      MOV      (R2),R4
2958 006420 111204      BIC      #177400,
2959 006422 142704 177400      MOV      (R2),R4
2960 006426      LET R3 := R3 OR R4      ;MERGE BYTES
2961 006426 050403      BIC      R4,R3
2962 006430      ELSE
2963 006430 000401      BR      50004$
2964 006432      LET R3 := (R2)      ;EVEN AD. FETCH
2965 006432 011203      MOV      (R2),R3
2966 006434      ENDIF
2967 006434      EI SE
2968 006434      BR      50004$
2969 006434 000402      BR      50005$
2970 006436      LET R3 := @DATARD      ;READ FWD FETCH
2971 006436 017703 174674      MOV      @DATARD,
2972 006436      ENDIF
2973 006442      BR      50005$
2974 006442
2975 006442
2976
2977 006442 000207      RTS      PC

```

```

2978 ; SUBROUTINE TO STORE A SET CHARACTERISTIC COMMAND AS
2979 ; THE FIRST ENTRY IN THE SEQUENCE TABLE.
2980 ; INPUTS:
2981 ; OUTPUTS:
2982 ; REGISTERS:
2983 ; CALLS:
2984
2985 006444 SETCH:: LET R1 := #CMDSEQ ;INIT COMMAND SEQUENCE TABLE POINTER.
2986 006444 012701 003460 ; MOV #CMDSEQ,
2987 006450 012721 140004 ; THIS CODE SETS UP A SET CHARACTERISTIC
2988 006454 012721 000040 ; COMMAND AS THE FIRST COMMAND IN THE
2989 006460 012721 000001 ; SEQUENCE TABLE.
2990 006464 005721 ; TST (R1). ;SKIP PATTERN LOCATION.
2991 006466 000207 ; RTS PC
2992
2993
2994
2995
2996 ; SUBROUTINE TO STORE A REWIND COMMAND IN THE SEQUENCE TABLE
2997 ; INPUTS:
2998 ; OUTPUTS:
2999 ; REGISTERS:
3000 ; CALLS:
3001
3002 006470 SETRW:: LET (R1). := #RWD ;CMD = REWIND.
3003 006470 012721 102010 ; MOV #RWD,(R1
3004 006474 ; LET (R1). := #1 ;BRF.
3005 006474 012721 000001 ; MOV #1,(R1).
3006 006500 ; LET (R1). := #1 ;# OF OPERATIONS.
3007 006500 012721 000001 ; MOV #1,(R1).
3008 006504 005721 ; TST (R1). ;SKIP PATTERN.
3009 006506 000207 ; RTS PC ;RETURN

```

```

3010 ; SUBROUTINE TO EXECUTE ALL COMMANDS IN THE SEQUENCE TABLE ON ALL
3011 ; DEVICES.
3012 ; INPUTS:
3013 ; OUTPUTS: R2 = TERMINATION INDICATOR (0-END OF TABLE,1-EOT)
3014 ; REGISTERS:
3015 ; CALLS: CMDAC, SETUP, EXSUB, CKMAE, NEXTU, FIRSTU, VFYDAT.
3016
3017 006510 EXALL:: LET R1 := #CMDSEQ ;INIT SEQUENCE TABLE POINTER.
3018 006510 012701 003460 ; MOV #CMDSEQ,
3019 006514 ; WHILE THERE ARE CMDS IN THE SEQUENCE TA
3020 006514 50006$:
3021 006514 021127 177777 ; CMP (R1),#EN
3022 006520 001527 ; BEQ 50007$
3023 006522 004737 007452 JSR PC,SETUP ;GO SETUP THE COMMAND BLOCK.
3024 006526 ; WHILE THERE ARE RECORDS REMAINING:
3025 006526 50010$:
3026 006526 023737 003340 003342 ; CMP NCNT,NCN
3027 006534 002116 ; BGE 50011$
3028 006536 004737 007344 JSR PC,CMDAC ;STORE CMD ASCII IN ERROR MESSAGE.
3029 006542 IFB RANDOM NE #0 THEN ;IF IN RANDOM MODE:
3030 006542 105737 003441 ; TSTB RANDCM
3031 006546 001435 ; BEQ 50012$
3032 006550 IF CMDWRD EQ #WRT THEN ;IF CMD IS A WRITE THEN:
3033 006550 023727 003346 104005 ; CMP CMDWRD,#
3034 006556 001031 ; BNE 50013$
3035 006560 IFB VFYFLG EQ #0 THEN ;IF DATA IS NOT TO BE VERIFIED THEN:
3036 006560 105737 003442 ; TSTB VFYFLG
3037 006564 001026 ; BNE 50014$
3038 006566 LET RANB := RANB + RANS ;GENERATE
3039 006566 063737 003362 003360 ; ADD RANS,RAN
3040 006574 LET RANS := RANS + RANB ;RANDOM
3041 006574 063737 003360 003362 ; ADD RANB,RAN
3042 006602 LET BRFCNT := RANS ;LENGTH
3043 006602 013737 003362 003344 ; MOV RANS,BRF
3044 006610 LET BRFCNT := BRFCNT CLR.BY LENMSK ;MASK RANDOM LENGTH.
3045 006610 043737 003356 003344 ; BIC LENMSK,B
3046 006616 IF BRFCNT LT #18. THEN ;DO NOT ALLOW BYTE COUNT OF LESS
3047 006616 023727 003344 000022 ; CMP BRFCNT,#
3048 006624 002003 ; BGE 50015$
3049 006626 LET BRFCNT := #18. ;CHANGE COUNT OF 0 17 TO 18.
3050 006626 012737 000022 003344 ; MOV #18.,BRF
3051 006634 ;
3052 006634 ;
3053 006634 LET CMDPKT.CP.CNT := BRFCNT ;MOVE BRF TO CMD PACKET.
3054 006634 013737 003344 002316 ; MOV BRFCNT,C
3055 006642 ;
3056 006642 ;
3057 006642 ;
3058 006642 ;
3059 006642 ;
3060 006642 ;
3061 006642 004737 007004 JSR PC,EXSUB ;ISSUE CMD TO ALL,AWAIT INTS,CHECK STATL
3062 006646 004737 016060 JSR PC,CKMAE ;CHECK HALT AFTER EACH CMD FLAG.
3063 006652 LET R2 := #1 ;SET ALL UNITS AT BOT/EOT.
3064 006652 012702 000001 ; MOV #1,R2
3065 006656 004737 015452 JSR PC,FIRSTU ;FIND FIRST UNIT.

```

```

3066 006662          WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE UNITS:
3067 006662          50016$:
3068 006662 026527 002532 177777          CMP      DEVTBL(R
3069 006670 001426          BEQ      50017$
3070 006672          IF #MOD.CO SETIN CMDWRD THEN ;IF CMD IS REVERSE THEN:
3071 006672 032737 000400 003346          BIT      #MOD.CO,
3072 006700 001406          BEQ      50020$
3073 006702          IF #XO.BOT NOTSETIN EOTFLG(R5) THEN ;IF NOT AT BOT THEN:
3074 006702 032765 000002 003426          BIT      #XO.BOT,
3075 006710 001001          BNE      50021$
3076 006712          LET R2 :- #0          ;CLEAR EOT/BOT FLAG.
3077 006712 005002          CLR      R2
3078 006714          ENDIF
3079 006714          50021$:
3080 006714          ELSE          ;ELSE IF CMD IS NOT REVERSE:
3081 006714 000411          BR      50022$
3082 006716          50020$:
3083 006716          IF #XO.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN
3084 006716 032765 000001 003426          BIT      #XO.EOT,
3085 006724 001404          BEQ      50023$
3086 006726 032737 000001 003346          BIT      #CMD.CO,
3087 006734 001001          BNE      50024$
3088 006736          50023$:
3089          ;IF NOT AT EOT OR NOT A MOTION CMD THEN:
3090 006736          LET R2 :- #0          ;CLEAR EOT/BOT FLAG.
3091 006736 005002          CLR      R2
3092 006740          ENDIF
3093 006740          50024$:
3094 006740          ENDIF
3095 006740          50022$:
3096 006740 004737 015520          JSR PC,NEXTU          ;FIND NEXT UNIT
3097 006744          ENDDO          ;
3098 006744 000746          BR      50016$
3099 006746          50017$:
3100 006746          IF R2 EQ #1 THEN          ;IF ALL UNIT ARE AT EOT/BOT THEN:
3101 006746 020227 000001          CMP      R2,#1
3102 006752 001001          BNE      50025$
3103 006754 000412          BR      EXARTN          ;RETURN WITH R2 = #1.
3104 006756          ENDIF
3105 006756          50025$:
3106 006756          LET NCNT :- NCNT + #1          ;UPDATE RECORD COUNT.
3107 006756 005237 003340          INC      NCNT
3108 006762          LET PCMDWD := CMDWRD          ;SAVE PREVIOUS COMMAND WORD.
3109 006762 013737 003346 003352          MOV      CMDWRD,P
3110 006770          ENDDO
3111 006770 000656          BR      50010$
3112 006772          50011$:
3113 006772 004737 014402          JSR PC,VFYDAT          ;IF LAST CMD WAS A WRITE VERIFY, THEN GO
3114          ;VERIFY THE LAST N RECORDS OF DATA.
3115 006776          ENDDO
3116 006776 000646          BR      50006$
3117 007000          50007$:
3118 007000          LET R2 := #0          ;SET NORMAL RETURN INDICATOR.
3119 007000 005002          CLR      R2
3120 007002 000207          EXARTN: RTS PC          ;RETURN.
3121

```



( r )

```

3122
3123
3124
3125
3126
3127
3128
3129
3130
3131 007004 004737 015452
3132 007010
3133 007010
3134 007010 026527 002532 177777
3135 007016 001465
3136 007020
3137 007020 032737 000400 003346
3138 007026 001421
3139 007030
3140 007030 032765 000002 003426
3141 007036 001014
3142 007040
3143 007040 032765 000001 003426
3144 007046 001406
3145 007050
3146 007050 105737 003450
3147 007054 001402
3148 007056 004737 010326
3149 007062
3150 007062
3151 007062
3152 007062 000402
3153 007064
3154 007064 004737 010326
3155 007070
3156 007070
3157 007070
3158 007070
3159 007070
3160 007070 000435
3161 007072
3162 007072
3163 007072 023727 003354 000002
3164 007100 001011
3165 007102 032765 000002 003426
3166 007110 001405
3167
3168 007112
3169 007112 016537 002544 003436
3170 007120
3171 007120 005077 174312
3172 007124
3173 007124
3174 007124
3175 007124 032765 000001 003426
3176 007132 001404
3177 007134 032737 000001 003346

```

```

; SUBROUTINE TO ISSUE COMMAND TO ALL DEVICES, WAIT FOR
; ALL INTERRUPTS, AND CHECK ALL STATUS.
; INPUTS:
; OUTPUTS:
; REGISTERS:
; CALLS: EXCUTE,GOWAIT,NEXTU,FIRSTU.

EXSUB:: JSR PC,FIRSTU ;SET UP FOR FIRST UNIT.
        WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
            50026$:
                CMP DEVTBL(R
                BEQ 50027$
                IF #MOD.CO SETIN CMDWRD THEN ;IF CMD IS REVERSE THEN:
                    BIT #MOD.CO,
                    BEQ 50030$
                IF #XO.BOT NOTSETIN EOTFLG(R5) THEN ;IF NOT AT BOT
                    BIT #XO.BOT,
                    BNE 50031$
                IF #XO.EOT SETIN EOTFLG(R5) THEN ;BUT IF AT EOT
                    BIT #XO.EOT,
                    BEQ 50032$
                IFB ALLEOT NE #0 THEN ;AND ALL OTHER AT EOT
                    TSTB ALLEOT
                    BEQ 50033$
                JSR PC,EXCUTE ;THEN EXECUTE FEV CMD
                ENDIF ;IF NOT ALL AT EOT, FREEZE UNIT
                50033$:
                ELSE ;IF NOT AT BOT AND
                    BR 50034$
                50032$:
                ;NOT AT EOT, EXEC REV CM
                JSR PC,EXCUTE
                ENDIF
                50034$:
                ENDIF
                50031$:
                ELSE ;ELSE IF CMD IS NOT REVERSE:
                    BR 50035$
                50030$:
                IF CMDLG EQ #2 AND #XO.BOT SETIN EOTFLG(R5) THEN
                    CMP CMDLG,#2
                    BNE 50036$
                    BIT #XO.BOT,
                    BEQ 50036$
                    ;CLEAR BAD SPOT COUNTS WHEN WRITING FROM
                    LET BTPT := BTADDR(R5)
                    MOV BTADDR(R
                    LET #BTPT := #0
                    CLR #BTPT
                ENDIF
                50036$:
                IF #XO.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN
                    BIT #XO.EOT,
                    BEQ 50037$
                    BIT #CMD.CO,

```

```

3178 007142 001003
3179 007144
3180
3181 007144 004737 010326      JSR PC,EXCUTE      ;IF NOT AT EOT OR NOT A MOTION CMD THEN:
3182 007150                      ELSE                  ;ISSUE CMD TO TS04
3183 007150 000405
3184 007152
3185 007152                      IFB ALLEOT NE #0 THEN
3186 007152 105737 003450
3187 007156 001402
3188 007160 004737 010326      JSR PC,EXCUTE      ;
3189 007164                      ENDIF
3190 007164
3191 007164                      ENDIF
3192 007164
3193 007164                      ENDIF
3194 007164
3195 007164 004737 015520      JSR      PC,NEXTU      ;FIND NEXT UNIT IN TEST CYCLE.
3196 007170                      ENDDO
3197 007170 000707
3198 007172
3199 007172                      IFB RPTFLG NE #0 THEN      ;IF REPORT HAS BEEN REQUESTED THEN:
3200 007172 105737 003443
3201 007176 001403
3202 007200                      LET RPTFLG :B #0      ;CLR THE FLAG.
3203 007200 105037 003443
3204 007204                      DORPT                      ;PRINT THE PERFORMANCE REPORT.
3205 007204 104424
3206 007206                      ENDIF
3207 007206
3208 007206 004737 015452      JSR PC,FIRSTU      ;SET UP FOR FIRST UNIT.
3209 007212                      WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
3210 007212
3211 007212 026527 002532 177777
3212 007220 001450
3213 007222
3214 007222 032737 000400 003346      IF #MOD.CO SETIN CMDWRD THEN ;IF CMD IS REVERSE THEN:
3215 007230 001421
3216 007232
3217 007232 032765 000002 003426      IF #X0.BOT NOTSETIN EOTFLG(R5) THEN ;IF NOT AT BOT
3218 007240 001014
3219 007242
3220 007242 032765 000001 003426      IF #X0.EOT SETIN EOTFLG(R5) THEN ;BUT IF AT EOT
3221 007250 001406
3222 007252
3223 007252 105737 003450
3224 007256 001402
3225 007260 004737 010636      JSR PC,GOWAIT      ;THEN WAIT FOR CMD END
3226 007264                      ENDIF                      ;IF NOT ALL AT EOT, DO N
3227 007264
3228 007264                      ELSE
3229 007264 000402
3230 007266
3231 007266 004737 010636      JSR PC,GOWAIT
3232 007272                      ENDIF
3233 007272

```

```

3234 007272          ENDIF
3235 007272
3236 007272          50047$:
3237 007272 000420    ;ELSE IF CMD IS FORWARD:
3238 007274          BR          50053$
3239 007274          50046$:
3240 007274 032765 000001 003426    IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN
3241 007302 001404          BIT      #X0.EOT,
3242 007304 032737 000001 003346    BEQ      50054$
3243 007312 001003          BIT      #CMD.CO,
3244 007314          BNE      50055$
3245          50054$:
3246 007314 004737 010636    ;IF NOT AT EOT OR NOT A MOTION CMD THEN:
3247 007320          JSR PC,GOWAIT    ;WAIT FOR INT,CHECK STATUS.
3248 007320 000405    ELSE
3249 007322          BR          50056$
3250 007322          50055$:
3251 007322 105737 003450    IFB ALLEOT NE #0 THEN
3252 007326 001402          TSTB    ALLEOT
3253 007330 004737 010636    BEQ      50057$
3254 007334          JSR PC,GOWAIT
3255 007334          ENDIF
3256 007334          ENDIF
3257 007334          50057$:
3258 007334          50056$:
3259 007334          50053$:
3260 007334 004737 015520    JSR      PC,NEXTU    ;FIND NEXT UNIT IN TEST CYCLE.
3261 007340          ENDDO
3262 007340 000724          BR          50044$
3263 007342          50045$:
3264 007342 000207    RTS PC          ;RETURN.

```

137

```

3265 ; THIS SUBROUTINE STORES THE ASCII FOR THE CURRENT COMMAND AND PREVIOUS
3266 ; COMMAND IN THE STANDARD ERROR MESSAGE. ON ENTRY LOCATION CMDWRD
3267 ; CONTAINS CURRENT CMD AND LOCATION PCMDWD CONTAINS PREVIOUS CMD.
3268 ; INPUTS:
3269 ; OUTPUTS:
3270 ; REGISTERS: R3, R4.
3271 ; CALLS: GCMDA
3272
3273 CMDAC:: LET R4 := CMDWRD ;R4 = CMD BINARY.
3274 007344 013704 003346 ;R4 = CMD BINARY. MOV CMDWRD,R
3275 007350 004737 007416 JSR PC,GCMDA ;GET CMD ASCII.
3276 007354 112337 005706 MOVB (R3),STAER1.2 ;MOVE CMD ASCII
3277 007360 112337 005707 MOVB (R3),STAER1.3 ;
3278 007364 111337 005710 MOVB (R3),STAER1.4 ;INTO MSG.
3279 007370 LET R4 := PCMDWD ;R4 = PREVIOUS CMD BINARY.
3280 007370 013704 003352 ;R4 = PREVIOUS CMD BINARY. MOV PCMDWD,R
3281 007374 004737 007416 JSR PC,GCMDA ;GET CMD ASCII.
3282 007400 LET STAER7.24 :B= (R3). ;MOVE CMD ASCII
3283 007400 112337 006022 ; MOVB (R3),ST
3284 007404 LET STAER7.25 :B= (R3). ;
3285 007404 112337 006023 ; MOVB (R3),ST
3286 007410 LET STAER7.26 :B= (R3) ;INTO MSG.
3287 007410 111337 006024 ; MOVB (R3),STA
3288 007414 000207 RTS PC ;RETURN. GO EXECUTE NEXT FUNCTION.
3289
3290
3291
3292 ; SUBROUTINE TO FIND THE ASCII EQUIVLENT OF THE COMMAND IN R4.
3293 ; ADDRESS OF ASCII 1ST WORD IS RETURNED IN R3.
3294 ; INPUTS: R4 = PRESENT COMMAND WORD.
3295 ; OUTPUTS: R3 = ADDRESS OF PRESENT COMMAND ASCII.
3296 ; REGISTERS:
3297 ; CALLS:
3298
3299 GCMDA:: LET R3 := #0 ;INIT CMD TBL POINTER.
3300 007416 005003 ;INIT CMD TBL POINTER. CLR R3
3301 007420 WHILE CMDTBL(R3) NE R4 DO ;UNTIL CURRENT CMD IS FOUND:
3302 007420 ; UNTIL CURRENT CMD IS FOUND: 50060$:
3303 007420 026304 003562 ; UNTIL CURRENT CMD IS FOUND: 50060$:
3304 007424 001403 ; UNTIL CURRENT CMD IS FOUND: 50060$:
3305 007426 LET R3 := R3 + #2 ;SEARCH CMD TABLE.
3306 007426 062703 000002 ;SEARCH CMD TABLE.
3307 007432 ENDDO ;SEARCH CMD TABLE.
3308 007432 000772 ;SEARCH CMD TABLE.
3309 007434 ;SEARCH CMD TABLE. 50061$:
3310 007434 LET R4 := R3 ;SEARCH CMD TABLE. 50061$:
3311 007434 010304 ;SEARCH CMD TABLE. 50061$:
3312 007436 LET R3 := R3 SHIFT 1 ;POINT TO ASCII FOR THAT COMMAND
3313 007436 006203 ;POINT TO ASCII FOR THAT COMMAND
3314 007440 000240 ;POINT TO ASCII FOR THAT COMMAND
3315 007442 060403 ;POINT TO ASCII FOR THAT COMMAND
3316 007444 062703 003650 ;POINT TO ASCII FOR THAT COMMAND
3317 007450 000207 RTS PC ;RETURN.

```

C7

```

3318      | THIS SUBROUTINE LOADS THE TSO4 COMMAND PACKET FROM ONE
3319      | ENTRY IN THE SEQUENCE TABLE.
3320      | INPUTS:
3321      | OUTPUTS:
3322      | REGISTERS:      R2, R3.
3323      | CALLS:          GENPAT.
3324
3325      SETUP:: LET CMDLG := #0          ;CLR CMD LOGGING CODE(DISABLE LOGGING)
3326      007452 005037 003354          ;CLR CMDLG
3327      007456 012137 002310          MOV (R1),CMDPKT          ;LOAD THE COMMAND WORD.
3328      007462 011137 002316          MOV (R1),CMDPKT.CP.CNT ;LOAD THE BYTE/RECORD/FILE COUNT.
3329      007466 011137 003344          MOV (R1),BRFCNT        ;SAVE BRFCNT FOR THIS COMMAND.
3330      007472 013702 002310          MOV CMDPKT,R2          ;GET CMD.
3331      007476 042702 177740          BIC @NCHD.C,R2        ;CLR ALL BUT CMD BITS.
3332      007502 010203                  MOV R2,R3              ;SAVE IT TWICE.
3333      007504 162703 000010          SUB @CMD.C3,R3         ;POSITION COMMAND?
3334      007510 001003                  BNE 2$                 ;BR IF NOT.
3335      007512 011137 002312          MOV (R1),CMDPKT*2     ;MOVE BPCR IN 2ND PKT WORD FOR POSITION
3336      007516 000461                  BR 3$
3337      007520                          2$: IF CMDPKT EQ @WTM THEN ;IF CMD IS A WRITE TAPE MARK THEN:
3338      007520 023727 002310 100011          CMP CMDPKT,@          ;
3339      007526 001003                  BNE 50062$            ;
3340      007530                          ;WTM LOGGING CODE IS 2.
3341      007530 012737 000002 003354          MOV @2,CMDLG
3342      007536                          ;
3343      007536                          50062$:
3344      007536 010203                  MOV R2,R3              ;IS IT A READ?
3345      007540 162 03 000001          SUB @CMD.CO,R3         ;BR IF NOT.
3346      007544 001017                  BNE 1$                 ;IF SO, LOAD THE BUFFER ADDR.
3347      007546 013737 003336 002312          MOV DATARD,CMDPKT.CP.ADL ;IF CMD IS A READ REV THEN:
3348      007554                          IF @MOD.CO SET IN CMDPKT THEN ;
3349      007554 032737 000400 002310          BIT @MOD.CO,          ;
3350      007562 001404                  BEQ 50063$            ;
3351      007564                          ;LOGGING CODE IS 4.
3352      007564 012737 000004 003354          MOV @4,CMDLG
3353      007572                          ;ELSE - IF CMD IS A READ FWD:
3354      007572 000403                  BR 50064$            ;
3355      007574                          50063$:
3356      007574                          ;LOGGING CODE IS 6.
3357      007574 012737 000006 003354          MOV @6,CMDLG
3358      007602                          ;
3359      007602                          50064$:
3360      007602 000427                  BR 3$                 ;CONTINUE.
3361      007604 010203                  1$: MOV R2,R3          ;IS IT
3362      007606 162703 000004          SUB @CMD.C2,R3         ;A SET CHARACTERISTICS CMD?
3363      007612 001011                  BNE 4$                 ;BR IF NOT.
3364      007614                          LET CMDPKT.CP.ADL := @SCHBK ;SET UP ADR LO FOR SET CHAR.
3365      007614 012737 002442 002312          MOV @SCHBK,C          ;
3366      007622 012737 000010 002316          MOV @SCHCNI,CMDPKT.CP.CNT ;SET BUFFER EXTENT
3367      007630                          LET SCHBK*6 := (R1) ;STORE CHARACTERISTIC CODE IN SCH BLOCK.
3368      007630 011137 002450          MOV (R1),SCH          ;
3369      007634 000412                  BR 3$                 ;CONTINUE.
3370      007636 010203                  4$: MOV R2,R3          ;IS IT
3371      007640 162703 000006          SUB @CMD.C1!CMD.C2,R3 ;A DIAGNOSTIC (DIA) CMD?
3372      007614 001006                  BNE 3$                 ;BR IF NOT.
3373      007646 012737 000020 002316          MOV @DIACNT,CMDPKT.CP.CNT ;LOAD BUFFER EXTENT.

```

```

3374 007654 012737 003334 002312      MOV      #DIABLK,CMDPKT+CP,ADL ;LOAD BUFFER ADR 1 OW.
3375 007662 005721                    3$:   TST      (R1). ;POINT TO N (NUMBER OF TIMES TO EXECUTE
3376 007664                                LET NCNT1 := (R1). ;SAVE NUMBER OF OPERATIONS
3377 007664 012137 003342                                MOV      (R1)+,NC
3378 007670                                ;CLEAR OPERATION COUNTER.
3379 007670 005037 003340                                CLR      NCNT
3380 007674 012137 003374      MOV      (R1)+,PATERN ;SAVE PATTERN CODE FOR CURRENT CMD.
3381 007700 010203      MOV      R2,R3 ;IS IT
3382 007702 162703 000005      SUB      #CMD.C0!CMD.C2,R3 ;A WRITE?
3383 007706 001010      BNE      5$ ;BR IF NOT.
3384 007710 013737 003334 002312      MOV      DATAWT,CMDPKT+CP,ADL ;LOAD WRITE BUFFER LO ORDER.
3385 007716 004737 010030      JSR      PC,GENPAT ;GO GENERATE THE WRITE PATTERN.
3386 007722                                LET CMDLG := #2 ;WRITE LOGGING CODE IS 2.
3387 007722 012737 000002 003354                                MOV      #2,CMDLG
3388 007730                    5$:   IF #VFY.C SET IN CMDPKT THEN ;IF DATA VERIFICATION IS REQUIRED:
3389 007730 032737 000100 002310                                BIT      #VFY.C.C
3390 007736 001407                                BEQ      50065$
3391 007740                                LET VFYFLG :B= #1 ;SET VERIFY FLAG.
3392 007740 112737 000001 003442                                MOV      #1,VFYFL
3393 007746 042737 000100 002310      BIC      #VFY.C,CMDPKT ;CLEAR VERIFY BIT(NOT USED BY HARDWARE).
3394 007754                                ELSE ;IF DATA VERIFICATION IS NOT REQUIRED:
3395 007754 000402                                BR      50066$
3396 007756                                50065$:
3397 007756                                LET VFYFLG :B= #0 ;CLR VERIFY FLAG.
3398 007756 105037 003442                                CLRB    VFYFLG
3399 007762                                ENDIF
3400 007762                                50066$:
3401 007762                                LET PCMDWD := CMDWRD ;SAVE PREVIOUS CMD WORD.
3402 007762 013737 003346 003352                                MOV      CMDWRD,P
3403 007770                                LET CMDWRD := CMDPKT ;SAVE PRESENT CMD WORD.
3404 007770 013737 002310 003346                                MOV      CMDPKT,C
3405 007776                                IFB SWBFLG NE #0 THEN ;IF SWAP BYTES IS ENABLED:
3406 007776 105737 003444                                TSTB    SWBFLG
3407 010002 001403                                BEQ      50067$
3408 010004                                LET CMDPKT := CMDPKT SET BY #SWB.C ;SET SWAP BIT IN COMMAND.
3409 010004 052737 010000 002310                                BIS      #SWB.C,C
3410 010012                                ENDIF
3411 010012                                50067$:
3412 010012 042737 004000 002310      BIC      #BRF.C,CMDPKT ;CLR BRF BIT (INTERNAL ONLY).
3413 010020                                LET CMDSAV := CMDPKT ;SAVE 1ST WORD OF COMMAND PACKET.
3414 010020 013737 002310 003350                                MOV      CMDPKT,C
3415 010026 000207                                RTS     PC ;RETURN.

```

```

3416 ; THIS SUBROUTINE SETS UP AND CALLS THE APPROPRIATE SUBROUTINE TO GENERAT
3417 ; THE DESIRED PATTERN FOR THE WRITE AND WRITE/VERIFY COMMANDS.
3418 ; INPUTS:
3419 ; OUTPUTS:
3420 ; REGISTERS: R2, R3, R4.
3421 ; CALLS: PATRO - PATR7
3422
3423 GENPAT:: LET R3 := PATERN SHIFT 1 ;SETUP PATTERN ROUTINE POINTER
3424 010030 013703 003374 MOV PATERN,R
3425 010034 006303 ASL R3
3426 010036 J13704 003344 LET R4 := BRFCNT * #1 ;SET LENGTH OF WRITE BFR
3427 010042 005204 MOV BRFCNT,R
3428 010044 042704 000001 INC R4
3429 010044 042704 000001 LET R4 := R4 CLR.BY #1 ;ROUNDED UP TO NEXT WORD
3430 010050 162704 000002 BIC #1,R4
3431 010054 013702 003334 LET R4 := R4 #2 ;WITH FIRST WORD RESERVED
3432 010060 062702 000002 SUB #2,R4
3433 010064 004773 010072 LET R2 := DATAWT * #2 ;FOR RECORD COUNT
3434 010070 000207 JSR PC,@PATIBL(R3) ;GO GENERATE THE APPROPRIATE PATTERN.
3435 010072 000207 RTS PC ;RETURN TO SETUP SUBROUTINE.
3436
3437 ;TS04 WRITE PATTERN LOOKUP TABLE. USED TO JSR TO THE
3438 ;CORRECT DATA PATTERN GENERATING ROUTINE.
3439
3440
3441
3442 PATIBL: PATRO
3443 PATR1
3444 PATR2
3445 PATR3
3446 PATR4
3447 PATR5
3448 PATR6
3449 PATR7
3450 PATR8
3451
3452
3453 ;INCREMENTING PATTERN. 0 - 377.
3454
3455 PATRO:: LET R3 := #400
3456 010114 012703 000400 MOV #400,R3
3457 010120 162704 000002 1$: LET R4 := R4 #2 ;DECREMENT WORD COUNT.
3458 010124 100411 SUB #2,R4
3459 010126 010322 BMI 2$ ;BR IF DONE.
3460 010130 062703 001002 LET (R2) := R3 ;STORE DATA WORD.
3461 010134 020327 001000 MOV R3,(R2)
3462 010140 001002 LET R3 := R3 * #1002 ;UPDATE PATTERN.
3463 010142 012703 000400 IF R3 EQ #1000 THEN ;IF PATTERN HAS WRAPPED AROUND THEN:
3464 010144 000764 CMP R3,#1000
3465 010146 000764 BNE 50070$
3466 010148 000764 LET R3 := #400 ;INIT THE PATTERN AGAIN.
3467 010150 000764 MOV #400,R3
3468 010152 000764 ENDIF
3469 010154 000764 BR 1$ ;DO IT AGAIN.
3470 50070$:
3471 010156 000764
    
```

```

3472 010150 000207          2$:   RTS    PC           ;RETURN.
3473
3474                          ;ALL ONE'S PATTERN.
3475
3476 010152 012703 177777  PATR1:: MOV    # 1,R3       ;ALL ONES PATTERN;.
3477 010156          ZROPAT: LET R4 := R4 - #2   ;DECREMENT BYTE COUNT.
3478 010156 162704 000002          BMI    1$           ;DONE?,BR IF YES.
3479 010162 100402          MOV    R3,(R2),     ;IF NOT LOAD NEXT BYTE WITH PATTERN.
3480 010164 010322          BR     ZROPAT       ;DO IT AGAIN.
3481 010166 000773
3482
3483 010170 000207          1$:   RTS    PC           ;RETURN.

```





```

3539 ; THIS SUBROUTINE INITIATES TSO4 COMMAND EXECUTION
3540 ; AND CHECKS FOR TSO4 RESPONSE.
3541 ; INPUTS:
3542 ; OUTPUTS:
3543 ; REGISTERS: R2, R3.
3544 ; CALLS: DROPU, MOVMSG, FIRSTU, NEXTU, WSSR.
3545
3546 010326 EXCUTE:: LET TIME1 := # 1 ;INIT TIMEOUT COUNTER.
3547 010326 012737 177777 003364 ;WAIT MOV # 1,TIME
3548 010334 REPEAT ;UPDATE TIMEOUT COUNTER. 50071$:
3549 010334 LET TIME1 := TIME1 #1 ;IF TIMED OUT: DEC TIME1
3550 010334 005337 003364 IF TIME1 EQ #0 THEN ;MOVE CURRENT PACKET MSG. TST TIME1
3551 010340 005737 003364 JSR PC,MOVMSG ;REPORT TSO4 NOT READY BNE 50072$
3552 010340 001011 ERRDF #2,NSSRM,STAERM ;TRAP C$ERDF
3553 010344 004737 011224 ;.WORD 2
3554 010352 104455 ;.WORD NSSRM
3555 010354 000002 ;.WORD STAERM
3556 010356 004271 JSR PC,DROPU ;DROP THE UNIT.
3557 010360 005372 BR EXCRTN ;RETURN.
3558 010362 004737 015554 ENDIF
3559 010370 UNTIL #TS.SSR SETIN @TSSR(R5) ;WAIT UNTIL DEVICE IS READY. 50072$:
3560 010370 032775 000200 002462 BIT #TS.SSR.
3561 010376 001756 IF CMDWRD EQ #SCH THEN ;IF WE ARE DOING A SET CHAR CMD THEN: 50071$
3562 010400 023727 003346 140004 CMP CMDWRD,#
3563 010406 001022 LET R5SAVE := R5 ;SAVE CURRENT DEVICE POINTER. 50073$
3564 010410 010537 003400 JSR PC,FIRSTU ;FIND FIRST UNIT. MOV R5,R5SAVE
3565 010414 004737 015452 WHILE DEVTBL(R5) NE #END DO
3566 010420 026527 002532 177777 ;WAIT FOR UNIT READY OR TIME OUT. 50074$:
3567 010426 001405 JSR PC,WSSR ;FIND NEXT UNIT. CMP DEVTBL(R
3568 010430 004737 011170 JSR PC,NEXTU ;RESTORE CURRENT DEVICE POINTER. 50075$
3569 010434 004737 015520 ENDDO BR 50074$
3570 010440 000767 LET R5 := R5SAVE ;SET UP ADR OF MSG PKT IN SCH BLOCK.
3571 010442 013705 003400 LET SCHBK := MSGPKA(R5) ;ADR OF THIS UNIT'S MSG PACKET. 50073$:
3572 010446 016537 002502 002442 ENDIF ;CLR COUNTER. MOV MSGPKA,R
3573 010454 016503 002502 LET R3 := MSGPKA(R5) ;WHILE THERE ARE MORE LOCATIONS:
3574 010460 005002 LET R2 := #0 ;CLR R2
3575 010462 016503 002502 WHILE R2 NE #MSGCNT DO ;WHILE THERE ARE MORE LOCATIONS:
3576 010462 016503 002502
3577 010462 016503 002502
3578 010462 016503 002502
3579 010462 016503 002502
3580 010462 016503 002502
3581 010462 016503 002502
3582 010462 016503 002502
3583 010462 016503 002502
3584 010462 016503 002502
3585 010462 016503 002502
3586 010462 016503 002502
3587 010462 016503 002502
3588 010462 016503 002502
3589 010462 016503 002502
3590 010462 016503 002502
3591 010462 016503 002502
3592 010462 016503 002502
3593 010462 016503 002502
3594 010462 016503 002502
    
```

```

3595 010462 020227 000016          CMP      R2,#MSGC
3596 010466 001405          BEQ      50077$
3597 010470          LET (R3) := # 1          ;INIT THE MSG PACKET WITH ALL 1'S
3598 010470 012723 177777          MOV      #-1,(R3)
3599 010474          LET R2 := R2 * #2          ;UPDATE COUNTER.
3600 010474 062702 000002          ADD      #2,R2
3601 010500          ENDDO
3602 010500 000770          BR      50076$
3603 010502          50077$:
3604 010502 105737 002210          TSTB    DINT          ;ARE INTERRUPTS DISABLED.
3605 010506 001023          BNE     1$          ;BR IF YES.
3606 010510          IFB INTFLG(R5) GT #1 THEN ;IF MORE THAN ONE INTERRUPT HAS OCCURED:
3607 010510 126527 003416 000001          CMPB    INTFLG(R
3608 010516 003412          BLE     50100$
3609 010520          LET TSSREG := @TSSR(R5)          ;FREEZE THE CURRENT STATUS REG F
3610 010520 017537 002462 003402          MCV     @TSSR(R5)
3611 010526          ERROF #15,TOOMM,STAERM          ;REPORT TOO MANY INTERRUPTS.
3612 010526 104455          TRAP   C$ERDF
3613 010530 000017          .WORD 15
3614 010532 004460          .WORD TOOMM
3615 010534 005372          .WORD STAERM
3616 010536 004737 015554          JSR PC,DROPU          ;DROP THE UNIT
3617 010542 000434          BR EXCRTN          ;RETURN UNIT HAS BEEN DROPPED.
3618 010544          ENDIF
3619 010544
3620 010544          LET INTFLG(R5) := #0          ;CLR INTERRUPT FLAG FOR THIS DEV.
3621 010544 005065 003416          CLR     INTFLG(R
3622 010550 052737 000200 002310          BIS     #IE.C,CMDPKT          ;SET INT ENABLE BIT.
3623 010556          IFB ERRREC EQ #0 THEN ;IF NOT RETRYING
3624 010556 105737 003415          1$:
3625 010562 001005          TSTB    ERRREC
3626 010564          BNE     50101$
3627 010564 005265 003324          INC     RECcnt(R
3628 010570          LET @DATAWT := RECcnt(R5)          ;THEN UPDATE REC COUNT TO WRITE IT ON TA
3629 010570 016577 003324 172536          MOV     RECcnt(R
3630 010576          ENDIF
3631 010576
3632 010576 012775 002310 002452          MOV     #CMDPKT,@TSDB(R5)          ;LOAD TSDB WITH CMDPKT ADDRESS
3633          ;THIS INITIATES COMMAND EXECUTION.
3634 010604          IF #TS.SSR SETIN @TSSR(R5) THEN ;IF READY DID NOT DROP THEN:
3635 010604 032775 000200 002462          BIT     #TS.SSR,
3636 010612 001410          BEQ     50102$
3637 010614 004737 011224          JSR PC,MOVMSG          ;MOVE CURRENT MESSAGE PACKET TO COMMON
3638 010620          ERROF #3,TOERM,STAERM          ;REPORT NO TS04 RESPONSE.
3639 010620 104455          TRAP   C$ERDF
3640 010622 000003          .WORD 3
3641 010624 004207          .WORD TOERM
3642 010626 005372          .WORD STAERM
3643 010630 004737 015554          JSR PC,DROPU          ;DROP THE UNIT
3644 010634          ENDIF
3645 010634
3646 010634 000207          EXCRTN: RTS PC          ;RETURN.

```



3703	010774	105737	002210				TSTB	DINT
3704	011000	001003					BNE	50111\$
3705	011002			LET R2 := INTFLG(R5)	;FETCH INTERRUPT OCCURRED FLAG.		MOV	INTFLG(R
3706	011002	016502	003416	ELSE	;IF IN BRUTUS MODE:		BR	50112\$
3707	011006							
3708	011006	000406						
3709	011010							
3710	011010			LET R3 := COMP #TS.SSR	;SET UP A MASK FOR THE DONE BIT.	50111\$:	MOV	#TS.SSR,
3711	011010	012703	000200				COM	R3
3712	011014	005103						
3713	011016			LET R2 := @TSSR(R5) CLR BY R3	;FETCH DONE BIT.		MOV	@TSSR(R5
3714	011016	017502	002462				BIC	R3,R2
3715	011022	040302						
3716	011024			ENDIF				
3717	011024							
3718	011024			LET TIME1 := TIME1 - #1	;UPDATE TIMEOUT COUNTER.	50112\$:		
3719	011024	005337	003364				DEC	TIME1
3720	011030			UNTIL R2 NE #0 OR TIME1 EQ #0	;REPEAT UNTIL INTERRUPT OR READY OCCURES		TST	R2
3721	011030	005702					BNE	50113\$
3722	011032	001003					TST	TIME1
3723	011034	005737	003364				BNE	50105\$
3724	011040	001310						
3725	011042							
3726	011042			IF TIME1 EQ #0 THEN	;IF TIME OUT HAS OCCURRED:	50113\$:		
3727	011042	005737	003364				TST	TIME1
3728	011046	001022					BNE	50114\$
3729	011050			LET @DATAWT := RECCNT(R5) #1	;RE ADJUST REC COUNT DOWN		MOV	RECCNT(R
3730	011050	016577	003324	172256			DEC	@DATAWT
3731	011056	005377	172252					
3732	011062	004737	011224	JSR PC,MOVMSG	;MOVE CURRENT MSG PACKET TO COMMON AREA.			
3733	011066			ERRDF #4,NOINTM,STAERM	;REPORT NO INTERRUPT.			
3734	011066	104455					TRAP	C\$ERDF
3735	011070	000004					.WORD	4
3736	011072	004421					.WORD	NOINTM
3737	011074	005372					.WORD	STAERM
3738	011076	004737	015554	JSR PC,DROPU	;DROP THE UNIT.			
3739	011102			LET R3 := #ENDERF				
3740	011102	012703	003416				MOV	#ENDERF,
3741	011106	004737	011154	JSR PC,CLRERR	;CLEAR ALL ERROR FLAGS			

```

3742
3743 011112
3744 011112 000417
3745 011114
3746 011114 004737 011224
3747 011120 004737 011310
3748 011124 004737 011456
3749 011130
3750 011130 105737 003407
3751 011134 001006
3752 011136 004737 014102
3753 011142
3754 011142 012703 003416
3755 011146 004737 011154
3756 011152
3757 011152
3758 011152
3759 011152
3760 011152 000207

ELSE
JSR PC,MOVMSG ;MOVE CURRENT MSG. PACKET TO COMMON AREA
JSR PC,RECUD ;UPDATE THE RECORD COUNT.
JSR PC,CHKERR ;CHECK FOR STATUS ERRORS.
IFB WRTYFG EQ #0 THEN ;
;
;LOG BYTES AND ERRORS.
TSTB WRTYFG
BNE 50116$
;CLEAR ALL ERROR FLAGS
MOV #ENDERF,
50116$:
50115$:
1$: RTS PC ;RETURN IF DONE.

```

```

3761      :      SUBROUTINE TO CLEAR FLAGS.
3762      :      INPUTS:          R3 = LWA TO BE CLEARED * 2.
3763      :      OUTPUTS:
3764      :      REGISTERS:      R2
3765      :      CALLS:
3766
3767 011154 CLRERR:: LET R2 := #BGNFLG
3768 011154 012702 003404      REPEAT
3769 011160      MOV      #BGNFLG,
3770 011160      50117$:
3771 011160      LET (R2)+ : #0
3772 011160 005022      CLR      (R2)+
3773 011162      UNTIL R2 EQ R3
3774 011162 020203      CMP      R2,R3
3775 011164 001375      BNE      50117$
3776 011166 000207      RTS PC
3777
3778
3779
3780      :      SUBROUTINE TO WAIT UNTIL CURRENT UNIT IS READY OR UNTIL TIME OUT.
3781      :      INPUTS:
3782      :      OUTPUTS:
3783      :      REGISTERS:
3784      :      CALLS:
3785
3786 011170 WSSR:: LET TIME1 := # 1      ;INIT TIMEOUT COUNTER.
3787 011170 012737 177777 003364      MOV      #-1,TIME
3788 011176      REPEAT      ;REPEAT UNTIL DEV READY OR TIMEOUT:
3789 011176      50120$:
3790 011176      BREAK      ;BREAK TO THE SUPERVISOR.
3791 011176 104422      TRAP      C$BRK
3792 011200      LET TIME1 := TIME1 #1      ;UPDATE TIMEOUT COUNTER.
3793 011200 005337 003364      DEC      TIME1
3794 011204      UNTIL #TS.SSR SETIN @TSSR(R5) OR TIME1 EQ #0
3795 011204 032775 000200 002462      BIT      #TS.SSR,
3796 011212 001003      BNE      50121$
3797 011214 005737 003364      TST      TIME1
3798 011220 001366      BNE      50120$
3799 011222
3800      50121$:
3801 011222 000207      ;REPEAT UNTIL DEV READY OR TIMEOUT.
3802      RTS PC      ;RETURN.
3803

```

```

3804
3805
3806
3807
3808
3809
3810
3811
3812 011224      MOVMSG:: LET TSSREG := @TSSR(R5)      ;FREEZE THE STATUS REG CONTENTS
3813 011224 017537 002462 003402      MOV      @TSSR(R5
3814 011232      LET R2 := TSSREG CLR.BY #TSC.TCC ;EXTRACT THE TERMINATION CLASS CODE,
3815 011232 013702 003402      MOV      TSSREG,R
3816 011236 042702 177761      BIC      #TSC.TCC
3817 011242      LET CTCC := R2 SHIFT -1      ;AND SAVE IT
3818 011242 010237 003376      MOV      R2,CTCC
3819 011246 006237 003376      ASR      CTCC
3820 011252      LET R3 := MSGPKA(R5)      ;ADR OF THIS DEVICE'S MSG.
3821 011252 016503 002502      MOV      MSGPKA(R
3822 011256      LET R2 := #0      ;CLR COUNTER.
3823 011256 005002      CLR      R2
3824 011260      WHILE R2 NE #MSGCNT DO      ;WHILE THERE ARE MORE LOCATIONS:
3825 011260      50122$:
3826 011260 020227 000016      CMP      R2,#MSGC
3827 011264 001405      BEQ      50123$
3828 011266      LET MSGPKT(R2) := (R3)*      ;MOVE MSG TO COMMON AREA.
3829 011266 012362 002334      MOV      (R3)*,MS
3830 011272      LET R2 := R2 + #2      ;UPDATE COUNTER.
3831 011272 062702 000002      ADD      #2,R2
3832 011276      ENDDO
3833 011276 000770      BR      50122$
3834 011300      50123$:
3835 011300      LET EOTFLG(R5) := MSGPKT*MS.XSO ;MOVE XSTATO TO EOT FLAG.
3836 011300 013765 002342 003426      MOV      MSGPKT*M
3837 011306 000207      RTS      PC
    
```



```

3838 ; SUBROUTINE TO ADJUST THE RECORD COUNT.
3839 ; INPUTS:
3840 ; OUTPUTS:
3841 ; REGISTERS:
3842 ; CALLS:
3843
3844 011310 RECUD:: IFB RECLOG FQ #0 THEN ;IF RECORD HAS NOT BEEN LOGGED:
3845 011310 105737 003411 TSTB RECLOG
3846 011314 001057 BNE 50124$
3847 011316 LET REC CNT(R5) := REC CNT(R5) #1
3848 011316 005365 003324 DEC REC CNT(R
3849 011322 IF #BITO NOTSETIN CTCC AND #X2.OPM SETIN MSGPKT.MS.XS2 THEN ;IF TAPE
3850 011322 032737 000001 003376 BIT #BITO.C*
3851 011330 001046 BNE 50125$
3852 011332 032737 100000 002346 BIT #X2.OPM,
3853 011340 001442 BEQ 50125$
3854 011342 LET RECLOG :B= RECLOG * #1 ;SET RECORD LOGGED.
3855 011342 105237 003411 INCB RECLOG
3856 011346 IF CMDWRD EQ #RWD THEN ;IF THIS IS A REWIND CMD:
3857 011346 023727 003346 102010 CMP CMDWRD.#
3858 011354 001003 BNE 50126$
3859 011356 LET REC CNT(R5) := #0 ;CLEAR RECORD COUNT.
3860 011356 005065 003324 CLR REC CNT(R
3861 011362 ELSE
3862 011362 000431 BR 50127$
3863 011364
3864 011364 IF #BRF.C SETIN CMDWRD THEN ;IF BRF USED, UPDATE RECORD COUN
50126$:
3865 011364 032737 004000 003346 BIT #BRF.C.C
3866 011372 001425 BEQ 50130$
3867 011374 IF #MOD.CO NOTSETIN CMDWRD THEN ;IF A FORWARD CMD:
3868 011374 032737 000400 003346 BIT #MOD.CO,
3869 011402 001007 BNE 50131$
3870 011404 IF #MOD.CO NOTSETIN PCMDWD THEN ;IF PREV CMD WAS A FWD ALSO:
3871 011404 032737 000400 003352 BIT #MOD.CO,
3872 011412 001002 BNE 50132$
3873 011414 LET REC CNT(R5) := REC CNT(R5) * #1 ;INCREMENT RECORD COUNT.
3874 011414 005265 003324 INC REC CNT(R
3875 011420 ENDF
3876 011420 ELSE ;IF REVERSE CMD:
50132$:
3877 011420 BR 50133$
3878 011420 000412
3879 011422 IF #MOD.CO SETIN PCMDWD THEN ;IF PREVIOUS CMD WAS A REV ALSO:
50131$:
3880 011422 032737 000400 003352 BIT #MOD.CO,
3881 011430 001406 BEQ 50134$
3882 011432 IF #X0.BOT NOTSETIN EDTFLG(R5) THEN ;WHEN NOT AT BOT THEN
3883 011432 032765 000002 003426 BIT #X0.BOT,
3884 011440 001002 BNE 50135$
3885 011442 LET REC CNT(R5) := REC CNT(R5) #1 ;DECREMENT RECORD COUN*
3886 011442 005365 003324 DEC REC CNT,R
3887 011442 ENDF
3888 011446 ENDF
3889 011446 ENDF
3890 011446 ENDF
3891 011446 ENDF
3892 011446 ENDF
3893 011446 ENDF
50135$:
50134$:
50133$:

```

3894 011446  
 3895 011446  
 3896 011446  
 3897 011446  
 3898 011446  
 3899 011446  
 3900 011446  
 3901 011446 016577 003324 171660  
 3902 011454  
 3903 011454  
 3904 011454 000207

ENDIF  
 ENDIF  
 ENDIF  
 LET @DATAWT := RECCNT(R5)  
 ENDIF  
 RTS PC ;RETURN.

501301:  
 501271:  
 501251:  
 MOV RECCNT(R  
 501241:

[ ] 3

```

3905      ; THIS IS THE ERROR CHECK SUBROUTINE. AFTER INTERRUPT THIS
3906      ; SUBROUTINE IS CALLED TO CHECK THE TS04 STATUS.
3907      ; IF SPECIAL COND IS SET THEN THE TCC HANDLING SUBROUTINE IS ENTERED.
3908      ; IF THE RFC IS NON ZERO FOR A COMMAND REQUIRING A BPCR,
3909      ; THEN AN ERROR RFC IS REPORTED.
3910      ; INPUTS:
3911      ; OUTPUTS:
3912      ; REGISTERS:      R2, R4.
3913      ; CALLS:          TCC0 TCC7.
3914
3915      ;CHKERR:: IF DEVTBL(R5) EQ #NINUSE THEN ;BTL
3916      ;                                ;                                CMP      DEVTBL(R
3917      ;                                ;                                BNE      50136$
3918      ;                                JMP 1$ ;BTL
3919      ;                                ELSE ;BTL
3920      ;                                ;                                BR      50137$
3921      ;                                ;                                50136$:
3922      ;                                ;                                50137$:
3923      ;                                ;                                50137$:
3924      ;                                IF #TS.SC SET IN TSSREG THEN ;IF SPECIAL COND STATUS IS SET T
3925      ;                                ;                                BIT      #TS.SC,T
3926      ;                                ;                                BEQ      50140$
3927      ;                                ;                                IF CTCC NE #2 THEN ;IF TCC IS NOT 2 THEN:
3928      ;                                ;                                ;                                CMP      CTCC,#2
3929      ;                                ;                                ;                                BEQ      50141$
3930      ;                                ;                                IFB ERRREC EQ #0 THEN ;IF NOT IN ERROR RECOVERY:
3931      ;                                ;                                ;                                TSTB   ERRREC
3932      ;                                ;                                ;                                BNE      50142$
3933      ;                                ;                                ;                                INC      SCCNT(R5) ;INC SC COUNTER.
3934      ;                                ;                                ;                                ENDIF
3935      ;                                ;                                ;                                50142$:
3936      ;                                ;                                ;                                ENDIF
3937      ;                                ;                                ;                                50141$:
3938      ;                                ;                                ;                                IF #TS.NXM SET IN TSSREG OR #TS.UPE SET IN TSSREG THEN ;WHEN NON-EXISTA
3939      ;                                ;                                ;                                BIT      #TS.NXM,
3940      ;                                ;                                ;                                BNE      50143$
3941      ;                                ;                                ;                                BIT      #TS.UPE,
3942      ;                                ;                                ;                                BEQ      50144$
3943      ;                                ;                                ;                                ;                                50143$:
3944      ;                                ;                                ;                                IF #X2.OPM NOT SET IN MSGPKT.MS.XS2 THEN ;AND TAPE NOT MO
3945      ;                                ;                                ;                                BIT      #X2.OPM,
3946      ;                                ;                                ;                                BNE      50145$
3947      ;                                ;                                ;                                ;                                LET R2 := #5 ;SET TCC5 INDEX
3948      ;                                ;                                ;                                MOV      #5,R2
3949      ;                                ;                                ;                                ELSE
3950      ;                                ;                                ;                                ;                                BR      50146$
3951      ;                                ;                                ;                                ;                                50145$:
3952      ;                                ;                                ;                                ;                                LET R2 := #4 ;TAPE MOVED. SET TCC4 INDEX
3953      ;                                ;                                ;                                ;                                MOV      #4,R2
3954      ;                                ;                                ;                                ;                                ENDIF
3955      ;                                ;                                ;                                ;                                50146$:
3956      ;                                ;                                ;                                ;                                ELSE
3957      ;                                ;                                ;                                ;                                ;                                BR      50147$
3958      ;                                ;                                ;                                ;                                ;                                ;                                50144$:
3959      ;                                ;                                ;                                ;                                ;                                LET R2 := CTCC ;SET DETECTED TCC INDEX
3960      ;                                ;                                ;                                ;                                ;                                MOV      CTCC,R2
    
```

```

3961 011576          ENDIF
3962 011576          50147$:
3963 011576          LET R2 := R2 SHIFT 1 ;CURRENT TCC X 2.
3964 011576 006302          ASL      R2
3965 011600 004772 011700  JSR    PC,@TCCRA(R2)      ;GO TO THE TCC HANDLING SUBROUTINE.
3966 011604          ELSE
3967 011604 000426          BR      50150$
3968 011606          50140$:
3969 011606          IF @BRF.C SETIN CMDWRD THEN ;IF BRF IS USED IN THIS CMD THEN:
3970 011606 032737 004000 003346  BIT    @BRF.C,C
3971 011614 001422          BEQ    50151$
3972 011616          IF MSGPKT.MS.RFC NE @0 THEN ;IF THERE IS AN RFC THEN:
3973 011616 005737 002340  TST    MSGPKT.M
3974 011622 001417          BEQ    50152$
3975 011624          IFB RANDOM EQ @0 ORB VFYFLG NE @0 THEN
3976 011624 105737 003441          TSTB   RANDOM
3977 011630 001403          BEQ    50153$
3978 011632 105737 003442          TSTB   VFYFLG
3979 011636 001411          BEQ    50154$
3980 011640          50153$:
3981          ;IF NOT IN RANDOM OR IF CMD IS WTV:
3982 011640          IFB IRE EQ @0 THEN ;IF RFC ERROR REPORTS ARE ALLOWED:
3983 011640 105737 003445          TSTB   IRE
3984 011644 001006          BNE    50155$
3985 011646          LET HRDCNT(R5) := HRDCNT(R5) + @1 ;UPDATE HARD ERROR COUNT
3986 011646 005265 003304          INC    HRDCNT(R
3987 011652          ERRHRD @13,RFCERM,STAERM ;REPORT RFC ERROR
3988 011652 104456          TRAP   C$ERRHRD
3989 011654 000015          .WORD  13
3990 011656 004254          .WORD  RFCERM
3991 011660 005372          .WORD  STAERM
3992 011662          ENDIF
3993 011662          50155$:
3994 011662          ENDIF
3995 011662          50154$:
3996 011662          ENDIF
3997 011662          50152$:
3998 011662          ENDIF
3999 011662          50151$:
4000 011662          ENDIF
4001 011662          50150$:
4002 011662          IFB RWERR NE @0 THEN ;IF A READ/WRITE ERROR HAS OCCURRED THEN
4003 011662 105737 003413          TSTB   RWERR
4004 011666 001403          BEQ    50156$
4005 011670          LET CMUPKT := CMDSAV ;RESTORE CMD PACKET AFTER ERROR RECCV.
4006 011670 013737 003350 002310  MOV    CMDSAV,C
4007 011676          ENDIF
4008 011676          50156$:
4009 011676 000207          1$:   RTS    PC      ;RETURN.

```

4010			:	ADDRESSES OF TCC HANDLING ROUTINES FOR TERMINATION CLASS CODES 0 - 7.
4011				
4012	011700	011720	TCCRA:	TCC0
4013	011702	011736		TCC1
4014	011704	011754		TCC2
4015	011706	012064		TCC3
4016	011710	012102		TCC4
4017	011712	012552		TCC5
4018	011714	012650		TCC6
4019	011716	012700		TCC7

```

4020 ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 0, UNDEFINED SPECIAL
4021 ; CONDITION ERROR.
4022 ; INPUTS:
4023 ; OUTPUTS:
4024 ; REGISTERS:
4025 ; CALLS:
4026
4027 011720 TCC0:: LET HRDCNT(R5) := HRDCNT(R5) + #1 ;UPDATE HARD ERROR COUNT.
4028 011720 005265 003304 ;INC HRDCNT(R
4029 011724 ERRHRD #5,SCERM,STAERM ;REPORT SPECIAL CONDITION ERROR.
4030 011724 104456 TRAP C$ERRHRD
4031 011726 000005 .WORD 5
4032 011730 004230 .WORD SCERM
4033 011732 005372 .WORD STAERM
4034 011734 000207 RTS PC ;RETURN.
4035
4036
4037
4038
4039
4040 ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 1, ATTENTION CONDITION.
4041 ; THIS TCC INDICATES THAT THE DRIVE HAS UNDERGONE A STATUS CHANGE
4042 ; SUCH AS GOING OFFLINE OR COMING ONLINE.
4043 ; INPUTS:
4044 ; OUTPUTS:
4045 ; REGISTERS: R2,R4
4046 ; CALLS: DROPU
4047
4048 011736 TCC1:: ERRDF #6,ATTNM,STAERM ;REPORT ATTENTION-UNIT OFF LINE.
4049 011736 104455 TRAP C$ERRDF
4050 011740 000006 .WORD 6
4051 011742 004335 .WORD ATTNM
4052 011744 005372 .WORD STAERM
4053 011746 004737 015554 JSR PC,DROPU ;DROP THE UNIT.
4054 011752 000207 RTS PC ;RETURN.

```

```

4055 ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 2, TAPE STATUS ALERT.
4056 ; A STATUS CONDITION HAS BEEN ENCOUNTERED THAT MAY HAVE SIGNIFICANCE
4057 ; TO THE PROGRAM. BITS OF INTEREST INCLUDE TMK, RLS, LET, RLL, BOT, EOT.
4058 ; INPUTS:
4059 ; OUTPUTS:
4060 ; REGISTERS:
4061 ; CALLS:
4062
4063 TCC2:: IF #X0.BOT SETIN MSGPKT*MS.XSO ANDB EXPBOT NE #0 THEN
4064 011754 032737 000002 002342 BIT #X0.BOT,
4065 011762 001404 BEQ 50157$
4066 011764 105737 003440 TSTB EXPBOT
4067 011770 001401 BEQ 50157$
4068 ; IF AT BOT AND BOT IS EXPECTED:
4069 011772 000433 BR TC2RTN ;RETURN TCC2 CAUSED BY EXPECTED BOT.
4070 011774 ENDF
4071 011774
4072 011774 50157$:
4073 011774 032737 170002 002342 IF #X0.RLS!X0.RLL!X0.TMK!X0.LET!X0.BOT SETIN MSGPKT*MS.XSO THEN
4074 012002 001427 BIT #X0.RLS!
4075 BEQ 50160$
4076 012004 ; IF TCC2 CAUSED BY ANYTHING BUT EOT:
4077 012004 105737 003441 IFB RANDOM EQ #0 ORB VFYFLG NE #0 THEN
4078 012010 001403 TSTB RANDOM
4079 012012 105737 003442 BEQ 50161$
4080 012016 001421 TSTB VFYFLG
4081 012020 BEQ 50162$
4082
4083 012020 IFB IRE EQ #0 THEN 50161$:
4084 012020 105737 003445 ; IF NOT IN RANDOM OR IF CMD IS WTV:
4085 012024 001016 ; IF RFC ERROR REPORTS ARE ALLOWED:
4086 012026 IFB ERRREC NE #0 THEN TSTB IRE
4087 012026 105737 003415 BNE 50163$
4088 012032 001403 ; IF WE ARE IN ERROR RECOVERY THE
4089 012034 LET UNREC :B UNREC * #1 ;SET UNRECOVERABLE FLAG FOR LO
4090 012034 105237 003414 TSTB ERRREC
4091 012040 ELSE ;ELSE - IF NOT IN ERROR RECOVERY
4092 012040 000402 BEQ 50164$
4093 012042 BR 50165$
4094 012042 LET SCCNT(R5) := SCCNT(R5) * #1 ;INCREMENT THE SPEC COND COUNT
4095 012042 005265 003264 INC SCCNT(R5)
4096 012046 ENDF
4097 012046
4098 012046 LET HRDCNT(R5) := HRDCNT(R5) * #1 ;UPDATE HARD ERROR COUNT.
4099 012046 005265 003304 INC HRDCNT(R)
4100 012052 ERRHRD #7,TSAM,STAERM ;REPORT TAPE STATUS ALERT.
4101 012052 104456 TRAP C$ERRHRD
4102 012054 000007 .WORD ?
4103 012056 004436 .WORD TSAM
4104 012060 005372 .WORD STAERM
4105 012062 ENDF
4106 012062 50163$:
4107 012062 ENDF
4108 012062 50162$:
4109 012062 ENDF
4110 012062 50160$:
    
```

```

4111 012062 000207          TC2RTN:  RTS PC          ;RETURN.
4112
4113
4114
4115
4116
4117
4118          ;      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 3, FUNCTION REJECT.
4119          ;      THE SPECIFIED FUNCTION WAS NOT INITIATED. BITS OF INTEREST ARE
4120          ;      RMR, OFL, VCK, BOT, ILC, WLE, ILA, AND NBA.
4121          ;      INPUTS:
4122          ;      OUTPUTS:
4123          ;      REGISTERS:      R2,R4
4124          ;      CALLS:          DROPU
4125 012064          TCC3::  ERRDF #8,FUNRM,STAERM          ;REPORT FUNCTION REJECT.
4126 012064 104455          ;
4127 012066 000010          ;          TRAP          C$ERDF
4128 012070 004353          ;          .WORD          8
4129 012072 005372          ;          .WORD          FUNPM
4130 012074 004737 015554          ;          .WORD          STAERM
4131 012100 000207          JSR PC,DROPU          ;DROP THE UNIT.
          RTS PC          ;RETURN.

```



```

4132 : SUBROUTINE TO HANDLE TERMINATION CLASS CODE 4, RECOVERABLE ERROR.
4133 : TAPE POSITION IS ONE RECORD BEYOND WHAT ITS POSITION WAS WHEN
4134 : THE FUNCTION WAS INITIATED. RECOVERY PROCEDURE IS TO LOG THE
4135 : ERROR AND ISSUE THE APPROPRIATE RETRY COMMAND.
4136 : 2 WRITE ERROR RECOVERY ALGORITHMS CAN BE SELECTED:
4137 : THE FIRST ONE, VIA BADTSW SWITCH, DOES DETECT BAD SPOTS ON TAPE.
4138 : IT CALLS A WRITE RETRY SUBR UNTIL THE RECORD IS RECOVERED
4139 : OR 20 BAD SPOTS HAVE BEEN LOGGED. ON REACHING 20 BAD
4140 : SPOTS LOGGED, A BAD TAPE OVERFLOW MSG IS PRINTED AND THE
4141 : UNIT DROPPED.
4142 : THE SECOND ALGORITHM ISSUES THE TS11 WRITE RETRY COMMAND
4143 : UP TO 16 TIMES BEFORE DROPPING THE UNIT OR PROCEEDING
4144 : WITH THE NEXT RECORD ON RECOVERY.
4145 : INPUTS:
4146 : OUTPUTS:
4147 : REGISTERS: R2,R4.
4148 : CALLS: RTLE, EXECUTE, GOWAIT, DROPU, WRTY
4149
4150 012102 TCC4:: IF DEVTBL(R5) EQ #NINUSE THEN ;BTL
4151 012102 026527 002532 177774 CMP DEVTBL(R
4152 012110 001003 BNE 50166$
4153 012112 000137 012550 JMP 3$ ;BTL
4154 012116 ELSE ;BTL
4155 012116 000400 BR 50167$
4156 012120 50166$:
4157 012120 ENDIF ;BTL
4158 012120 50167$:
4159 012120 IF CMDLG EQ #2 ANDB BADTSW NE #0 THEN
4160 012120 023727 003354 000002 CMP CMDLG,#2
4161 012126 001134 BNE 50170$
4162 012130 105737 002206 TSTB BADTSW
4163 012134 001531 BEQ 50170$
4164 012136 IFB ERRREC EQ #0 ANDB ERCVER NE #0 THEN
4165 012136 105737 003415 TSTB ERRREC
4166 012142 001007 BNE 50171$
4167 012144 105737 002205 TSTB ERCVER
4168 012150 001404 BEQ 50171$
4169 012152 ERRSOFT #9,RERM,STAERM ;
4170 012152 104457 TRAP C$ERSOFT
4171 012154 000011 .WORD 9
4172 012156 004550 .WORD RERM
4173 012160 005372 .WORD STAERM
4174 012162 ENDIF
4175 012162 50171$:
4176 012162 IFB IREC EQ #0 THEN ;
4177 012162 105737 002211 TSTB IREC
4178 012166 001111 BNE 50172$
4179 012170 LET ERRREC :B= ERRREC + #1 ;RETRY FLAG FOR EXECUTE SUBR: DON'T UPDAT
4180 012170 105237 003415 INCB ERRREC
4181 012174 LET WRTYER :B= WRTYER + #1 ;REWRITE ERROR FLAG FOR WRTY SUBR
4182 012174 105237 003410 INCB WRTYER
4183 012200 IFB WRTYFG EQ #0 THEN ;FIRST RETRY ON THIS RECORD: SUBSEQUENT
4184 012200 105737 003407 TSTB WRTYFG
4185 012204 001101 BNE 50173$
4186 ;RETRIES WITH TCC4 ERRORS BY PASS THIS
4187 012206 LET WTYWRD :- CMDWRD ;SAVE WRITE COMMAND PACKET

```

4188 012206 013737 003346 013366  
4189 012214  
4190 012214 013737 002310 013364  
4191 012222  
4192 012222 013737 002316 013370  
4193 012230  
4194 012230 105237 003413  
4195 012234  
4196 012234 105237 003407  
4197 012240  
4198 012240  
4199 012240  
4200 012240 005265 003244  
4201 012244  
4202 012244 005037 003404  
4203 012250  
4204 012250 105037 003406  
4205 012254 004737 013044  
4206 012260  
4207 012260 026527 002532 177774  
4208 012266 001003  
4209 012270 000137 012550  
4210 012274  
4211 012274 000400  
4212 012276  
4213 012276  
4214 012276  
4215 012276  
4216 012276 105737 003410  
4217 012302 001404  
4218 012304 027727 171126 000050  
4219 012312 103752  
4220 012314  
4221  
4222 012314  
4223 012314 027727 171116 000050  
4224 012322 103423  
4225 012324  
4226 012324 012746 013457  
4227 012330 012746 000001  
4228 012334 010600  
4229 012336 104414  
4230 012340 062706 000004  
4231 012344 004737 013576  
4232 012350  
4233 012350 005365 003324  
4234 012354 004737 015554  
4235 012360  
4236 012360 005065 003324  
4237 012364  
4238 012364 012775 002330 002452  
4239 012372  
4240 012372  
4241 012372  
4242 012372 105037 003407  
4243 012376

```

MOV CMDWRD,W
LET WTYCMD := CNDPKT ;
MOV CNDPKT,W
LET WTYBRF := CNDPKT*CP.CNT ;
MOV CNDPKT*C
LET RWERR :B= RWERR * #1 ;LOG SUBR FLAG: COUNT WRT ERRORS
INCB RWERR
LET WRTYFG :B= WRTYFG * #1 ;RETRY IN PROGRESS FLAG
INCB WRTYFG
REPEAT
50174$:
LET WRTYCT(R5) := WRTYCT(R5) * #1 ;COUNT GLOBAL WRITE RETR
INC WRTYCT(R
LET RETRYC := #0 ;CLEAR # OF RETRIES PER RECORD
CLR RETRYC
LET RPTCNT :B= #0 ;CLEAR # OF REPEATS
CLRB RPTCNT
JSR PC,WRTY ;CALL WRITE RETRY
IF DEVTBL(R5) EQ #NINUSE THEN ;BTL
CMP DEVTBL(R
BNE 50175$
JMP 3$ ;BTL
ELSE ;BTL
BR 50176$
50175$:
ENDIF ;BTL
50176$:
UNTILB WRTYER EQ #0 OR @BTPT HIS #40. ;REPEAT RETRIES ON SAME
TSTB WRTYER
BEQ 50177$
CMP @BTPT,#4
BLO 50174$
50177$:
IF @BTPT HIS #40. THEN ;UNTIL RECOVERED OR 20 BAD SPOTS
;WHEN 20 BAD SPOTS LOGGED
CMP @BTPT,#4
BLO 50200$
PRINTB @BTMSG2 ;PRINT BAD TAPE OVERFLOW MSG
MOV @BTMSG2,
MOV #1,(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP
JSR PC,BORERS ;ERASE BAD RECORD
LET RECCNT(R5) := RECCNT(R5) #1 ;
DEC RECCNT(R
JSR PC,DROPU ;DROP UNIT
LET RECCNT(R5) := #0 ;
CLR RECCNT(R
LET @TSDB(R5) := #RWCPK ;REWIND UNIT
MOV #RWCPK,@
ENDIF
50200$:
LET WRTYFG :B= #0 ;RETRY COMPLETE FLAG
CLRB WRTYFG
LET MISCFG :B= MISCFG * #1 ;DO NOT HALT ON THIS CMD FLG

```

```

4244 012376 105237 003455
4245 012402
4246 012402 013737 013366 003352
4247 012410
4248 012410
4249 012410
4250 012410 000402
4251 012412
4252 012412
4253 012412 105237 003414
4254 012416
4255 012416
4256 012416
4257 012416 000454
4258 012420
4259 012420 004737 012716
4260 012424
4261 012424 023727 003354 000002
4262 012432 003411
4263 012434
4264 012434 012702 000020
4265 012440 006202
4266 012442
4267 012442 023702 003404
4268 012446 002403
4269 012450
4270 012450 052737 020000 002310
4271 012456
4272 012456
4273 012456
4274 012456
4275 012456
4276 012456 005737 003404
4277 012462 001007
4278 012464 105737 002205
4279 012470 001404
4280 012472
4281 012472 104457
4282 012474 000011
4283 012476 004550
4284 012500 005372
4285 012502
4286 012502
4287 012502
4288 012502 005237 003404
4289 012506
4290 012506 052737 001000 002310
4291 012514
4292 012514 105737 002211
4293 012520 001011
4294 012522
4295 012522 105237 003415
4296 012526
4297 012526 012602
4298 012530 012602
4299 012532 004737 010326

      LET PCMDWD := WTYWRD          ;RESTORE ORIGINAL WRT CMD AFTER
      ;MOV INCB MISCFCG
      ;RECOVERY WTYWRD,P
      ENDIF
      ELSE
      50173$:
      BR 50201$
      50172$:
      INCB UNREC
      50201$:
      BR 50202$
      50170$:
      ;CHECK FOR RETRY LIMIT EXCEEDED.
      ;IF READ CMD THEN:
      JSR PC,RTLE
      IF CMDLG GT #2 THEN
      LET R2 := #RRECL SHIFT 1 ;R2=READ RETRY COUNT LIMIT / 2
      ;CMP CMDLG,#2
      ;BLE 50203$
      ;MOV #RRECL,R
      ;ASR R2
      IF RETRYC GE R2 THEN ;IF RETRY COUNT IS MORE THAN HAL
      ;CMP RETRYC,R
      ;BLT 50204$
      LET CMDPKT := CMDPKT SET.BY #OPP.C ;SET OPPOSITE BIT FOR RE
      ;BIS #OPP.C,C
      ENDIF
      50204$:
      ENDIF
      50203$:
      IF RETRYC EQ #0 ANDB ERCVER NE #0 THEN ;IF THIS IS THE ORIGINAL ERROR
      ;TST RETRYC
      ;BNE 50205$
      ;TSTB ERCVER
      ;BEQ 50205$
      ERRSOFT #9,RERM,STAERM ;REPORT RECOVERABLE ERROR
      ;TRAP C$ERSOFT
      ;.WORD 9
      ;.WORD RERM
      ;.WORD STAERM
      ENDIF ;PROVIDED OPERATOR HAS ENABLED THE REPOR
      50205$:
      LET RETRYC := RETRYC + #1 ;UPDATE RETRY COUNT.
      ;INC RETRYC
      LET CMDPKT := CMDPKT SET.BY #MOD.C1 ;SET RETRY BIT IN CMD PACKET.
      ;BIS #MOD.C1,
      IFB IREC EQ #0 THEN ;IF ERROR RECOVERY ENABLED:
      ;TSTB IREC
      ;BNE 50206$
      LET ERRREC := ERRREC + #1 ;SET ERROR RECOVERY FLAG.
      ;INCB ERRREC
      POP R2,R2 ;POP 2 RTN ADRS FROM STACK.
      ;MOV (SP),R2
      ;MOV (SP),R2
      JSR PC,EXECUTE ;GO EXECUTE THE RETRY COMMAND.

```

```

4300 012536 000137 010636          JMP GOWAIT          ;GO WAIT FOR INTERRUPT + CHECK STATUS.
4301 012542                                ELSE                ;ELSE IF ERROR RECOVERY IS NOT ENABLED:
4302 012542 000402                                BR                  50207$
4303 012544                                ;SET UNRECOVERABLE ERROR FLAG.
4304 012544                                LET UNREC :B= UNREC + #1 ;SET UNRECOVERABLE ERROR FLAG.
4305 012544 105237 003414                                INCB                UNREC
4306 012550                                ENDIF
4307 012550                                ;SET UNRECOVERABLE ERROR FLAG.
4308 012550                                ;SET UNRECOVERABLE ERROR FLAG.
4309 012550                                ;SET UNRECOVERABLE ERROR FLAG.
4310 012550 000207          3$: RTS PC          ;RETURN          50202$:

```

```

4311      : SUBROUTINE TO HANDLE TERMINATION CLASS CODE 5, RECOVERABLE ERROR.
4312      : TAPE POSITION HAS NOT CHANGED. RECOVERY PROCEDURE IS TO LOG THE
4313      : ERROR AND RE-ISSUE THE ORIGINAL COMMAND.
4314      : INPUTS:
4315      : OUTPUTS:
4316      : REGISTERS:      R2,R4.
4317      : CALLS:          RTLE, EXCUTE, GOWAIT, DROPU.
4318
4319 012552 004737 012716      TCC5:: JSR PC,RTLE          ;CHECK FOR RETRY LIMIT EXCEEDED
4320 012556      IF RETRYC EQ #0 THEN      ;IF THIS IS THE ORIGINAL ERROR THEN:
4321 012556 005737 003404      TST          RETRYC
4322 012562 001004      BNE          50210$
4323 012564      ERRSOFT #10,RERM,STAERM ;REPORT RECOVERABLE ERROR.
4324 012564 104457      TRAP          C$ERSOFT
4325 012566 000012      .WORD        10
4326 012570 004550      .WORD        RERM
4327 012572 005372      .WORD        STAERM
4328 012574      ENDIF
4329 012574      50210$:
4330 012574      LET RETRYC := RETRYC * #1 ;UPDATE RETRY COUNTER.
4331 012574 005237 003404      INC          RETRYC
4332 012600      IFB IREC EQ #0 THEN      ;IF ERROR RECOVERY IS ENABLED:
4333 012600 105737 002211      TSTB          IREC
4334 012604 001016      BNE          50211$
4335 012606      LET ERRREC :B= ERRREC * #1 ;SET ERROR RECOVERY FLAG.
4336 012606 105237 003415      INCB          ERRREC
4337 012612      LET RECCNT(R5) := RECCNT(R5) * #1 ;UPDATE REC COUNT
4338 012612 005265 003324      INC          RECCNT(R
4339 012616      LET @DATAWT := RECCNT(R5) ;AND INSERT IT INTO WRT BFR
4340 012616 016577 003324 170510      MOV          RECCNT(R
4341 012624      POP R2,R2                ;POP 2 RTN ADRS FROM STACK.
4342 012624 012602      MOV          (SP)+,R2
4343 012626 012602      MOV          (SP)+,R2
4344 012630 004737 010326      JSR PC,EXCUTE      ;GO RE ISSUE THE COMMAND.
4345 012634 000137 010636      JMP GOWAIT      ;GO WAIT FOR INTERRUPT * CHECK STATUS.
4346 012640      ELSE                    ;ELSE IF ERROR RECOVERY IS NOT ENABLED:
4347 012640 000402      BR          50212$
4348 012642      50211$:
4349 012642      LET UNREC :B= UNREC * #1 ;SET UNRECOVERABLE ERROR FLAG.
4350 012642 105237 003414      INCB          UNREC
4351 012646      ENDIF
4352 012646      50212$:
4353 012646 000207      RTS PC          ;RETURN.
4354
4355

```

```

4356 ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 6, UNRECOVERABLE ERROR.
4357 ; TAPE POSITION HAS BEEN LOST. THE ONLY VALID RECOVERY PROCEDURE
4358 ; IS TO REWIND AND START OVER AT BOT UNLESS THE TAPE HAS LABELS OR
4359 ; SEQUENCE NUMBERS. THIS DIAGNOSTIC WILL REWIND AND RETRY THE
4360 ; COMMAND ONLY IF DENSITY CHECK IS SET, OTHERWISE THE UNIT WILL BE
4361 ; DROPPED FROM THE TEST SEQUENCE.
4362 ; INPUTS:
4363 ; OUTPUTS:
4364 ; REGISTERS: R2, R4
4365 ; CALLS: RTIF, WSSR, EXECUTE, GOWAIT, DROPU
4366
4367 012650 TCC6:: LET @TSDB(R5) :- @RWCPK ;ISSUE A REWIND COMMAND.
4368 012650 012775 002330 002452 ; MOV @RWCPK,@
4369 012656 004737 011170 JSR PC,WSSR ;WAIT FOR SUBSYSTEM READY.
4370 012662 ERRDF #11,URERM,STAERM ;REPORT UNRECOVERABLE ERROR.
4371 012662 104455 TRAP C:ERRDF
4372 012664 000013 .WORD 11
4373 012666 004572 .WORD URERM
4374 012670 005372 .WORD STAERM
4375 012672 004737 015554 JSR PC,DROPU ;REPORT ERROR - DROP UNIT.
4376 012676 000207 RTS PC ;RETURN
  
```



)

```

4433 013004 004737 015554          JSR PC,DROPU          ;DROP THE UNIT.
4434 013010                          POP R2
4435 013010 012602                          MOV      (SP)+,R2
4436 013012                          ENDIF
4437 013012                          50215$:
4438 013012 ELSE                          ;ELSE - CMD IS A READ:
4439 013012 000413                          BR      50216$
4440 013014                          50214$:
4441 013014 IF RETRYC EQ 0RRECL THEN        ;IF RETRY COUNT HAS REACHED LIMIT:
4442 013014 023727 003404 000020          CMP     RETRYC,0
4443 013022 001007                          BNE    50217$
4444 013024                          LET UNREC :B= UNREC + 01 ;SET UNRECOVERABLE FLAG
4445 013024 105237 003414          ERRHRD 014,RLEXM,STAERM ;REPORT RECOVERABLE ERROR.
4446 013030                          INCB   UNREC
4447 013030 104456          TRAP   C$ERRHRD
4448 013032 000016          .WORD  14
4449 013034 004310          .WORD  RLEXM
4450 013036 005372          .WORD  STAERM
4451 013040                          POP R2
4452 013040 012602                          MOV     (SP)+,R2
4453 013042                          ENDIF
4454 013042                          50217$:
4455 013042                          ENDIF
4456 013042                          50216$:
4457 013042 000207          RTLRTN: RTS PC          ;RETURN

```







4570	013306	105037	003413					
4571	013312			LET RPTCNT :B= #0		;CLEAR REPEAT COUNT FOR	CLRB	RWERR
4572	013312	105037	003406				CLRB	NEXT RET
4573	013316			ENDIF				RPTCNT
4574	013316							
4575	013316			UNTIL RETRYC EQ #4		50234\$:		
4576	013316	023727	003404			;LIMIT: 4 RETRIES		
4577	013324	001256	000004				CMP	RETRYC.#
4578	013326			END RETRY			BNE	50223\$
4579	013326							
4580	013326			IFB WRTYER NE #0 THEN		50222\$:		
4581	013326	105737	003410					
4582	013332	001413					TSTB	WRTYER
4583	013334						BEQ	50240\$
4584	013334	105737	002205	IFB ERCVER NE #0 THEN				
4585	013340	001410					TSTB	ERCVER
4586	013342						BEQ	50241\$
4587	013342	012746	013527	PRINTB #BTMSG3		;PRINT RETRY FAILED		
4588	013346	012746	000001				MOV	#BTMSG3.
4589	013352	010600					MOV	#1. (SP)
4590	013354	104414					MOV	SP,R0
4591	013356	062706	000004				TRAP	C\$PNTB
4592	013362			ENDIF			ADD	#4.SP
4593	013362					50241\$:		
4594	013362			ENDIF				
4595	013362					50240\$:		
4596	013362	000207		1\$: RTS PC				
4597								
4598								
4599								
4600								

```

4601
4602 013364 000000      WTYCMD: .WORD 0      ;STORAGE FOR WRITE CMD WHILE RETRYING
4603 013366 000000      WTYWRD: .WORD 0      ;STORAGE FOR WRITE CMD WORD WHILE RETRYING
4604 013370 000000      WTYBRF: .WORD 0      ;STORAGE FOR WRITE BPCR WHILE RETRYING
4605
4606
4607 013372 040445 052523 050123      BTMSG1: .ASCIZ /#ASUSPECT BAD SPOT AFTER #D1#A RETRY, #D1#A REPEAT#N/
4608 013400 041505 020124 040502
4609 013406 020104 050123 052117
4610 013414 040440 052106 051105
4611 013422 022440 030504 040445
4612 013430 051040 052105 054522
4613 013436 020054 042045 022461
4614 013444 020101 042522 042520
4615 013452 052101 047045 000
4616 013457 045 022516 041101      BTMSG2: .ASCIZ /#N#ABAD TAPE OVERFLOW: CHANGE TAPE!#N#N/
4617 013464 042101 052040 050101
4618 013472 020105 053117 051105
4619 013500 046106 053517 020072
4620 013506 044103 047101 042507
4621 013514 052040 050101 020505
4622 013522 047045 047045 000
4623 013527 045 051101 052105      BTMSG3: .ASCIZ /#ARETRY FAILED ON BAD SPOT...ERASED!#N/
4624 013534 054522 043040 044501
4625 013542 042514 020104 047117
4626 013550 041040 042101 051440
4627 013556 047520 027124 027056
4628 013564 051105 051501 042105
4629 013572 022441 000116
4630
      .EVEN

```

I 9

```

4631 ; SUBR TO BACSPACE ONE RECORD
4632 ; IF THE ERASE FLAG IS SET, THEN ERASE THAT RECORD
4633 ; INPUTS: ERSFLG 1 = DO ERASE
4634 ; OUTPUT:
4635 ; REGISTERS:
4636 ; CALLS: EXECUTE, GOWAIT, CKHAE
4637
4638 BORERS:: LET PCMDWD := CMDWRD ;SET COMMAND TO SPACE REV
4639 013576 013737 003346 003352 MOV CMDWRD,P
4640 013604 LET CMDWRD := #SRR ;
4641 013604 012737 104410 003346 MOV #SRR,CMD
4642 013612 LET CNDPKT := CMDWRD CLR.BY #BRF.C ;
4643 013612 013737 003346 002310 MOV CMDWRD,C
4644 013620 042737 004000 002310 BIC #BRF.C,C
4645 013626 LET CMDSAV := CNDPKT ;
4646 013626 013737 002310 003350 MOV CNDPKT,C
4647 013634 LET CNDPKT.CP.ADL := #1 ;
4648 013634 012737 000001 002312 MOV #1,CNDPK
4649 013642 LET CMDLG := #0 ;
4650 013642 005037 003354 CLR CMDLG
4651 013646 004737 007344 JSR PC,CMDAC ;
4652 013652 004737 010326 JSR PC,EXECUTE ;
4653 013656 004737 010636 JSR PC,GOWAIT ;
4654 013662 004737 016060 JSR PC,CKHAE ;
4655 013666 IFB ERSFLG NE #0 THEN ;WHEN ERASE FLAG IS SET, DO ERASE
4656 013666 105737 003451 TSTB ERSFLG
4657 013672 001426 BEQ 50242$
4658 013674 LET PCMDWD := CMDWRD ;
4659 013674 013737 003346 003352 MOV CMDWRD,P
4660 013702 LET CMDWRD := #ERS ;
4661 013702 012737 100411 003346 MOV #ERS,CMD
4662 013710 LET CNDPKT := CMDWRD ;
4663 013710 013737 003346 002310 MOV CMDWRD,C
4664 013716 LET CMDSAV := CNDPKT ;
4665 013716 013737 002310 003350 MOV CNDPKT,C
4666 013724 004737 007344 JSR PC,CMDAC ;
4667 013730 004737 010326 JSR PC,EXECUTE ;
4668 013734 004737 010636 JSR PC,GOWAIT ;
4669 013740 004737 016060 JSR PC,CKHAE ;
4670 013744 LET ERSFLG := #0
4671 013744 105037 003451 CLR B ERSFLG
4672 013750 ENDIF
4673 013750 50242$:
4674 013750 000207 RTS PC
4675 ; SUBR TO REWRITE A BADLY WRITTEN RECORD
4676
4677 REWRT: IF DEVIBL(R5) EQ #NINUSE THEN ;BTL
4678 013752 026527 002532 177774 CMP DEVIBL(R
4679 013760 001003 BNE 50243$
4680 013762 000137 014100 JMP 1$ ;BTL
4681 013766 ELSE ;BTL
4682 013766 000400 BR 50244$
4683 013770 50243$:
4684 013770 ENDIF ;BTL
4685 013770 50244$:
4686 013770 LET PCMDWD := CMDWRD ;RESTORE WRITE COMMAND PACKET
    
```

SEQ 0113

4687	013770	013737	003346	003352				
4688	013776							
4689	013776	013737	013366	003346	LET CMDWRD := WTYWRD ;		MOV	CMDWRD,P
4690	014004							
4691	014004	013737	013364	002310	LET CMDPKT := WTYCMD ;		MOV	WTYWRD,C
4692	014012							
4693	014012	013737	002310	003350	LET CMDSAV := CMDPKT ;		MOV	WTYCMD,C
4694	014020							
4695	014020	013737	003334	002312	LET CMDPKT.CP.ADL := DATAWT ;		MOV	CMDPKT,C
4696	014026							
4697	014026	013737	013370	002316	LET CMDPKT.CP.CNT := WTYBRF ;		MOV	DATAWT,C
4698	014034							
4699	014034	012737	000002	003354	LET CMDLG := #2 ;		MOV	WTYBRF,C
4700	014042	004737	007344					
4701	014046	004737	010326		JSR PC,CMDAC		MOV	#2,CMDLG
4702	014052				JSR PC,EXECUTE			
4703	014052	026527	002532	177774	IF DEVTBL(R5) EQ #NINUSE THEN ;RE WRITE RECORD			
4704	014060	001003			;			
4705	014062	000137	014100				CMP	DEVTBL(R
4706	014066				JMP 1\$		BNE	50245\$
4707	014066	000400			ELSE			
4708	014070							
4709	014070						BR	50246\$
4710	014070				ENDIF			
4711	014070	004737	010636				50245\$:	
4712	014074	004737	016060		JSR PC,GOWAIT			
4713	014100	000207			JSR PC,CKHAE		50246\$:	
					RTS PC			

1\$:

```

4714 : SUBROUTINE TO LOG BYTES READ/WITTEN.
4715 : ALSO UPDATES READ/WRITE ERROR COUNTERS.
4716 : INPUTS:
4717 : OUTPUTS:
4718 : REGISTERS: R2, R3, R4.
4719 : CALLS:
4720
4721 014102 LOG:: IFB ERLOG EQ #0 THEN ;IF DATA AND ERRORS HAVE NOT BEEN LOGGED
4722 014102 105737 003412 ;IF DATA AND ERRORS HAVE NOT BEEN LOGGED
4723 014106 001126 TSTB ERLOG
4724 014110 LET ERLOG :B= ERLOG * #1 ;SET LOG DONE FLAG. BNE 50247$
4725 014110 105237 003412 INCB ERLOG
4726 014114 LET R4 := CMDLG ;GET CURRENT CMD LOGGING CODE.
4727 014114 013704 003354 MOV CMDLG,R4
4728 014120 IF R4 NE #0 THEN ;IF THERE IS A CODE THEN:
4729 014120 005704 TST R4
4730 014122 001520 BEQ 50250$
4731 014124 LET R4 := R4 #2 ;ADJUST THE CODE FOR TABLE INDEX.
4732 014124 162704 000002 SUB #2,R4
4733 014130 LET R2 := R5 + BINC(R4) + #CNTBGN ;R2 - ADR OF BYTE COUNT LSW.
4734 014130 010502 MOV R5,R2
4735 014132 066402 014366 ADD BINC(R4)
4736 014136 062702 002554 ADD #CNTBGN,
4737 014142 LET (R2) := (R2) + BRFCNT ;ADD BRFCNT TO LSW.
4738 014142 063712 003344 IF MSGPKT*MS.RFC LOS BRFCNT THEN ;IF THE RFC IS LOWER OR THE SAME AS
4739 014146 023737 002340 003344 CMP MSGPKT*M
4740 014146 101002 BHI 50251$
4741 014154 LET (R2) := (R2) - MSGPKT*MS.RFC ;SUBTRACT RFC FROM EXPECTED BRFCNT.
4742 014156 163712 002340 SUB MSGPKT*M
4743 014162 ENDF
4744 014162
4745 014162
4746 014162 LET R3 := R2 * #10 ;R3 = ADR OF 2ND WORD. 50251$:
4747 014162 010203 MOV R2,R3
4748 014164 062703 000010 ADD #10,R3
4749 014170 WHILE (R2) GT #999. DO
4750 014170
4751 014170 021227 001747 50252$: CMP (R2),#99
4752 014174 003404 BLE 50253$
4753 014176 LET (R2) := (R2) - #1000. ;UPDATE BYTE COUNT
4754 014176 162712 001750 SUB #1000,.(
4755 014202 LET (R3) := (R3) * #1 ;2ND WORD.
4756 014202 005213 INC (R3)
4757 014204 ENDDO BR 50252$
4758 014204 000771
4759 014206
4760 014206 LET R2 := R3 * #10 ;R2 = ADR OF 3RD WORD. 50253$:
4761 014206 010302 MOV R3,R2
4762 014210 062702 000010 ADD #10,R2
4763 014214 WHILE (R3) GT #999. DO
4764 014214
4765 014214 021327 001747 50254$: CMP (R3),#99
4766 014220 003404 BLE 50255$
4767 014222 LET (R3) := (R3) - #1000. ;UPDATE BYTE COUNT
4768 014222 162713 001750 SUB #1000,.(
4769 014226 LET (R2) := (R2) * #1 ;3RD WORD.

```

```

4770 014226 005212                                INC      (R2)
4771 014230                                ENDDO
4772 014230 000771                                BR       50254$
4773 014232                                50255$:
4774 014232                                LET R3 := R2 + #10 ;R3 = ADR OF 4TH WORD.
4775 014232 010203                                MOV      R2,R3
4776 014234 062703 000010                        ADD      #10,R3
4777 014240                                WHILE (R2) GT #999. DO
4778 014240                                50256$:
4779 014240 021227 001747                        CMP      (R2),#99
4780 014244 003404                                BLE      50257$
4781 014246                                LET (R2) := (R2) + #1000. ;UPDATE BYTE COUNT
4782 014246 162712 001750                        LET (R3) := (R3) + #1 ;4TH WORD.
4783 014252                                SUB      #1000.,(
4784 014252 005213                                INC      (R3)
4785 014254                                ENDDO
4786 014254 000771                                BR       50256$
4787 014256                                50257$:
4788 014256                                IFB RWERR NE #0 THEN ;IF R/W ERROR, UPDATE ERROR COUNT.
4789 014256 105737 003413                        TSTB    RWERR
4790 014262 001440                                BEQ     50260$
4791 014264                                LET R2 := R5 + EINC(R4) + #WRREC ;R2 = ADR OF COUNTER.
4792 014264 010502                                MOV     R5,R2
4793 014266 066402 014374                        ADD     EINC(R4)
4794 014272 062702 002714                        ADD     #WRREC,R
4795 014276                                IFB UNREC NE #0 THEN ;IS THE ERROR UNRECOVERABLE?
4796 014276 105737 003414                        TSTB    UNREC
4797 014302 001404                                BEQ     50261$
4798 014304                                LET R2 := R2 + #10 ;YES, POINT TO NEXT COUNTER.
4799 014304 062702 000010                        ADD     #10,R2
4800 014310                                LET (R2) := (R2) + #1 ;UPDATE THE ERROR COUNTER
4801 014310 005212                                INC     (R2)
4802 014312                                ELSE ;ELSE IF ERROR IS RECOVERABLE:
4803 014312 000424                                BR      50262$
4804 014314                                50261$:
4805 014314                                LET (R2) := (R2) + #1 ;UPDATE THE ERROR COUNTER
4806 014314 005212                                INC     (R2)
4807 014316                                IFB IREC EQ #0 THEN ;IF ERFOR RECOVERY IS ENABLED:
4808 014316 105737 002211                        TSTB    IREC
4809 014322 001020                                BNE     50263$
4810 014324                                IFB DROPED EQ #0 ANDB ERCVER NE #0 THEN ;IF UNIT HAS NOT BEEN DR
4811 014324 105737 003446                        TSTB    DROPED
4812 014330 001015                                BNE     50264$
4813 014332 105737 002205                        TSTB    ERCVER
4814 014336 001412                                BEQ     50264$
4815 014340                                PRINTB #NURTY1,RETRYC ;PRINT # OF RETRIES TO RECOVER
4816 014340 013746 003404                        MOV     RETRYC,
4817 014344 012746 005073                        MOV     #NURTY1,
4818 014350 012746 000002                        MOV     #2,(SP)
4819 014354 010600                                MOV     SP,R0
4820 014356 104414                                TRAP   C$PNTB
4821 014360 062706 000006                        ADD     #6,SP
4822 014364                                ENDDO
4823 014364                                ENDDO ;PROVIDED PRINT HAS BEEN ENABLED
4824 014364                                50264$:
4825 014364                                50263$:

```



4826 014364  
 4827 014364  
 4828 014364  
 4829 014364  
 4830 014364  
 4831 014364  
 4832 014364  
 4833 014364  
 4834 014364 000207  
 4835  
 4836 014366 000000  
 4837 014370 000040  
 4838 014372 000100  
 4839  
 4840 014374 000000  
 4841 014376 000020  
 4842 014400 000040  
 4843  
 4844

```

      ENDIF
      ENDIF
      ENDIF
      ENDIF
      RTS PC
      INDEXES TO BYTE COUNTERS.
      BINC: 0           ;WRITE.
           40          ;READ REV.
           100         ;READ FWD.
      INDEXES TO READ/WRITE ERROR COUNTERS.
      EINC: 0           ;WRITE.
           20          ;READ REV.
           40          ;READ FWD.
  
```

50262\$:  
 50260\$:  
 50250\$:  
 50247\$:

```

4845
4846
4847
4848
4849
4850
4851
4852
4853
4854 014402          ; IF A WRITE/VERIFY COMMAND IS ISSUED, CONTROL IS THEN
4855 014402 105737 003442 ; TRANSFERRED TO THIS SUBROUTINE TO READ REVERSE, CHECK DATA,
4856 014406 001435      ; READ FORWARD, CHECK DATA, THEN CONTINUE TO NEXT COMMAND.
4857 014410          ; INPUTS:
4858 014410 013737 003346 003352 ; OUTPUTS:
4859 014416          ; REGISTERS:
4860 014416 012737 104401 003346 ; CALLS:          VFEXC.
4861 014424          ;
4862 014424 012737 000004 003354 VFYDAT:: IFB VFYFLG NE #0 THEN ;IF DATA IS TO BE VERIFIED:
4863 014432 004737 014504          ; TSTB VFYFLG
4864 014436          ; BEQ 50265$
4865 014436 026527 002532 177774 ; LET PCMDWD := CMDWRD ;SAVE THE PREVIOUS COMMAND WORD.
4866 014444 001003          ; MOV CMDWRD,P
4867 014446 000137 014502          ; LET CMDWRD :- #RDR ;COMMAND IS READ REV.
4868 014452          ; MOV #RDR,CMD
4869 014452 000400          ; LET CMDLG := #4 ;SET UP CMD LOGGING INDEX.
4870 014454          ; MOV #4,CMDLG
4871 014454          ; JSR PC,VFEXC ;GO READ ALL THE RECORDS REV.
4872 014454          ; IF DEVTBL(R5) EQ #NINUSE THEN ;BTL
4873 014454          ; CMP DEVTBL(R
4874 014454          ; BNE 50266$
4875 014462          ; JMP 1$ ;BTL
4876 014462 012737 104001 003346 ; ELSE ;BTL
4877 014470          ; BR 50267$
4878 014470 012737 000006 003354 ; ENDIF ;BTL
4879 014476 004737 014504          ; LET PCMDWD := CMDWRD ;SAVE THE PREVIOUS COMMAND WORD.
4880 014502          ; MOV CMDWRD,P
4881 014502          ; LET CMDWRD := #RDF ;COMMAND IS READ FWD.
4882 014502 000207          ; LET CMDLG : #6 ;SET UP CMD LOGGING INDEX.
                                ; MOV #6,CMDLG
                                ; JSR PC,VFEXC ;GO READ ALL RECORDS FWD.
                                ; ENDF
                                ; RTS PC ;RETURN.
                                ; 50265$:
                                ; 50266$:
                                ; 50267$:
    
```

```

4883
4884
4885
4886
4887
4888
4889 014504
4890 014504 013737 003346 002310
4891 014512 042737 004000 002310
4892 014520
4893 014520 105737 003444
4894 014524 001403
4895 014526
4896 014526 052737 010000 002310
4897 014534
4898 014534
4899 014534
4900 014534 013737 002310 003350
4901 014542 013737 003336 002312
4902 014550
4903 014550 005037 003340
4904 014554
4905 014554
4906 014554 023737 003340 003342
4907 014562 002071
4908 014564 004737 007344
4909 014570 004737 015452
4910 014574
4911 014574
4912 014574 026527 002532 177777
4913 014602 001442
4914 014604
4915 014604 032737 000400 003346
4916 014612 001421
4917 014614
4918 014614 032765 000002 003426
4919 014622 001014
4920 014624
4921 014624 032765 000001 003426
4922 014632 001406
4923 014634
4924 014634 105737 003450
4925 014640 001402
4926 014642 004737 014750
4927 014646
4928 014646
4929 014646
4930 014646 000402
4931 014650
4932 014650 004737 014750
4933 014654
4934 014654
4935 014654
4936 014654
4937 014654
4938 014654 000412

```

```

SUBROUTINE TO EXECUTE THE READ AND VERIFY, FORWARD OR REVERSE.
INPUTS:
OUTPUTS:
REGISTERS: R2
CALLS: CMDAC, FIRSTU, VFISU, NEXTU, CKMAE.

VFEXC:: LET CNDPKT := CMDWRD CLR BY @BRF.C ;COMMAND PACKET = READ REV OR FWD.
MOV CMDWRD.C
BIC @BRF.C.C
IFB SWBFLG NE #0 THEN ;IF BITES ARE TO BE SWAPPED:
TSTB SWBFLG
BEQ 502708
LET CNDPKT := CNDPKT SET BY @SWB.C ;SET SWAB BIT IN CMD PACKET.
BIS @SWB.C.C
ENDIF
LET CNDPCKT := CNDPKT ;SAVE COMMAND PACKET 1ST WORD.
MOV CNDPKT.C
MOV DATARD,CNDPKT.CP.ADL ;SAVE BUFFER START ADDRESS.
LET NCNT := #0 ;CLEAR NUMBER OF OPERATIONS.
CLR NCNT
WHILE NCNT LT NCNT1 DO ;WHILE THERE ARE RECORDS REMAINING:
502718:
CMP NCNT,NCNT1
BGE 502728
JSR PC,CMDAC ;STORE CMD ASCII IN ERROR MSG.
JSR PC,FIRSTU ;SET UP FOR FIRST UNIT.
WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE DEVICES REMAINING:
502738:
CMP DEVTBL(CP)
BEQ 502748
IF @MOD.CO SET IN CMDWRD THEN ;IF CMD IS REVERSE THEN:
BIT @MOD.CO
BEQ 502758
IF @XO.BOT NOT SET IN EOTFLG(R5) THEN ;IF NOT AT BOT
BIT @XO.BOT
BNE 502768
IF @XO.EOT SET IN EOTFLG(R5) THEN ;BUT IF AT EOT
BIT @XO.EOT
BEQ 502778
IFB ALLEOT NE #0 THEN ;AND ALL OTHERS AT EOT
TSTB ALLEOT
BEQ 503008
JSR PC,VFISU ;THEN READ VERIFY
ENDIF ;IF NOT ALL AT EOT. FREEZE
503008:
ELSE ;IF NOT AT BOT AND
BR 503018
502778:
;NOT AT EOT. READ VFY
503018:
ENDIF
502768:
ELSE IF CMD IS NOT REVERSE:
BR 503008

```



```

4974      ; SUBROUTINE TO ISSUE COMMAND, AWAIT INTERRUPT,
4975      ; CHECK STATUS, CHECK DATA.
4976      ; INPUTS:
4977      ; OUTPUTS:
4978      ; REGISTERS:      R2
4979      ; CALLS:          EXECUTE, GOWAIT, CKDATA.
4980
4981 014750 VFISU::      LET R2 := DATARD * #8.      ;INIT READ BUFFER POINTER.
4982 014750 013702 003336      ;MOV DATARD,R
4983 011754 062702 000010      ;ADD #8.,R2
4984 014760      WHILE R2 NE DATARD DO      ;UNTIL 8 BYTES HAVE BEEN SET,
4985 014760      ;50307$:
4986 014760 020237 003336      ;CMP R2,DATARD
4987 014764 001403      ;BEQ 50310$
4988 014766      LET (R2) := # 1      ;INIT READ BUFFER.
4989 014766 012742 177777      ;MOV # 1, (R2)
4990 014772      ENDDO
4991 014772 000772      ;BR 50307$
4992 014774      ;50310$:
4993 014774 004737 010326      JSR PC,EXECUTE      ;GO EXECUTE THE COMMAND.
4994 015000      IFB DROPED EQ #0 THEN      ;IF UNIT HAS NOT BEEN DROPPED THEN:
4995 015000 105737 003446      ;TSTB DROPED
4996 015004 001002      ;BNE 50311$
4997 015006 004737 010636      JSR PC,GOWAIT      ;GO WAIT FOR DONE BIT.
4998 015012      ENDIF
4999 015012      ;50311$:
5000 015012      IFB DROPED EQ #0 THEN      ;IF UNIT HAS NOT BEEN DROPPED THEN:
5001 015012 105737 003446      ;TSTB DROPED
5002 015016 001006      ;BNE 50312$
5003 015020      IF #X0.BOT NOTSET IN EOTFLG(R5) THEN      ;WHEN NOT REVERSED INTO B
5004 015020 032765 000002 003426      ;BIT #X0.BOT.
5005 015026 001002      ;BNE 50313$
5006 015030 004737 015036      JSR PC,CKDATA      ;GO VERIFY DATA.
5007 015034      ENDIF
5008 015034      ;50313$:
5009 015034      ENDIF
5010 015034      ;50312$:
5011 015034 000207      RTS PC
5012

```

```

5013 ; SUBROUTINE TO COMPARE DATA BETWEEN READ AND WRITE BUFFERS
5014 ; AND PRINT ERROR MESSAGE ON MISCOMPARE.
5015 ; INPUTS:
5016 ; OUTPUTS:
5017 ; REGISTERS: R2, R3, R4.
5018 ; CALLS: GCMDB
5019
5020 CKDATA:: LET R3 := BRFCNT MSGPKT*MS.RFC ; COMPUTE REC LENGTH READ
5021 015036 013703 003344 MOV BRFCNT,R
5022 015042 163703 002340 SUB MSGPKT,M
5023 015046 IF R3 EQ #0 THEN ; WHEN NO DATA RECEIVED
5024 015046 005703 TST R3
5025 015050 001015 BNE 50314$
5026 015052 ERRHRD 17,WTVERM,DTAERM ; PRINT ERROR AND EXIT
5027 015052 104456 TRAP C$ERRHRD
5028 015054 000021 .WORD 17
5029 015056 004164 .WORD WTVERM
5030 015060 005224 .WORD DTAERM
5031 015062 PRINTB #DTAER4 ; COMPARE ROUTINE
5032 015062 012746 005010 MOV #DTAER4,
5033 015066 012746 000001 MOV #1,(SP)
5034 015072 010600 MOV SP,R0
5035 015074 104414 TRAP C$PNTB
5036 015076 062706 000004 ADD #4,SP
5037 015102 ELSE
5038 015102 000560 BR 50315$
5039 015104
5040 015104 IF R3 HI BRFCNT THEN ; WHEN REC READ IS LONGER
5041 015104 020337 003344 CMP R3,BRFCN
5042 015110 101417 BLOS 50316$
5043 015112 ERRHRD 17,WTVERM,DTAERM ; THAN EXPECTED, PRINT
5044 015112 104456 TRAP C$ERRHRD
5045 015114 000021 .WORD 17
5046 015116 004164 .WORD WTVERM
5047 015120 005224 .WORD DTAERM
5048 015122 PRINTB #DTAERS,CMDPKT*CP.CNT ; AN ERROR MESSAGE
5049 015122 013746 002316 MOV CMDPKT,C
5050 015126 012746 005031 MOV #DTAERS,
5051 015132 012746 000002 MOV #2,(SP)
5052 015136 010600 MOV SP,R0
5053 015140 104414 TRAP C$PNTB
5054 015142 062706 000006 ADD #6,SP
5055 015146 ELSE ; AND EXIT ROUTINE
5056 015146 000536 BR 50317$
5057 015150
5058 015150 LET CKDCNT := R3 #1 ; SAVE VERIFICATION LENGTH 1.
5059 015150 010337 015446 MOV R3,CKDCN
5060 015154 005337 015446 DEC CKDCNT
5061 015160 005037 015450 CLR CKOFF ; CLEAR # OF BYTES IN ERROR COUNTER.
5062 015164 005002 CLR R2 ; INIT BYTE COUNTER
5063 015166 LET R3 := DATAWT ; GET WRITE BUFFER ADDRESS.
5064 015166 013703 003334 MOV DATAWT,R
5065 015172 LET R4 := DATARD ; GET READ BUFFER ADDRESS.
5066 015172 013704 003336 MOV DATARD,R
5067 015176 IFB T1SWB NE #0 THEN ; WHEN RUNNING TEST1 SUB
5068 015176 105737 003447 TSTB T1SWB
    
```

```

5069 015202 001401
5070 015204 000313
5071 015206
5072 015206
5073 015206
5074 015206
5075 015206
5076 015206 020237 015446
5077 015212 001011
5078 015214
5079 015214 105737 003444
5080 015220 001406
5081 015222
5082 015222 032737 000001 015446
5083 015230 001002
5084 015232 105723
5085 015234 105724
5086 015236
5087 015236
5088 015236
5089 015236
5090 015236
5091 015236
5092 015236 121314
5093 015240 001452
5094 015242 005737 015450
5095 015246 001010
5096 015250 005265 003274
5097 015254 005265 003304
5098 015260
5099 015260 104456
5100 015262 000021
5101 015264 004164
5102 015266 005224
5103 015270
5104 015270 005237 015450
5105 015274 111437 003364
5106 015300 042737 177400 003364
5107 015306 111337 003366
5108 015312 042737 177400 003366
5109 015320
5110 015320 023727 015450 000013
5111 015326 002017
5112 015330
5113 015330 005046
5114 015332 153716 003366
5115 015336 005046
5116 015340 153716 003364
5117 015344 010246
5118 015346 012746 004677
5119 015352 012746 000004
5120 015356 010600
5121 015360 104415
5122 015362 062706 000012
5123 015366
5124 015366

SWAB (R3) ;SWAP FIRST WORD OF WRT BFR
ENDIF ;WHICH CONTAINS THE RECORD COUNT
REPEAT ;REPEAT UNTIL ALL DATA IS COMPARED:
IF R2 EQ CKDCNT THEN ;IF THIS IS THE LAST BYTE THEN:
    CMP R2,CKDCN
    BNE 50322$
    IFB SWBFLG NE #0 THEN ;IF BYTE SWAPPING IS ENABLED THEN:
        TSTB SWBFLG
        BEQ 50323$
        IF #BIT00 NOTSETIN CKDCNT THEN ;IF RECORD LENGTH IS ODD
            BIT #BIT00,C
            BNE 50324$
            TSTB (R3). ;LAST BYTE WILL BE IN
            TSTB (R4). ;THE UPPER BYTE.
        ENDIF
    ENDIF
ENDIF
CMPB (R3),(R4) ;ARE THEY EQUAL.
BEQ 3$ ;BR IF SO.
TST CKDFF ;1 ST TIME THRU?
BNE 2$ ;BR IF NOT.
INC VFYCNT(R5) ;INC THE VERIFY ERROR COUNTER.
INC HRDCNT(R5) ;INC THE HARD ERROR COUNT.
ERRHRD #17,WTVERM,DTAERM ;REPORT WRITE/VERIFY ERROR.
TRAP C$ERRRD
.WORD 17
.WORD WTVERM
.WORD DTAERM
2$: LET CKDFF := CKDFF + #1 ;INCREMENT # OF BYTES IN ERROR.
INC CKDFF
MOVW (R4),TIME1 ;SAVE WAS DATA FOR TYP0UT.
BIC #177400,TIME1 ;CLEAR GARBAGE.
MOVW (R3),TIME2 ;SAVE SHOULD BE DATA FOR TYP0UT.
BIC #177400,TIME2 ;CLEAR GARBAGE.
IF CKDFF LT #11. THEN ;IF ERROR BYTE COUNT IS LESS THAN 11:
    PRINTX #DTAER2,R2,<B,TIME1>,<B,TIME2> ;PRINT EXP - ACT DATA.
    CMP CKDFF,#1
    BGE 50325$
    CLR -(SP)
    BISB TIME2,(S
    CLR -(SP)
    BISB TIME1,(S
    MOV R2,(SP)
    MOV #DTAER2,
    MOV #4, -(SP)
    MOV SP,R0
    TRAP C$PNTX
    ADD #12,SP
ENDIF
50325$:

```

# GLO

GLOBAL AREAS MACY11 30(1046) 06 APR 84 08:49  
 CZTSMO.P11

06 APR 84 08:51 PAGE 125  
 GLOBAL SUBROUTINES SECTION

SEQ 0123

```

5125 015366 105723          3$:          TSTB (R3).          ;UPDATE WRITE BUFFER ADDRESS.
5126 015370 105724          TSTB (R4).          ;UPDATE READ BUFFER ADDRESS.
5127 015372 105722          TSTB (R2).          ;UPDATE BYTE COUNTER.
5128 015374          UNTIL R2 GT CKDCNT          ;END OF DATA COMPARE REPEAT LOOP.
5129 015374 020237 015446          CMP          R2,CKDCN
5130 015400 003702          BLE          50321$
5131 015402          LET CKDCNT := CKDCNT + #1          ;CKDCNT EQUALS RECORD LENGTH.
5132 015402 005237 015446          IF CKDFF NE #0 THEN          ;IF COMPARE ERROR HAS OCCURED THEN:
5133 015406          PRINTB #DTAER3,CKDFF,CKDCNT          ;PRINT # OF BYTES IN ERROR.
5134 015406 005737 015450          TST          CKDFF
5135 015412 001414          BEQ          50326$
5136 015414          MOV          CKDCNT,
5137 015414 013746 015446          MOV          CKDFF,-(
5138 015420 013746 015450          MOV          #DTAER3,
5139 015424 012746 004746          MOV          #3,(SP)
5140 015430 012746 000003          MOV          SP,R0
5141 015434 010600          TRAP          C$PNTB
5142 015436 104414          ADD          #10,SP
5143 015440 062706 000010          ENDIF
5144 015444          ENDIF
5145 015444          ENDIF
5146 015444          ENDIF
5147 015444          RTS          PC
5148 015444          ;OTHERWISE, RETURN.
5149 015444          ;# OF BYTES TO BE VERIFIED 1.
5150 015444 000207          ;# OF BYTES IN ERROR COUNTER.
5151
5152 015446 000000          CKDCNT: .WORD 0
5153 015450 000000          CKDFF: .WORD 0
  
```



```

5154      ;      SUBROUTINE TO FIND THE FIRST DEVICE IN THE TEST SEQUENCE.
5155      ;      INPUTS:
5156      ;      OUTPUTS:
5157      ;      REGISTERS:
5158      ;      CALLS:
5159
5160 015452      FIRSTU:: LET DROPE :B= #0      ;CLR UNIT DROPPED FLAG
5161 015452 105037 003446      ;CLR DEVICE PC  "FR.      CLR      DROPE
5162 015456      LET R5 := #0      ;CLR DEVICE PC  "FR.      CLR      R5
5163 015456 005005      WHILE DEVTBL(R5) EQ #NINUSE DO ;WHILE DEVICES ARE NOT IN USE:
5164 015460      ;      50327$:
5165 015460      ;      CMP      DEVTBL(R
5166 015460 026527 002532 177774      ;      BNE      50330$
5167 015466 001003      LET R5 := R5 + #2      ;POINT TO NEXT DEVICE.
5168 015470      ;      ADD      #2,R5
5169 015470 062705 000002      ENDDO
5170 015474      ;      BR      50327$
5171 015474 000771      ;      50330$:
5172 015476      IF DEVTBL(R5) EQ #END THEN ;IF ALL UNITS HAVE BEEN DROPPED THEN:
5173 015476      ;      CMP      DEVTBL(R
5174 015476 026527 002532 177777      ;      BNE      50331$
5175 015504 001001      ;      DOCLN      ;DO CLEAN CODE AND TERMINATE PASS.
5176 015506      ;      TRAP      C$DOCLN
5177 015506 104444      ENDF
5178 015510      ;      50331$:
5179 015510      LET L$LUN := DEVTBL(R5) ;SET UNIT # IN "HEADER" FOR ERROR REPORT
5180 015510      ;      MOV      DEVTBL(R
5181 015510 016537 002532 002074      RTS      PC      ;RETURN WITH 1ST DEVICE IN R5.
5182 015516 000207
5183
5184
5185
5186
5187
5188      ;      SUBROUTINE TO FIND THE NEXT UNIT IN THE TEST CYCLE.
5189      ;      INPUTS:
5190      ;      OUTPUTS:
5191      ;      REGISTERS:
5192      ;      CALLS:
5193
5194 015520      NEXTU:: LET DROPE :B= #0      ;CLR UNIT DROPPED FLAG
5195 015520 105037 003446      ;CLR UNIT DROPPED FLAG      CLR      DROPE
5196 015524 042705 177770      BIC      #177770,R5      ;BTL
5197 015530      REPEAT      ;REPEAT UNTIL THE NEXT DEVICE IS FOUND.
5198 015530      ;      50332$:
5199 015530      LET R5 := R5 + #2      ;UPDATE DEVICE TABLE POINTER.
5200 015530 062705 000002      ;      ADD      #2,R5
5201 015534      UNTIL DEVTBL(R5) NE #NINUSE
5202 015534 026527 002532 177774      ;      CMP      DEVTBL(R
5203 015542 001772      ;      BEQ      50332$
5204 015544      LET L$LUN := DEVTBL(R5) ;SET UNIT # IN HEADER FOR ERROR REPORT
5205 015544 016537 002532 002074      ;      MOV      DEVTBL(R
5206 015552 000207      RTS      PC      ;RETURN.
5207
5208
5209

```

```

5210 ; SUBROUTINE TO DROP A DEVICE FROM THE TEST SEQUENCE.
5211 ;
5212 ; INPUTS:
5213 ; OUTPUTS:
5214 ; REGISTERS:
5215 ; CALLS: MOVMSG, PRXST, LOG
5216 015554 DROPU:: LET R5 := R5SAVE ;BTL
5217 015554 013705 003400 LET FTLCNT(R5) := FTLCNT(R5) + #1 ;INCREMENT THE FATAL ERROR COUNT.
5218 015560 MOV R5SAVE,R
5219 015560 005265 003314 INC FTLCNT(R
5220 015564 LET R4 := MSGPKT.M5.XS3 CLR.BY #377 ;GET UDIAG ERROR CODE FROM XSTAT3.
5221 015564 013704 002350 MOV MSGPKT.M
5222 015570 042704 000377 BIC #377,R4
5223 015574 LET R3 := MSGPKA(R5) ;ADR OF THIS UNIT'S MSG PACKET.
5224 015574 016503 002502 MOV MSGPKA(R
5225 015600 LET R2 := #0 ;CLR COUNTER.
5226 015600 005002 CLR R2
5227 015602 WHILE R2 NE #MSGCNT DO ;WHILE THERE ARE MORE LOCATIONS:
5228 015602 50333$:
5229 015602 020227 000016 CMP R2,#MSGC
5230 015606 001405 BEQ 50334$
5231 015610 LET (R3)+ := #1 ;INIT THE MSG PACKET WITH ALL 1'S
5232 015610 012723 177777 MCV #-1,(R3)
5233 015614 LET R2 := R2 + #2 ;UPDATE COUNTER.
5234 015614 062702 000002 ADD #2,R2
5235 015620 ENDDO BR 50333$
5236 015620 000770 50334$:
5237 015622 LET @TSD8(R5) := #GSCPK ;INITIATE A GET STATUS COMMAND.
5238 015622 MOV #GSCPK,@
5239 015622 012775 002320 002452 JSR PC,WSSR ;WAIT A WHILE FOR SSR=1
5240 015630 004737 011170 JSR PC,MOVMSG ;MOVE MSG PACKET TO COMMON AREA.
5241 015634 004737 011224 IF R4 EQ #X3.RNY THEN ;IF WE HAVE A CAPSTAN RUNAWAY THEN:
5242 015640 020427 157400 CMP R4,#X3.R
5243 015640 001005 BNE 50335$
5244 015644 ERRDF #16,RNYM,STAERM ;REPORT CAPSTAN RUNAWAY WITH TACH CNT.
5245 015646 TRAP C$ERDF
5246 015646 104455 .WORD 16
5247 015650 000020 .WORD RNYM
5248 015652 004504 .WORD STAERM
5249 015654 005372 ELSE ;ELSE IF NOT A RUNAWAY:
5250 015656 BR 50336$
5251 015656 000402 50335$:
5252 015660 JSR PC,PRXST ;PRINT EXTENDED STATUS REGISTERS.
5253 015660 004737 015776 ENDF
5254 015664 50336$:
5255 015664 IF B RECLOG NE #0 THEN ;IF THE RECORD HAS BEEN LOGGED THEN:
5256 015664 TSTB RECLOG
5257 015664 105737 003411 BEQ 50337$
5258 015670 001404 LET DROPED := B+ DROPED + #1 ;SET UNIT DROPPED FLAG.
5259 015672 105237 003446 JSR PC,LOG INCB DROPED
5260 015672 004737 014102 ENDF ;LOG DATA BYTES + RD WR ERRORS.
5261 015676 014102 DORPT 50337$:
5262 015702 ;PRINT PERFORMANCE REPORT
5263 015702 TRAP C$DORPT
5264 015702 104424
5265 015702

```

```

5266 015704          DROPUA: IF PASCNT(R5) NE #0 THEN
5267 015704 005765 003254          TST      PASCNT(R
5268 015710 001402          BEQ      50340$
5269 015712          LET PASCNT(R5) := PASCNT(R5) #1
5270 015712 005365 003254          DEC      PASCNT(R
5271 015716          ENDIF
5272 015716          50340$:
5273 015716          LET DROPN := DEVTBL(R5) ;SAVE # OF UNIT TO BE DROPPED.
5274 015716 016537 002532 015774          MOV      DEVTBL(R
5275 015724          LET RO := R5 SHIFT 1 ;RO=LOGICAL DEVICE NUMBER
5276 015724 010500          MOV      R5,RO
5277 015726 006200          ASR      RO
5278 015730          DODU RO ;DROP THE UNIT: EXEC BGNDU ENDDU CODE IF IDU = 0
5279 015730 104451          TRAP     C$DODU
5280 015732          IF DEVTBL(R5) NE #NINUSE THEN ;IF UNIT NOT DROPPED
5281 015732 026527 002532 177774          CMP      DEVTBL(R
5282 015740 001410          BEQ      50341$
5283 015742          IFB IREC EQ #0 THEN ;IF RECOVERY IS ENABLED THEN:
5284 015742 105737 002211          TSTB     IREC
5285 015746 001005          BNE      50342$
5286 015750 000240          NOP
5287 015752 000240          NOP
5288 015754 000240          NOP
5289 015756          LET STAFLG :B= STAFLG * #1 ;SET START FLAG TO ENABLE REWIND.
5290 015756 105237 003452          INCB     STAFLG
5291 015762          ENDIF
5292 015762          50342$:
5293 015762          ENDIF
5294 015762          50341$:
5295 015762          DRORTN: LET DROPED :B= DROPED * #1 ;SET UNIT DROPPED FLAG.
5296 015762 105237 003446          INCB     DROPED
5297 015766          LET R5 := R5SAVE ;BTL
5298 015766 013705 003400          MOV      R5SAVE,R
5299 015772 000207          RTS      PC ;RETURN.
5300
5301 015774 000000          DROPN: .WORD 0 ;# OF UNIT TO BE DROPPED
    
```

```

5302      ; SUBROUTINE TO PRINT EXTENDED STATUS REGISTERS.
5303      ; INPUTS:
5304      ; OUTPUTS:
5305      ; REGISTERS:
5306      ; CALLS:
5307
5308      PRXST:: PRINTX #GETSTM
5309      015776 012746 005157      MOV      #GETSTM,
5310      016002 012746 000001      MOV      #1, (SP)
5311      016006 010600      MOV      SP,R0
5312      016010 104415      TRAP    C$PNTX
5313      016012 062706 000004      ADD     #4,SP
5314      016016      PRINTX #STAERS,MSGPKT*MS. ,MSGPKT*MS.XS1,MSGPKT*MS.XS2,MSGPKT*MS.XS3
5315      016016 013746 002350      MOV     MSGPKT.M
5316      016022 013746 002346      MOV     MSGPKT.M
5317      016026 013746 002344      MOV     MSGPKT.M
5318      016032 013746 002342      MOV     MSGPKT.M
5319      016036 012746 006217      MOV     #STAERS,
5320      016042 012746 000005      MOV     #5, -(SP)
5321      016046 010600      MOV     SP,R0
5322      016050 104415      TRAP    C$PNTX
5323      016052 062706 000014      ADD     #14,SP
5324      016056 000207      RTS    PC
5325
5326
5327
5328
5329      ; SUBROUTINE TO HALT AFTER EACH COMMAND.
5330      ; INPUTS:
5331      ; OUTPUTS:
5332      ; REGISTERS:      R3, R4
5333      ; CALLS:
5334
5335      CKHAE:: IFB HAE NE #0 THEN      ;IF HALT FLAG IS SET:
5336      016060 105737 002204      TSTB   HAE
5337      016064 001430      BEQ    50343$
5338      016066      IFB MISCFG EQ #0 THEN      ;
5339      016066 105737 003455      TSTB   MISCFG
5340      016072 001023      BNE    50344$
5341      016074      MANUAL      ;IS MANUAL INTERVENTION ALLOWED?
5342      016074 104450      TRAP   C$MANI
5343      016076      BNCOMPLETE CKHRTN      ;BR IF NOT.
5344      016076 103023      BCC    CKHRTN
5345      016100      LET R4 := CMDWRD      ;COMMAND WORD.
5346      016100 013704 003346      MOV    CMDWRD,R
5347      016104 004737 007416      JSR   PC,GCMDA      ;FETCH ADR OF CMD ASCII.
5348      016110      LET HALTM :B= (R3).      ;MOVE CMD ASCII
5349      016110 112337 004042      MOVB   (R3),.HA
5350      016114      LET HALTM+1 :B= (R3).
5351      016114 112337 004043      MOVB   (R3),.HA
5352      016120      LET HALTM+2 :B= (R3)      ;INTO MESSAGE.
5353      016120 111337 004044      MOVB   (R3),HAL
5354      016124      GMANIL HAL'1,TIME1,1,YES      ;HALT WAIT FOR AN OEPRTOR INPLY.
5355      016124 104443      TRAP   C$GMAN
5356      016126 000404      BR     10000$
5357      016130 003364      .WORD TIME1
    
```

5358 016132 000130  
 5359 016134 004042  
 5360 016136 000001  
 5361 016140  
 5362 016140  
 5363 016140 000402  
 5364 016142  
 5365 016142  
 5366 016142 105037 003455  
 5367 016146  
 5368 016146  
 5369 016146  
 5370 016146  
 5371 016146 000207  
 5372  
 5373  
 5374 016150

10000\$:

ELSE

LET MISCFG :B= #0

ENDIF

ENDIF

CKHRTN: RTS PC

.EVEN

ENDMOD

.WORD T\$CODE  
 .WORD HALTM  
 .WORD 1

50344\$: BR 50345\$

CLR B MISCFG

50345\$:

50343\$:

;RETURN

```

5375
5376 .TITLE MISCELLANEOUS SECTIONS
5377 .SBTTL REPORT CODING SECTION
5378
5379 016150 BGNMOD
5380
5381 ;**
5382 ; THE REPORT CODING SECTION CONTAINS THE
5383 ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
5384 ; -
5385
5386 016150 BGNRPT
5387 016150 L$RPT::
5388
5389
5390 016150 LET R$SAVE := R5 ;SAVE CURRENT DEVICE POINTER.
5391 016150 010537 003400 MOV R5,R$SAVE
5392 016154 004737 015452 JSR PC,FIRSTU ;FIND THE FIRST UNIT.
5393 016160 WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
5394 016160 50346$:
5395 016160 026527 002532 177777 CMP DEVTBL(R
5396 016166 001562 BEQ 50347$
5397 016170 PRINTS #RPT1A,DEVTBL(R5),PASCNT(R5),RECCNT(R5)
5398 016170 016546 003324 MOV RECCNT(R
5399 016174 016546 003254 MOV PASCNT(R
5400 016200 016546 002532 MOV DEVTBL(R
5401 016204 012746 017012 MOV #RPT1A,
5402 016210 012746 000004 MOV #4,-(SP)
5403 016214 010600 MOV SP,R0
5404 016216 104416 TRAP C$PNTS
5405 016220 062706 000012 ADD #12,SP
5406 016224 PRINTS #RPT1B,WRBC+30(R5),WRBC+20(R5),WRBC+10(R5),WRBC(R5)
5407 016224 016546 002554 MOV WRBC(R5)
5408 016230 016546 002564 MOV WRBC+10(
5409 016234 016546 002574 MOV WRBC+20(
5410 016240 016546 002604 MOV WRBC+30(
5411 016244 012746 017067 MOV #RPT1B,-
5412 016250 012746 000005 MOV #5,-(SP)
5413 016254 010600 MOV SP,R0
5414 016256 104416 TRAP C$PNTS
5415 016260 062706 000014 ADD #14,SP
5416 016264 PRINTS #RPT1C,RRBC+30(R5),RRBC+20(R5),RRBC+10(R5),RRBC(R5)
5417 016264 016546 002614 MOV RRBC(R5)
5418 016270 016546 002624 MOV RRBC+10(
5419 016274 016546 002634 MOV RRBC+20(
5420 016300 016546 002644 MOV RRBC+30(
5421 016304 012746 017140 MOV #RPT1C,-
5422 016310 012746 000005 MOV #5,-(SP)
5423 016314 010600 MOV SP,R0
5424 016316 104416 TRAP C$PNTS
5425 016320 062706 000014 ADD #14,SP
5426 016324 PRINTS #RPT1D,RFBC+30(R5),RFBC+20(R5),RFBC+10(R5),RFBC(R5)
5427 016324 016546 002654 MOV RFBC(R5)
5428 016330 016546 002664 MOV RFBC+10(
5429 016334 016546 002674 MOV RFBC+20(
5430 016340 016546 002704 MOV RFBC+30(

```

```

5431 016344 012746 017211      MOV      #RPT1D,-
5432 016350 012746 000005      MOV      #5,-(SP)
5433 016354 010600                MOV      SP,R0
5434 016356 104416                TRAP     C$PNTS
5435 016360 062706 000014                ADD      #14,SP
5436 016364                PRINTS   #RPT1F,WRREC(R5),RRREC(R5),RFREC(R5)
5437 016364 016546 002754                MOV      RFREC(R5)
5438 016370 016546 002734                MOV      RRREC(R5)
5439 016374 016546 002714                MOV      WRREC(R5)
5440 016400 012746 017315                MOV      #RPT1F,
5441 016404 012746 000004                MOV      #4,-(SP)
5442 016410 010600                MOV      SP,R0
5443 016412 104416                TRAP     C$PNTS
5444 016414 062706 000012                ADD      #12,SP
5445 016420                PRINTS   #RPT1G,WRUNR(R5),RRUNR(R5),RFUNR(R5)
5446 016420 016546 002764                MOV      RFUNR(R5)
5447 016424 016546 002744                MOV      RRUNR(R5)
5448 016430 016546 002724                MOV      WRUNR(R5)
5449 016434 012746 017366                MOV      #RPT1G,
5450 016440 012746 000004                MOV      #4,-(SP)
5451 016444 010600                MOV      SP,R0
5452 016446 104416                TRAP     C$PNTS
5453 016450 062706 000012                ADD      #12,SP
5454 016454                IFB     BADTSW NE #0 THEN ;
5455 016454 105737 002206                TSTB    BADTSW
5456 016460 001402                BEQ     50350$
5457 016462 004737 016544                JSR     PC,BTRPT ;GO PRINT BAD TAPE SPOTS WHEN ENABLED
5458 016466                ENDF
5459 016466                PRINTS   #RPT1I,SCCNT(R5),HRDCNT(R5),FTLCNT(R5),VFYCNTR(R5)
5460 016466                50350$:
5461 016466 016546 003274                MOV      VFYCNTR(R5)
5462 016472 016546 003314                MOV      FTLCNT(R5)
5463 016476 016546 003304                MOV      HRDCNT(R5)
5464 016502 016546 003264                MOV      SCCNT(R5)
5465 016506 012746 017563                MOV      #RPT1I,-
5466 016512 012746 000005                MOV      #5,-(SP)
5467 016516 010600                MOV      SP,R0
5468 016520 104416                TRAP     C$PNTS
5469 016522 062706 000014                ADD      #14,SP
5470 016526 004737 015520                JSR     PC,NEXTU ;FIND THE NEXT UNIT.
5471 016532                ENDDO
5472 016532 000612                BR      50346$
5473 016534                50347$:
5474 016534                LET     R5 : R$SAVE ;RESTORE CURRENT DEVICE POINTER.
5475 016534 013705 003400                MOV      R$SAVE,R
5476 016540                EXIT    RPT
5477 016540 000157                .WORD   J$JMP
5478 016542 001130                .WORD   L10010 2
5479
5480
5481
5482
5483 ;
5484 ;
5485 ;
5486 ;
SUBR TO PRINT BAD TAPES SPOTS DURING THE REPORT PRINTS
WRITE RETRIES: CUMULATIVE COUNT
BAD TAPE SPOTS: COUNT PER TAPE PASS ONLY, NOT CUMULATIVE.
COUNT OF RECOVERABLE WRITE ERRORS EXCLUDES BAD TAPE SPOTS.

```

```

5487
5488
5489 016544          BTRPT: PRINTS @RPT1E,WRTYCT(R5)          ;PRINT GLOBAL WRITE RETRY COUNT
5490 016544 016546 003244          MOV WRTYCT(R
5491 016550 012746 017437          MOV @RPT1E,
5492 016554 012746 000002          MOV @2.(SP)
5493 016560 010600          MOV SP,R0
5494 016562 104416          TRAP C$PNTS
5495 016564 062706 000006          ADD @6.SP
5.96 016570          LET BTPT := BTADDR(R5) ;BTPT IS BOTH THE BAD TAPE SPOT COUNTER
5497 016570 016537 002544 003436          MOV BTADDR'R
5498 016576          LET R3 := @BTPT SHIFT 1          ;AND THE LUGGING INDEX
5499 016576 017703 164634          MOV @BTPT,R3
5500 016602 006203          ASR R3
5501 016604          PRINTS @RPT1J,R3          ;PRINT # OF BAD TAPE SPOTS
5502 016604 010346          MOV R3.(SP)
5503 016606 012746 017467          MOV @RPT1J,
5504 016612 012746 000002          MOV @2.(SP)
5505 016616 010600          MOV SP,R0
5506 016620 104416          TRAP C$PNTS
5507 016622 062706 000006          ADD @6.SP
5508 016626          IF R3 NE #0 THEN          ;PRINT RECORD # IF BAD SPOTS DETECTED
5509 016626 005703          TST R3
5510 016630 001457          BEQ 50351$
5511 016632          IF R3 HI #20. THEN          ;
5512 016632 020327 000024          CMP R3,#20.
5513 016636 101402          BLOS 50352$
5514 016640          LET R3 := #20.          ;20 BAD SPOTS IS THE LIMIT
5515 016640 012703 000024          MOV #20.,R3
5516 016.44          ENDIF
5517 016644          PRINTS @CRLF SP          ; 50352$:
5518 016644          MOV @CRLF SP,
5519 016644 012746 005216          MOV #1.(SP)
5520 016650 012746 000001          MOV SP,R0
5521 016654 010600          TRAP C$PNTS
5522 016656 104416          ADD @4.SP
5523 016660 062706 000004          LET R4 := BTPT * #2          ;FETCH A BAD SPOT ID
5524 016664          MOV BTPT,R4
5525 016664 013704 003436          ADD #2,R4
5526 016670 062704 000002          LET R2 := #0          ;R2 = PRINT COUNT PER LINE: 10 MAX
5527 016674          REPEAT          ;
5528 016674 005002          PRINTS @RPT1K,(R4)          ;PRINT A BAD SPOT ID          50353$:
5529 016676          MOV (R4).(S
5530 016676          MOV @RPT1K,
5531 016676          MOV @2.(SP)
5532 016676 011446          MOV SP,R0
5533 016700 012746 017554          TRAP C$PNTS
5534 016704 012746 000002          ADD @6.SP
5535 016710 010600          LET R2 := R2 * #1          ;COUNT PRINTS
5536 016712 104416          LET R4 := R4 * #2          ;NEXT
5537 016714 062706 000006          INC R2
5538 016720          LET R4 := R4 * #2          ;
5539 016720 005202          IF R2 EQ #10. THEN          ;
5540 016722          ADD #2,R4
5541 016722 062704 000002
5542 016726

```



5543	016726	020227	000012						
5544	016732	001014							
5545	016734								
5546	016734	012746	005216						
5547	016740	012746	000001						
5548	016744	010600							
5549	016746	104416							
5550	016750	062706	000004						
5551	016754			LET R3 := R3	010.				
5552	016754	162703	000012						
5553	016760			LET R2 := R2	010.				
5554	016760	162702	000012						
5555	016764			ENDIF					
5556	016764								
5557	016764			UNTIL R2 EQ R3					
5558	016764	020203							
5559	016766	001343							
5560	016770			ENDIF					
5561	016770								
5562	016770			PRINTS	0CRLF				
5563	016770	012746	005213						
5564	016774	012746	000001						
5565	017000	010600							
5566	017002	104416							
5567	017004	062706	000004						
5568	017010	000207							
5569				RTS	PC				
5570									
5571									

017012	047045	047045	040445	RPT1A:	.ASCIZ	/#N#N#AUNIT #D1#S3#APASS:#D5#S3#ARECORD:#D5#N/
017067	045	041101	052131	RPT1B:	.ASCIZ	/#ABYTES WRITTEN #D3#A,#Z3#A,#Z3#A,#Z3#N/
017140	040445	054502	042524	RPT1C:	.ASCIZ	/#ABYTES READ REV #D3#A,#Z3#A,#Z3#A,#Z3#N/
017211	045	041101	052131	RPT1D:	.ASCII	/#ABYTES READ FWD #D3#A,#Z3#A,#Z3#A,#Z3#N/
017261	045	031123	022463		.ASCIZ	/#S23#AWRT#S4#ARDR#S4#ARDF#N/
017315	045	051101	041505	RPT1F:	.ASCIZ	/#ARECOVERABLE ERRORS #D5#S2#D5#S2#D5#N/
017366	040445	047125	042522	RPT1G:	.ASCIZ	/#AUNRECOVERABLE ERRORS #D5#S2#D5#S2#D5#N/
017437	045	053501	044522	RPT1E:	.ASCIZ	/#AWRITE RETRIES#S8#D5#N/
017467	045	022516	031104	RPT1J:	.ASCIZ	/#N#D2#A BAD SPOTS THIS TAPE P/SS PRECEDING RECORD #:/
017554	042045	022465	030523	RPT1K:	.ASCIZ	/#D5#S1/
017563	045	051501	042520	RPT1I:	.ASCII	"#ASPEC COND#S3#AHARD#S3#AFATAL#S3#ACOMPARE#N
017637	045	031523	042045		.ASCIZ	/#S3#D5#S3#D5#S3#D5#S3#D5#N#N/

5572				.LIST	BEX				
5573				.EVEN					
5574	017674			ENDRPT					
5575	017674			L10010:					
5576	017674	104425						TRAP	C#RPT
5577									
5578				.SBTTL	LOAD DEVICE PROTECTION TABLE				
5579									
5580				...					
5581				TABLE FOR SUPERVISOR TO IDENTIFY THE P TBL FOR THE LOAD DEV					
5582				THE SUPERVISOR USES THE TBL TO WARN THE OPERATOR WHEN HE TRIES TO TEST THE LOAD					
5583									
5584									
5585	017676			BGNPROT					

5586	017676	
5587	017676	000000
5588	017700	177777
5589	017702	177777
5590	017704	

```

L$PROT::
        .WORD 0
        .WORD 1
        .WORD 1
ENDPROT

```

```

;P TBL OFFSET OF TSSR, THE TS11 CSR
;P TBL OFFSET OF MASS BUS UNIT #: 1 = NOT A MAS
;P TBL OFFSET OF DRIVE #: 1 = NONE, ONE DRIVE P

```

```

5591 .SBTTL INITIALIZE SECTION
5592
5593
5594 ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
5595 ; AT THE BEGINNING OF EACH PASS.
5596 ;
5597
5598 017704 BGNINIT
5599 017704 L$INIT::
5600
5601 017704 INIT10: IF #BIT0!BIT1 SET IN #CMOPKT THEN ;IF CMD PACKET IS NOT ON MODULO 4 BOUN
5602 017704 032727 000003 002310 BIT #BIT0!BI
5603 017712 001426 BEQ 50355$
5604 017714 ERRSF #1,CMOPKM ;PRINT ERROR MSG,
5605 017714 104454 TRAP C$ERSF
5606 017716 000001 .WORD 1
5607 017720 004102 .WORD CMOPKM
5608 017722 000000 .WORD 0
5609 017724 012746 000010 MOV #8., (SP) ;SETUP STACK FOR LONG DELAY
5610 017730 97$: DELAY 250. ;GO TO SUPERVISOR, WAIT 2 SECONDS.
5611 017730 012727 000372 MOV #250.,(P
5612 017734 000000 .WORD 0
5613 017736 013727 002116 MOV L$DLV,(P
5614 017742 000000 .WORD 0
5615 017744 005367 177772 DEC -6(PC)
5616 017750 001375 BNE . 4
5617 017752 005367 177756 DEC -22(PC)
5618 017756 001367 BNF .-20
5619 017760 005316 DEC (SP) ;SUBTRACT 1
5620 017762 001362 BNE 97$ ;BRANCH UNTIL DONE
5621 017764 005726 TST (SP). ;CLEAN UP THE STACK
5622 017766 000746 BR INIT10 ;
5623 017770 ENDIF
5624 017770
5625
5626 017770 IFB CLRFLG NE #0 THEN ;IF CLR COUNTERS FLAG SET:
5627 017770 105737 002202 TSTB CLRFLG
5628 017774 001413 BEQ 50356$
5629 017776 105037 002202 CLRFB CLRFLG ;INIT CLR FLAG.
5630 020002 LET R2 := #0
5631 020002 005002 CLR R2
5632 020004 WHILE R2 NE #CNTLEN DO
5633 020004
5634 020004 020227 000550 50357$: CMP R2,#CNTL
5635 020010 001405 BEQ 50360$
5636 020012 LET WRBC(R2) := #0 ;CLR ALL STATISTICAL COUNTERS.
5637 020012 005062 002554 CLR WRBC(R2)
5638 020016 LET R2 := R2 + #2
5639 020016 062702 000002 ADD #2,R2
5640 020022 ENDDO
5641 020022 000770 BR 50357$
5642 020024 50360$:
5643 020024 ENDIF
5644 020024 50356$:
5645
5646 020024 IFB RRANV NE #0 THEN ;IF RESET RANDOM VARIABLE FLAG IS SET TM

```

5647	020024	105737	002203				TSTB	RRANV
5648	020030	001406					BEQ	50361\$
5649	020032				LET RANB := #RANBC			
5650	020032	012737	153624	003360				
5651	020040				LET RANS := #RANSC			
5652	020040	012737	032561	003362				
5653	020046				ENDIF			
5654	020046							
5655	020046				READEF #EF.START		50361\$:	
5656	020046	012700	000040					
5657	020052	104447						
5658	020054				BNCOMPLETE INIT15			
5659	020054	103026						
5660	020056				LET STAFLG :B= STAFLG + #1			
5661	020056	105237	003452					
5662	020062				LET R5 := #6			
5663	020062	012705	000006					
5664	020066				REPEAT			
5665	020066							
5666	020066				LET DEVTBL(R5) := #NINUSE		50362\$:	
5667	020066	012765	177774	002532				
5668	020074				LET R5 := R5 - #2			
5669	020074	162705	000002					
5670	020100				UNTIL R5 EQ #0			
5671	020100	005705						
5672	020102	001371						
5673	020104				LET R5 := L\$UNIT SHIFT 1			
5674	020104	013705	002012					
5675	020110	006305						
5676	020112				REPEAT			
5677	020112							
5678	020112				LET R5 := R5 - #2		50363\$:	
5679	020112	162705	000002					
5680	020116				LET DEVTBL(R5) := R5 SHIFT 1			
5681	020116	010565	002532					
5682	020122	006265	002532					
5683	020126				UNTIL R5 EQ #0			
5684	020126	005705						
5685	020130	001370						
5686								
5687	020132				INIT15: READEF #EF.PWR			
5688	020132	012700	000034					
5689	020136	104447						
5690	020140				BNCOMPLETE INIT16			
5691	020140	103004						
5692	020142				LET STAFLG :B= STAFLG + #1			
5693	020142	105237	003452					
5694	020146				LET PWRFLG :B= PWRFLG + #1			
5695	020146	105237	003453					
5696								
5697	020152				INIT16: RFLAGS OPFLAG			
5698	020152	104421						
5699	020154	010037	003456					
5700	020160				LET R3 := #0			
5701	020160	005003						
5702	020162				IFB PWRFLG EQ #0 THEN			

5703	020162	105737	003453			TSTB	PWRFLG
5704	020166	001020				BNE	503648
5705	020170			READF #EF.NEW	;UPDATE PASS COUNT WHEN		
5706	020170	012700	000035			MOV	#EF.NEW.
5707	020174	104447				TRAP	C\$REFG
5708	020176			IFCOND CS THEN	;SUPERVISOR IS IN NEW PASS	BCC	503658
5709	020176	103014					
5710	020200			IFB STAF LG EQ #0 THEN	;AND DIAG WAS NEITHER STARTED	TSTB	STAF LG
5711	020200	105737	003452			BNE	503668
5712	020204	001010					
5713	020206			READF #EF.RES	;NOR		
5714	020206	012700	000037			MOV	#EF.RES.
5715	020212	104447				TRAP	C\$REFG
5716	020214			IFCOND CC THEN	;RESTARTED		
5717	020214	103402				BCS	503678
5718	020216			LET R3 := COMP R3	;DO IT	COM	R3
5719	020216	005103					
5720	020220			ELSE		BR	503708
5721	020220	000401					
5722	020222						503678:
5723	020222			LET R3 := R3 + #1	;SET 1ST PASS IF NEW PASS AND	INC	R3
5724	020222	005203					
5725	020224			ENDIF	;RESTARTING		
5726	020224						503708:
5727	020224			ELSE		BR	503718
5728	020224	000401					
5729	020226						503668:
5730	020226			LET R3 := R3 + #1	;SET 1ST PASS IF NEW PASS AND	INC	R3
5731	020226	005203					
5732	020230			ENDIF	;STARTING		
5733	020230						503718:
5734	020230			ENDIF	;DO NOT UPDATE IT ON CONTINUE		
5735	020230						503658:
5736	020230			ENDIF	;OR ON POWER FAIL		
5737	020230						503648:
5738	020230	004737	015452	JSR PC,FIRSTU	;INIT DEVICE POINTER.		
5739	020234			LET R2 := #0	;INIT DEVICE COUNTER.		
5740	020234	005002				CLR	R2
5741	020236			WHILE DEVTBL(R5) NE #END DO			
5742	020236						503728:
5743	020236	026527	002532 177777			CMP	DEVTBL(R
5744	020244	001450				BEQ	503738
5745	020246			LET R2 := R2 + #1			
5746	020246	005202				INC	R2
5747	020250			LET R0 := R5 SHIFT 1			
5748	020250	010500				MOV	R5,R0
5749	020252	006200				ASR	R0
5750	020254			GPHARD R0,R0	;GET HARDWARE P TABLE FROM SUPER.	TRAP	C\$GPHRD
5751	020254	104442					
5752	020256			IFCOND CS THEN			
5753	020256	103036				BCC	503748
5754	020260			LET TSSR(R5) := (R0)	;SAVE TSSR ADDRESS.		
5755	020260	011065	002462			MOV	(R0),TSS
5756	020264			LET TSDB(R5) := (R0) + #2	;SAVE TSDB ADDRESS.		
5757	020264	012065	002452			MOV	(R0),TS
5758	020270	162765	000002 002452			SUB	#2,TSDB

5759	020276			LET TSVCT(R5) := (R0)	;SAVE INTERRUPT VECTOR ADDRESS.		
5760	020276	011065	002472			MOV	(R0),TSV
5761	020302			SETVEC TSVCT(R5),TS4INT(R5),#INTPRI	;SET UP INTERUPT PROCESSING COND		
5762	020302	012746	000340			MOV	#INTPRI,
5763	020306	016546	002512			MOV	TS4INT(R
5764	020312	016546	002472			MOV	TSVCT(R5
5765	020316	012746	000003			MOV	#3,-(SP)
5766	020322	104437				TRAP	C#SVEC
5767	020324	062706	000010			ADD	#10,SP
5768	020330			IF R3 NE #0 THEN	;ACTUAL PASSCOUNT UPDATE PER R3		
5769	020330	005703				TST	R3
5770	020332	001410				BEQ	50375\$
5771	020334			IF R3 LT #0 THEN			
5772	020334	005703				TST	R3
5773	020336	002003				BGE	50376\$
5774	020340			LET PASCNT(R5) := PASCNT(R5) + #1			
5775	020340	005265	003254			INC	PASCNT(R
5776	020344			ELSE			
5777	020344	000403				BR	50377\$
5778	020346						50376\$:
5779	020346			LET PASCNT(R5) := #1			
5780	020346	012765	000001 003254			MOV	#1,PASCNT
5781	020354			ENDIF			
5782	020354						50377\$:
5783	020354			ENDIF			
5784	020354						50375\$:
5785	020354			ENDIF			
5786	020354						50374\$:
5787	020354			LET RECCNT(R5) := #0	;CLEAR RECORD COUNT		
5788	020354	005065	003324			CLR	RECCNT(R
5789	020360	004737	015520	JSR PC,NEXTU	;DO IT FOR ALL DEVICES.		
5790	020364			ENDDO			
5791	020364	000724				BR	50372\$
5792	020366						50373\$:
5793							
5794	020366			IF R2 EQ #0 THEN	;IF THERE ARE NO UNITS:		
5795	020366	005702				TST	R2
5796	020370	001033				BNE	50400\$
5797	020372			PRINTF #AUDRPM	;PRINT ALL UNITS DROPPED.		
5798	020372	012746	004645			MOV	#AUDRPM,
5799	020376	012746	000001			MOV	#1,-(SP)
5800	020402	010600				MOV	SP,R0
5801	020404	104417				TRAP	C#PNTE
5802	020406	062706	000004			ADD	#4,SP
5803	020412	012746	000010	MOV #8,-(SP)	;SETUP STACK FOR LONG DELAY		
5804	020416			98\$: DELAY 250.	;GO TO SUPERVISOR, WAIT 2 SECONDS.		
5805	020416	012727	000372			MOV	#250,(P
5806	020422	000000				.WORD	0
5807	020424	013727	002116			MOV	L#DLT,(P
5808	020430	000000				.WORD	0
5809	020432	005367	177772			DEC	6(PC)
5810	020436	001375				BNE	,-4
5811	020440	005367	177756			DEC	22(PC)
5812	020444	001367				BNE	,-20
5813	020446	005316		DEC (SP)	;SUBTRACT 1		
5814	020450	001362		BNE 98\$	;BRANCH UNTIL DONE		

5815	020452	005726			TST (SP) ; CLEAN UP THE STACK	
5816	020454				BREAK ; GO TO SUPERVISOR, CHECK TTY.	
5817	020454	104422				C\$BRK TRAP
5818	020456				DOCLN ; DO CLEAN CODE + ABORT PASS.	
5819	020456	104444				C\$DOCLN TRAP
5820	020460				ENDIF	
5821	020460					50400\$:
5822						
5823	020460				SETPRI #PRI00 ; LOWER CPU PRIORITY TO 0	
5824	020460	012700	000000			MOV #PRI00,R
5825	020464	104441				C\$SPRI TRAP
5826	020466				IFB IREC EQ #0 AND #ADR NOTSETIN OPFLAG THEN ; IF ERROR RECOVERY IS EN	
5827	020466	105737	002211			TSTB IREC
5828	020472	001152				BNE 50401\$
5829	020474	032737	000020	003456		BIT #ADR,OPF
5830	020502	001146				BNE 50401\$
5831	020504	004737	015452		JSR PC,FIRSTU ; AND AUTO DROP NOT CALLED, THEN SET UP F	
5832	020510				WHILE DEVTBL(R5) NE #END DO ; WHILE THERE ARE MORE DEVICES:	
5833	020510					50402\$:
5834	020510	026527	002532	177777		CMP DEVTBL(R
5835	020516	001540				BEQ 50403\$
5836	020520				BEGIN COUNTER ; START 3.5 MINUTE COUNTER	
5837	020520				INCR TIME1 FROM #1 TO #25 BY #1	
5838	020520	012737	000001	003364		MOV #1,TIME1
5839	020526	000402				BR 50405\$
5840	020530					50406\$:
5841	020530	005237	003364			INC TIME1
5842	020534					50405\$:
5843	020534	023727	003364	000025		CMP TIME1,#2
5844	020542	003113				BGT 50407\$
5845	020544				LET #TSDB(R5) := #GSCP ; AND GET UNITS STATUS	
5846	020544	012775	002320	002452		MOV #GSCP,#2
5847	020552				DELAY 1 ; WAIT	
5848	020552	012727	000001			MOV #1,(PC)
5849	020556	000000				.WORD 0
5850	020560	013727	002116			MOV L\$DL1,(P
5851	020564	000000				.WORD 0
5852	020566	005367	177772			DEC -6(PC)
5853	020572	001375				BNE -.4
5854	020574	005367	177756			DEC -22(PC)
5855	020600	001367				BNE .20
5856	020602				IF #TS.SSR SETIN #TSSR(R5) THEN	
5857	020602	032775	000200	002462		BIT #TS.SSR,
5858	020610	001420				BEQ 50410\$
5859	020612				IF #TS.OFL NOTSETIN #TSSR(R5) THEN	
5860	020612	032775	000100	002462		BIT #TS.OFL,
5861	020620	001001				BNE 50411\$
5862	020622				LEAVE COUNTER ; EXIT COUNTER WHEN UNIT ON LINE	
5863	020622	000463				BR 50404\$
5864	020624				ELSE	
5865	020624					50411\$:
5866	020624				PRINTF #OFL INM,DEVTBL(R5) ; PRINT UNIT OFF LINE EVERY 10 SEC	
5867	020624	016546	002532			MOV DEVTBL(R
5868	020630	012746	005127			MUL #OFL INM,
5869	020634	012746	000002			MOV #2,(SP)
5870	020640	010600				MOV SP,R0

5871	020642	104417							TRAP	C\$PNTF
5872	020644	062706	000006						ADD	#6,SP
5873	020650									
5874	020650									
5875	020650									
5876	020650	000412								
5877	020652									
5878	020652									
5879	020652	016546	002532							
5880	020656	012746	021616							
5881	020662	012746	000002							
5882	020666	010600								
5883	020670	104417								
5884	020672	062706	000006							
5885	020676									
5886	020676									
5887	020676									
5888	020676	012737	000001	003366						
5889	020704	000402								
5890	020706									
5891	020706	005237	003366							
5892	020712									
5893	020712	023727	003366	000013						
5894	020720	003023								
5895	020722	012746	000004							
5896	020726									
5897	020726	012727	000372							
5898	020732	000000								
5899	020734	013727	002116							
5900	020740	000000								
5901	020742	005367	177772							
5902	020746	001375								
5903	020750	005367	177756							
5904	020754	001367								
5905	020756	005316								
5906	020760	001362								
5907	020762	005726								
5908	020764									
5909	020764	104422								
5910	020766									
5911	020766	000747								
5912	020770									
5913	020770									
5914	020770	000657								
5915	020772									
5916	020772									
5917	020772									
5918	020772									
5919	020772	023727	003364	000025						
5920	021000	003404								
5921	021002	004737	011224							
5922	021006	004737	011736							
5923	021012									
5924	021012									
5925										
5926	021012	004737	015520							

```

ENDIF
ELSE
PRINTF #NRDYM,DEVTBL(R5)
MOV DEVTBL(R
MOV #NRDYM,-
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #6,SP
ENDIF
INCR TIME2 FROM #1 TO #13 BY #1
MOV #1,TIME2
BR 50414$
50415$: INC TIME2
50414$: CMP TIME2,#1
BGT 50416$
;SETUP STACK FOR LONG DELAY
;GO TO SUPERVISOR, WAIT 1 SECOND
MOV #250,(P
.WORD 0
MOV L$DLY,(P
.WORD 0
DEC -6(PC)
BNE -4
DEC -22(PC)
BNE .20
DEC (SP) ;SUBTRACT 1
BNE 99$ ;BRANCH UNTIL DONE
TST (SP). ;CLEAN UP THE STACK
BREAK ;ALLOW TERMINAL INTERRUPT
TRAP C$BRK
ENDINC
BR 50415$
50416$:
BR 50406$
50407$:
50404$:
IF TIME1 GT #25 THEN ;IF OFF LINE FOR 3.5 MINUTES
CMP TIME1,#2
BLE 50417$
JSR PC,MOVMSG ;GET MESSAGE PACKET
JSR PC,TCC1 ;PRINT ERROR AND DROP OFF LINE UNIT
ENDIF
50417$:
JSR PC,NEXTU ;REPEAT UNTIL ON LINE OR TIMED OUT.
;SET UP FOR NEXT UNIT.

```



5927	021016				ENDDO			
5928	021016	000634						
5929	021020						50403\$:	BR 50402\$
5930	021020				ENDIF			
5931	021020						50401\$:	
5932	021020				IFB PWRFLG EQ #0 THEN			
5933	021020	105737	003453					TSTB PWRFLG
5934	021024	001026						BNE 50420\$
5935	021026				MEMORY DATAW			;REQUEST MEMORY FROM SUPER FOR RD/WR BUF
5936	021026	104431						TRAP C\$MEM
5937	021030	010037	003334					MOV RO,DATAW
5938	021034				LET DATARD :- DATAW + #DATCNT			;SET RD BFR AD
5939	021034	013737	003334	003336				MOV DATAW,D
5940	021042	062737	004000	003336				ADD #DATCNT,
5941	021050				IF @DATAW LT #DATCNT THEN			;WHEN NOT ENOUGH FREE MEMO AVAILABLE
5942	021050	027727	162260	004000				CMP @DATAW,
5943	021056	002011						BGE 50421\$
5944	021060				PRINTF #MEMOM			;WARN OPERATOR
5945	021060	012746	021126					MOV #MEMOM,
5946	021064	012746	000001					MOV #1,-(SP)
5947	021070	010600						MOV SP,RO
5948	021072	104417						TRAP C\$PNTF
5949	021074	062706	000004					ADD #4,SP
5950	021100				DOCLN			;AND ABORT PASS
5951	021100	104444						TRAP C\$DCLN
5952	021102				ENDIF			;DIAG MUST BE RE LOADED IN A CPU WITH LARGER MEMO
5953	021102							50421\$:
5954	021102				ENDIF			
5955	021102							50420\$:
5956								
5957	021102				LET CHGFLG :B= #0			;CLR CHANGE CMD SEQ TBL FLAG.
5958	021102	105037	002212					CLRB CHGFLG
5959	021106				LET R3 := #ENDFLG			
5960	021106	012703	003452					MOV #ENDFLG,
5961	021112	004737	011154		JSR PC,CLRERR			;CLEAR ALL FLAGS.
5962	021116				LET PWRFLG :B= #0			;CLEAR THE POWER FAIL FLAG.
5963	021116	105037	003453					CLRB PWRFLG
5964								
5965	021122				EXIT INIT			
5966	021122	104432						TRAP C\$EXIT
5967	021124	000104						.WORD L10012 .

5968  
5969 021126 040445 051106 042505  
5970 021134 046440 046505 020117  
5971 021142 047524 020117 046523  
5972 021150 046101 020114 047506  
5973 021156 020122 042122 053455  
5974 021164 020122 043102 051522  
5975 021172 047045  
5976 021174 040445 042522 046055  
5977 021202 040517 020104 047111  
5978 021210 046040 051101 042507  
5979 021216 020122 042515 047515  
5980 021224 047045 000  
5981 021230  
5982  
5983 021230  
5984 021230  
5985 021230 104411

MEMOM: .ASCII /#AFREE MEMO TOO SMALL FOR RD WR BFRS#N/

.ASCIZ /#ARE-LOAD IN LARGER MEMO#N/

.EVEN

ENDINIT

L10012:

TRAP C\$INIT

```

5986 .SBTTL AUTO DROP SECTION
5987
5988
5989 ;**
5990 ;SECTION EXECUTED AFTER THE INIT CODE WHEN "ADR" FLAG IS SET BY OPERATOR
5991 ;SECTION CHECKS FOR A VALID INTERFACE LOCATION. DROPS UNIT IF NO RESPONSE
5992 ;FROM INTERFACE
5993 ;
5994
5995 L$AUTO::
5996
5997 021232 004737 015452 JSR PC,FIRSTU ;FIND FIRST UNIT
5998 021236 WHILE DEVTBL(R5) NE #END DO ;
5999 021236 ;
6000 021236 026527 002532 177777 50422$:
6001 021244 001525 CMP DEVTBL(R
6002 021246 LET TRAPD4 :B= #0 ; BEQ 50423$
6003 021246 105037 003454 CLR B TRAPD4
6004 021252 SETVEC #4,#TRAP4,#PRI07 ;SET VECTOR 4
6005 021252 012746 000340 MOV #PRI07,-
6006 021256 012746 021646 MOV #TRAP4,-
6007 021262 012746 000004 MOV #4,-(SP)
6008 021266 012746 000003 MOV #3,-(SP)
6009 021272 104437 TRAP C$SVEC
6010 021274 062706 000010 ADD #10,SP
6011 021300 LET R2 := @TSSR(R5) ;ADDRESS TS11 INTERFACE
6012 021300 017502 002462 MOV @TSSR(R5
6013 021304 CLRVEC #4 ;CLEAR VECTOR AT 4
6014 021304 012700 000004 MOV #4,R0
6015 021310 104436 TRAP C$CVEC
6016 021312 IFB TRAPD4 NE #0 THEN
6017 021312 105737 003454 TSTB TRAPD4
6018 021316 001423 BEQ 50424$
6019 021320 LET FTLCNT(R5) := FTLCNT(R5) + #1
6020 021320 005265 003314 INC FTLCNT(R
6021 021324 PRINTF #AUTODM,TSSR(R5) ;PRINT ERROR
6022 021324 016546 002462 MOV TSSR(R5)
6023 021330 012746 021522 MOV #AUTODM,
6024 021334 012746 000002 MOV #2,-(SP)
6025 021340 010600 MOV SP,R0
6026 021342 104417 TRAP C$PNTF
6027 021344 062706 000006 ADD #6,SP
6028 021350 LET DROPN :- DEVTBL(R5) ;SAVE # OF UNIT TO BE DROPPED.
6029 021350 016537 002532 015774 MOV DEVTBL(R
6030 021356 LET R0 := R5 SHIFT 1 ;R0-LOGICAL DEVICE NUMBER
6031 021356 010500 MOV R5,R0
6032 021360 006200 ASR R0
6033 021362 DODU R0 ;DROP THE UNIT: EXEC BGNDU-ENDDU CODE IF
6034 021362 104451 TRAP C$DODU
6035 021364 ELSE
6036 021364 000452 BR 50425$
6037 021366
6038 021366 LET @TSDB(R5) :- #GSCPK ;SEND GET STATUS COMMAND
6039 021366 012775 002320 002452 50424$:
6040 021374 004737 011170 MOV #GSCPK,#
6041 021400 JSR PC,WSSR ;WAIT
IF #TS.SSR SETIN @TSSR(R5) THEN

```

6042	021400	032775	000200	002462		BIT	#TS.SSR,
6043	021406	001423				BEQ	50426\$
6044	021410				IF #TS.OFL SETIN @TSSR(R5) THEN		
6045	021410	032775	000100	002462		BIT	#TS.OFL,
6046	021416	001416				BEQ	50427\$
6047	021420				LET FTLCNT(R5) := FTLCNT(R5) + #1		
6048	021420	005265	003314		PRINTF #OFLINM,DEVTBL(R5)	INC	FTLCNT(R
6049	021424					MOV	DEVTBL(R
6050	021424	016546	002532			MOV	#OFLINM,
6051	021430	012746	005127			MOV	#2,-(SP)
6052	021434	012746	000002			MOV	SP,RO
6053	021440	010600				TRAP	C\$PNTF
6054	021442	104417				ADD	#6,SP
6055	021444	062706	000006				
6056	021450	004737	015704		JSR PC,DROPUA		
6057	021454				ENDIF		
6058	021454					50427\$:	
6059	021454				ELSE		
6060	021454	000416				BR	50430\$
6061	021456					50426\$:	
6062	021456				LET FTLCNT(R5) := FTLCNT(R5) + #1		
6063	021456	005265	003314		PRINTF #NRDYM,DEVTBL(R5)	INC	FTLCNT(R
6064	021462					MOV	DEVTBL(R
6065	021462	016546	002532			MOV	#NRDYM,-
6066	021466	012746	021616			MOV	#2,-(SP)
6067	021472	012746	000002			MOV	SP,RO
6068	021476	010600				TRAP	C\$PNTF
6069	021500	104417				ADD	#6,SP
6070	021502	062706	000006				
6071	021506	004737	015704		JSR PC,DROPUA		
6072	021512				ENDIF		
6073	021512					50430\$:	
6074	021512				ENDIF		
6075	021512					50425\$:	
6076	021512	004737	015520		JSR PC,NEXTU		
6077	021516				ENDDO		
6078	021516	000647				BR	50422\$
6079	021520					50423\$:	
6080							
6081	021520				ENDAUTO		
6082	021520				L10013:		
6083	021520	104461				TRAP	C\$AUTO
6084							
6085	021522	040445	052502	020123	AUTODM: .ASCII /#ABUS TRAP AT #06#N/		
6086	021530	051124	050101	040440			
6087	021536	020124	047445	022466			
6088	021544	116					
6089	021545	045	044501	052116	.ASCIZ /#AINTERFACE BAD OR NOT SET TO ABOVE AD#N/		
6090	021552	051105	040506	042503			
6091	021560	041040	042101	047440			
6092	021566	020122	047516	020124			
6093	021574	042523	020124	047524			
6094	021602	040440	047502	042526			
6095	021610	040440	022504	000116			
6096	021616	040445	047125	052111	NRDYM: .ASCIZ /#AUNIT #D1#A NOT RD1#N/		
6097	021624	022440	030504	040445			

B.L.P

6098 021632 047040 052117 051040  
6099 021640 054504 047045 000  
6100 021646  
6101  
6102  
6103  
6104  
6105  
6106 021646  
6107 021646 105237 003454  
6108 021657 000002  
6109  
6110  
6111

```
.EVEN  
;  
; DEVICE BUS TRAP HANDLER  
; OUTPUT: TRAPD4 BYTE 1: TRAPED AT 4  
; 0: NO TRAP  
TRAP4:: LET TRAPD4 :B- TRAPD4 . #1  
RTI
```

INCB TRAPD4

CLP

6112  
 6113  
 6114  
 6115  
 6116  
 6117  
 6118  
 6119 021654  
 6120 021654  
 6121  
 6122  
 6123 021654 004737 015452  
 6124 021660  
 6125 021660  
 6126 021660 026527 002532 177777  
 6127 021666 001410  
 6128 021670 004737 011170  
 6129 021674  
 6130 021674 016500 002472  
 6131 021700 104436  
 6132 021702 004737 015520  
 6133 021706  
 6134 021706 000764  
 6135 021710  
 6136  
 6137 021710  
 6138 021710 104432  
 6139 021712 000002  
 6140  
 6141  
 6142 021714  
 6143 021714  
 6144 021714 104412

```

.SBTTL CLEANUP CODING SECTION

***
; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
; AT THE END OF EACH PASS.
;

        BGNCLN
L10014:

        JSR    PC,FIRSTU        ;FIND FIRST UNIT.
        WHILE DEVTBL(R5) NE #END DO
                                50431$:
                                CMP    DEVTBL(R
                                BEQ    50432$
                                ;WAIT FOR UNIT READY OR TIMEOUT.
                                JSR PC,WSSR        ;RELEASE INTERRUPT VECTORS FOR ALL DEV.
                                CLRVEC          TSVCT(R5)
                                MOV    TSVCT(R5
                                TRAP   C$CVEC

        JSR    PC,NEXTU        ;FIND NEXT UNIT.
        ENDDO

                                BR    50431$
                                50432$:
                                TRAP   C$EXIT
                                .WORD  L10014 .

        .EVEN

        ENDCLN
L10014:

                                TRAP   C$CLEAN
    
```

```

6145
6146
6147
6148
6149
6150
6151
6152
6153 021716
6154 021716
6155
6156 021716
6157 021716 010005
6158 021720 006305
6159 021722
6160 021722 012765 177774 002532
6161 021730
6162 021730 016500 002472
6163 021734 104436
6164 021736
6165 021736 013746 015774
6166 021742 012746 004616
6167 021746 012746 000002
6168 021752 010600
6169 021754 104417
6170 021756 062706 000006
6171
6172 021762
6173 021762 000167
6174 021764 000000
6175
6176
6177 021766
6178 021766
6179 021766 104453

      .SBTTL  DROP UNIT SECTION
      ;**
      ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
      ; TO NO LONGER BE TESTED. THAT CODE SHALL BE EXECUTED WHEN DODU
      ; MACRO IS CALLED WHILE IDU FLAG IS NOT SET BY OPERATOR
      ;
      BGNDU
L$DU::
      LET R5 := RO SHFT 1           ;R5 = LOGICAL DEVICE NUMBER x 2.
                                     MOV      RO,R5
                                     ASL      R5
      LET DEVTBL(R5) := #NINUSE     ;SET NOT IN USE FLAG FOR THE DEVICE.
                                     MOV      #NINUSE.
      CLRVEC TSVCT(R5)             ;RELEASE THE INTERRUPT VECTOR.
                                     MOV      TSVCT(R5
      PRINTF #DROPPM,DROPN         ;PRINT DROP DEVICE MESSAGE
                                     TRAP    C$CVEC
                                     MOV      DROPN,-(
                                     MOV      #DROPPM.
                                     MOV      #2,(SP)
                                     MOV      SP,RO
                                     TRAP    C$PNTF
                                     ADD     #6,SP
      EXIT      DU
                                     .WORD   JSJMP
                                     .WORD   110015 2
      .EVEN
      ENDDU
L10015:
                                     TRAP    C$DL
  
```

812

```

6180 .SBTTL ADD UNIT SECTION
6181
6182 ;**
6183 ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
6184 ; TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING. IF
6185 ; "EF.AUNIT" IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
6186 ;
6187
6188 021770 BGNAU
6189 021 770 L$AU::
6190
6191
6192 021770 LET R5 := R0 SHIFT 1 ;R5 = LOGICAL DEVICE NUMBER X 2.
6193 021770 010005 MOV R0,R5
6194 021772 006305 ASL R5
6195 021774 LET DEVTBL(R5) := R0 ;STORE UNIT # IN DEVICE TABLE.
6196 021774 010065 002532 MOV R0,DEVTB
6197 022000 GPHARD R0,R0 ;GET HARDWARE P TABLE FROM SUPER.
6198 022000 104442 TRAP C$GPHRD
6199 022002 LET TSSR(R5) := (R0) ;SAVE TSSR ADDRESS.
6200 022002 011065 002462 MOV (R0),TSS
6201 022006 LET TSDB(R5) := (R0) * #2 ;SAVE TSDB ADDRESS.
6202 022006 012065 002452 MOV (R0),TS
6203 022012 162765 000002 002452 SUB #2,TSDB(
6204 022020 LET TSVCT(R5) := (R0) ;SAVE INTERRUPT VECTOR ADDRESS.
6205 022020 011065 002472 MOV (R0),TSV
6206 022024 SETVEC TSVCT(R5),TS4INT(R5),#INTPRI ;SET UP INTERUPT PROCESSING COND
6207 022024 012746 000340 MOV #INTPRI,
6208 022030 016546 002512 MOV TS4INT(R
6209 022034 016546 002472 MOV TSVCT(R5
6210 022040 012746 000003 MOV #3,-(SP)
6211 022044 104437 TRAP C$SVEC
6212 022046 062706 000010 ADD #10,SP
6213 022052 LET INTFLG(R5) := #0 ;CLEAR INTERRUPT FLAGS.
6214 022052 005065 003416 CLR INTFLG(R
6215
6216 022056 EXIT AU
6217 022056 000167 .WORD JSJMP
6218 022060 000000 .WORD L10016-2
6219
6220 .EVEN
6221
6222 022062 ENDAU
6223 022062 L10016:
6224 022062 104452 TRAP C$AU
6225
6226 022064 ENDMOD
6227

```



```

6228
6229
6230
6231
6232
6233
6234
6235
6236
6237 022064
6238
6239 022064
6240 022064
6241
6242 022064
6243 022064 105037 003441
6244 022070
6245 022070 105037 003440
6246
6247 022074
6248 022074
6249 022074 104402
6250
6251 022076
6252 022076 012702 022722
6253 022102 004737 022676
6254 022106 004737 006510
6255 022112 004737 015452
6256 022116
6257 022116
6258 022116 026527 002532 177777
6259 022124 001434
6260 022126
6261 022126 016502 002502
6262 022132
6263 022132 062702 000012
6264 022136
6265 022136 011265 002522
6266 022142 042765 177400 002522
6267 022150
6268 022150 026527 003254 000001
6269 022156 001014
6270 022160
6271 022160 016546 002522
6272 022164 016546 002532
6273 022170 012746 003772
6274 022174 012746 000003
6275 022200 010600
6276 022202 104417
6277 022204 062706 000010
6278 022210
6279 022210
6280 022210 004737 015520
6281 022214
6282 022214 000740
6283 022216

.TITLE HARDWARE TFSTS
.SBTTL TEST 1: BASIC FUNCTIONS.
***
; TEST TO EXECUTE ALL TS04 FUNCTIONS.
;
      BGNMOD
      BGNTST
T1::
      LET RANDOM :B= #0           ;CLR THE RANDOM OPERATIONS FLAG.
                                   CLRB   RANDOM
      LET EXPBOT :B= #0          ;CLR EXPECT BOT FLAG.
                                   CLRB   EXPBOT
      BGNSUB                      ;SUBTEST 1 - SET CHAR, DRIVE INIT, GET S
T1.1:
                                   TRAP   C#BSUB
      LET R2 := #BFSEQO           ;ADR OF CMD SEQ.
                                   MOV    #BFSEQO,
      JSR PC,BFSEQ                ;SET UP CMD SEQ.
      JSR PC,EXALL                ;EXECUTE CMD SEQ ON ALL DEVICES.
      JSR PC,FIRSTU              ;FIND THE FIRST UNIT.
      WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
                                   50433$:
                                   CMP    DEVTBL(R
                                   BEQ    50434$
      LET R2 := MSGPKA(R5)        ;GET MSG PACKET ADR.
                                   MOV    MSGPKA(R
      LET R2 := R2 + #12         ;GET XSTAT2 ADR.
                                   ADD    #12,R2
      LET TS4CL(R5) := (R2) CLR.BY #177400 ;STORE CODE LEVEL FROM DTR BYTE.
                                   MOV    (R2),TS4
                                   BIC    #177400,
      IF PASCNT(R5) EQ #1 THEN   ;IF THIS IS PASS 1 THEN:
                                   CMP    PASCNT(R
                                   BNE    50435$
      PRINTF #CODELM,DEVTBL(R5),TS4CL(R5) ;PRINT THE TS04 MICROCODE LEVEL.
                                   MOV    TS4CL(R5
                                   MOV    DEVTBL(R
                                   MOV    #CODELM,
                                   MOV    #3,-(SP)
                                   MOV    SP,R0
                                   TRAP   C#PNTF
                                   ADD    #10,SP
      ENDIF
      JSR PC,NEXTU                ;FIND NEXT UNIT.
      ENDDO
                                   50435$:
                                   BR    50433$
                                   50434$:
    
```

6284	022216			ENDSUB			
6285	022216			L10020:			
6286	022216	104403				TRAP	C\$ESUB
6287							
6288	022220			BGNSUB		;SUBTEST 2 REWIND.	
6289	022220			T1.2:			
6290	022220	104402				TRAP	C\$BSUB
6291							
6292	022222			LET R2 := #BFSEQ1		;ADR OF CMD SEQ.	
6293	022222	012702	022774			MOV	#BFSEQ1.
6294	022226	004737	022676	JSR PC,BFSEQ		;SET UP CMD SEQ.	
6295	022232	004737	006510	JSR PC,EXALL		;EXECUTE CMD SEQ ON ALL DEVICES.	
6296	022236			LET STAF LG :B= #0		;CLEAR START FLAG	
6297	022236	105037	003452			CLRB	STAF LG
6298	022242			ENDSUB			
6299	022242			L10021:			
6300	022242	104403				TRAP	C\$ESUB
6301							
6302	022244			BGNSUB		;SUBTEST 3 WRITE/VERIFY.	
6303	022244			T1.3:			
6304	022244	104402				TRAP	C\$BSUB
6305							
6306	022246			LET R2 := #BFSEQ2		;ADR OF CMD SEQ.	
6307	022246	012702	023006			MOV	#BFSEQ2.
6308	022252	004737	022676	JSR PC,BFSEQ		;SET UP CMD SEQ.	
6309	022256	004737	006510	JSR PC,EXALL		;EXECUTE CMD SEQ ON ALL DEVICES.	
6310	022262			ENDSUB			
6311	022262			L10022:			
6312	022262	104403				TRAP	C\$ESUB
6313							
6314	022264			BGNSUB		;SUBTEST 4 WRITE TAPE MARK, ERASE.	
6315	022264			T1.4:			
6316	022264	104402				TRAP	C\$BSUB
6317							
6318	022266			LET R2 := #BFSEQ3		;ADR OF CMD SEQ.	
6319	022266	012702	023100			MOV	#BFSEQ3.
6320	022272	004737	022676	JSR PC,BFSEQ		;SET UP CMD SEQ.	
6321	022276	004737	006510	JSR PC,EXALL		;EXECUTE CMD SEQ ON ALL DEVICES.	
6322	022302			ENDSUB			
6323	022302			L10023:			
6324	022302	104403				TRAP	C\$ESUB
6325							
6326	022304			BGNSUB		;SUBTEST 5 - SPACE FILES.	
6327	022304			T1.5:			
6328	022304	104402				TRAP	C\$BSUB
6329							
6330	022306			LET R2 := #BFSEQ4		;ADR OF CMD SEQ.	
6331	022306	012702	023152			MOV	#BFSEQ4.
6332	022312	004737	022676	JSR PC,BFSEQ		;SET UP CMD SEQ.	
6333	022316	004737	006510	JSR PC,EXALL		;EXECUTE CMD SEQ ON ALL DEVICES.	
6334	022322			ENDSUB			
6335	022322			L10024:			
6336	022322	104403				TRAP	C\$ESUB
6337							
6338	022324			BGNSUB		;SUBTEST 6 - SPACE RECORDS.	
6339	022324			T1.6:			

H12

SEQ 0150

6340	022324	104402				TRAP	C1BSUB
6341							
6342	022326			LET R2 := #BFSEQ5			
6343	022326	012702	023214				
6344	022332	004737	022676	JSR PC,BFSEQ			
6345	022336	004737	006510	JSR PC,EXALL			
6346	022342			ENDSUB			
6347	022342			L10025:			
6348	022342	104403				TRAP	C1ESUB
6349							
6350	022344			BGNSUB			
6351	022344			T1.7:			
6352	022344	104402				TRAP	C1BSUB
6353							
6354	022346			LET R2 := #BFSEQ6			
6355	022346	012702	023266				
6356	022352	004737	022676	JSR PC,BFSEQ			
6357	022356	004737	006510	JSR PC,EXALL			
6358	022362			ENDSUB			
6359	022362			L10026:			
6360	022362	104403				TRAP	C1ESUB
6361							
6362	022364			BGNSUB			
6363	022364			T1.8:			
6364	022364	104402				TRAP	C1BSUB
6365							
6366	022366			LET R2 := #BFSEQ7			
6367	022366	012702	023320				
6368	022372	004737	022676	JSR PC,BFSEQ			
6369	022376	004737	006510	JSR PC,EXALL			
6370	022402			ENDSUB			
6371	022402			L10027:			
6372	022402	104403				TRAP	C1ESUB
6373							
6374	022404			BGNSUB			
6375	022404			T1.9:			
6376	022404	104402				TRAP	C1BSUB
6377							
6378	022406			LET R2 := #BFSEQ8			
6379	022406	012702	023352				
6380	022412	004737	022676	JSR PC,BFSEQ			
6381	022416	004737	006510	JSR PC,EXALL			
6382	022422			ENDSUB			
6383	022422			L10030:			
6384	022422	104403				TRAP	C1ESUB
6385							
6386	022424			BGNSUB			
6387	022424			T1.10:			
6388	022424	104402				TRAP	C1BSUB
6389							
6390	022426			LET R2 := #BFSEQ9			
6391	022426	012702	023404				
6392	022432	004737	022676	JSR PC,BFSEQ			
6393	022436	004737	006510	JSR PC,EXALL			
6394	022442			ENDSUB			
6395	022442			L10031:			

112

6396	022442	104403					TRAP	C\$ESUB
6397								
6398	022444							
6399	022444							
6400	022444	104402			T1.11:	BGNSUB		;SUBTEST 11 - WTV SWAPPED DATA BYTES.
6401								
6402	022446					LET R2 := #BFSE10		;ADR OF CMD SEQ.
6403	022446	012702	023426				MOV	#BFSE10.
6404	022452	004737	022676			JSR PC,BFSEQ		;SET UP CMD SEQ.
6405	022456	004737	006510			JSR PC,EXALL		;WRITE/VERIFY RECORDS 1 AND 2.
6406	022462					LET SWBFLG :B= #1		;ENABLE BYTE SWAPPING.
6407	022462	112737	000001	003444			MOVB	#1,SWBFL
6408	022470	004737	006510			JSR PC,EXALL		;WRITE/VERIFY RECORDS 3 AND 4.
6409	022474					LET SWBFLG :B= #0		;DISABLE BYTE SWAPPING.
6410	022474	105037	003444				CLRB	SWBFLG
6411	022500					ENDSUB		
6412	022500				L10032:			
6413	022500	104403					TRAP	C\$ESUB
6414								
6415	022502					LET R2 := DATAWT * #10.		;INIT WRITE BUFFER POINTER.
6416	022502	013702	003334				MOV	DATAWT,R
6417	022506	062702	000012				ADD	#10.,R2
6418	022512					WHILE R2 NE DATAWT DO		;UNTIL 10 BYTES HAVE BEEN SWAPPED.
6419	022512							50436\$:
6420	022512	020237	003334				CMP	R2,DATAW
6421	022516	001402					BEQ	50437\$
6422	022520	000342				SWAB (R2)		;SWAP DATA BYTES IN WRITE BUFFER.
6423	022522					ENDDO		
6424	022522	000773					BR	50436\$
6425	022524							50437\$:
6426	022524					LET T1SWB :B= T1SWB * #1		;SET T1 SWAP BYTES FLAG FOR "CKDATA" SUB
6427	022524	105237	003447				INCB	T1SWB
6428								
6429	022530					BGNSUB		;SUBTEST 12 - READ SWAPPED DATA BYTES.
6430	022530				T1.12:			
6431	022530	104402					TRAP	C\$BSUB
6432								
6433	022532					LET CMDWRD := #RDR		;CMD IS READ REV.
6434	022532	012737	104401	003346			MOV	#RDR,CMD
6435	022540	004737	014504			JSR PC,VFEXC		;VERIFY ODD LENGTH SWAP (RECORD 4).
6436	022544					LET CMDPKT.CP.CNT := #12		;CHANGE BYTE COUNT TO 10.
6437	022544	012737	000012	002316			MOV	#12,CMDP
6438	022552	004737	014504			JSR PC,VFEXC		;VERIFY EVEN LENGTH SWAP (RECORD 3).
6439	022556					LET SWBFLG :B= #1		;ENABLE BYTE SWAPPING.
6440	022556	112737	000001	003444			MOVB	#1,SWBFL
6441	022564					LET CMDPKT.CP.CNT := #11		;CHANGE BYTE COUNT TO 9.
6442	022564	012737	000011	002316			MOV	#11,CMDP
6443	022572	004737	014504			JSR PC,VFEXC		;VERIFY ODD LENGTH SWAP (RECORD 2).
6444	022576					LET CMDPKT.CP.CNT := #12		;CHANGE BYTE COUNT TO 10.
6445	022576	012737	000012	002316			MOV	#12,CMDP
6446	022604	004737	014504			JSR PC,VFEXC		;VERIFY EVEN LENGTH SWAP (RECORD 1).
6447	022610					LET CMDWRD := #RDF		;CMD IS READ FWD.
6448	022610	012737	104001	003346			MOV	#RDF,CMD
6449	022616	004737	014504			JSR PC,VFEXC		;VERIFY EVEN LENGTH SWAP (RECORD 1).
6450	022622					LET CMDPKT.CP.CNT := #11		;CHANGE BYTE COUNT TO 9.
6451	022622	012737	000011	002316			MOV	#11,CMDP



```

6473 ; SUBROUTINE TO MOVE A COMMAND SEQUENCE TO THE SEQUENCE TABLE.
6474 ; INPUTS: R2 = FWA OF COMMAND SEQUENCE.
6475 ; OUTPUTS:
6476 ; REGISTERS:
6477 ; CALLS:
6478
6479 022676 BFSEQ: LET R1 := #CMDSEQ ;INIT SEQ TABLE ADDRESS.
6480 022676 012701 003460 ;WHILE THERE ARE MORE COMMANDS:
6481 022702 ;WHILE (R2) NE #END DO
6482 022702 50440$:
6483 022702 021227 177777 CMP (R2),#EN
6484 022706 001402 BEQ 50441$
6485 022710 LET (R1)+ := (R2)+ ;MOVE COMMANDS TO SEQ TABLE.
6486 022710 012221 ;MOV (R2)+,(R
6487 022712 ENDDO
6488 022712 000773 BR 50440$
6489 022714 50441$:
6490 022714 LET (R1) := #END ;STORE END OF SEQUENCE CODE.
6491 022714 012711 177777 MOV #END,(R1
6492 022720 000207 RTS PC ;RETURN.
6493
6494
6495
6496 ; BASIC FUNCTION COMMAND SEQUENCE
6497
6498 022722 140004 BFSEQ0: .WORD SCH ;SET CHAR. 200. (1)
6499 022724 000200 200
6500 022726 000001 1
6501 022730 000000 0
6502 022732 100013 DRI ;DRIVE INIT. (2)
6503 022734 000001 1
6504 022736 000001 1
6505 022740 000000 0
6506 022742 140004 SCH ;SET CHAR. 20 (3)
6507 022744 000020 20
6508 022746 000001 1
6509 022750 000000 0
6510 022752 100017 GES ;GET STATUS. (4)
6511 022754 000001 1
6512 022756 000001 1
6513 022760 000000 0
6514 022762 140004 SCH ;SET CHAR. 40. (5)
6515 022764 000040 40
6516 022766 000001 1
6517 022770 000000 0
6518 022772 177777 .WORD END
6519
6520 022774 102010 BFSEQ1: RWD ;REWIND TWICE. (6)
6521 022776 000001 1
6522 023000 000002 2
6523 023002 000000 0
6524 023004 177777 .WORD END
6525
6526 023006 104105 BFSEQ2: WTV ;WRITE/VERIFY PAT 1. (7)
6527 023010 004000 DATCNT
6528 023012 000001 1
    
```

6529	023014	000001		1		
6530	023016	104105		WTV	;WTV PAT 2.	(8)
6531	023020	004000		DATCNT		
6532	023022	000001		1		
6533	023024	000002		2		
6534	023026	104105		WTV	;WTV PAT 3.	(9)
6535	023030	004000		DATCNT		
6536	023032	000001		1		
6537	023034	000003		3		
6538	023036	104105		WTV	;WTV PAT 4.	(10)
6539	023040	004000		DATCNT		
6540	023042	000001		1		
6541	023044	000004		4		
6542	023046	104105		WTV	;WTV PAT 5.	(11)
6543	023050	004000		DATCNT		
6544	023052	000001		1		
6545	023054	000005		5		
6546	023056	104105		WTV	;WTV PAT 6.	(12)
6547	023060	004000		DATCNT		
6548	023062	000001		1		
6549	023064	000006		6		
6550	023066	104105		WTV	;WTV PAT 0.	(13)
6551	023070	004000		DATCNT		
6552	023072	000001		1		
6553	023074	000000		0		
6554	023076	177777	.WORD	END		
6555						
6556	023100	100011	BFSEQ3:	WTM	;WRITE TAPE MARK.	(14)
6557	023102	000001		1		
6558	023104	000001		1		
6559	023106	000000		0		
6560	023110	104005		WRT	;WRITE 10 RECORDS.	(15)
6561	023112	004000		DATCNT		
6562	023114	000010		10		
6563	023116	000001		1		
6564	023120	100411		ERS	;ERASE 10 TIMES.	(16)
6565	023122	000001		1		
6566	023124	000010		10		
6567	023126	000000		0		
6568	023130	100011		WTM	;WRITE TAPE MARK.	(17)
6569	023132	000001		1		
6570	023134	000001		1		
6571	023136	000000		0		
6572	023140	101011		WTR	;WTM RETRY	(18)
6573	023142	000001		1		
6574	023144	000001		1		
6575	023146	000000		0		
6576	023150	177777	.WORD	END		
6577						
6578	023152	105410	BFSEQ4:	SFR	;SPACE 2 FILES REV.	(19)
6579	023154	000002		2		
6580	023156	000001		1		
6581	023160	000000		0		
6582	023162	105010		SFF	;SPACE 2 FILES FWD.	(20)
6583	023164	000002		2		
6584	023166	000001		1		

6585	023170	000000		0		
6586	023172	105410		SFR	;SPACE 2 FILES REV.	(21)
6587	023174	000001		1		
6588	023176	000002		2		
6589	023200	000000		0		
6590	023202	105010		SFF	;SPACE 2 FILES FWD.	(22)
6591	023204	000001		1		
6592	023206	000002		2		
6593	023210	000000		0		
6594	023212	177777	.WORD	END		
6595						
6596	023214	102010	BFSEQ5:	RWD	;REWIND.	(23)
6597	023216	000001		1		
6598	023220	000001		1		
6599	023222	000000		0		
6600	023224	104010		SRF	;SPACE 7 RECORDS FWD.	(24)
6601	023226	000007		7		
6602	023230	000001		1		
6603	023232	000000		0		
6604	023234	104410		SRR	;SPACE 7 RECORDS REV.	(25)
6605	023236	000007		7		
6606	023240	000001		1		
6607	023242	000000		0		
6608	023244	104010		SRF	;SPACE 7 RECORDS FWD.	(26)
6609	023246	000001		1		
6610	023250	000007		7		
6611	023252	000000		0		
6612	023254	104410		SRR	;SPACE 7 RECORDS REV.	(27)
6613	023256	000001		1		
6614	023260	000007		7		
6615	023262	000000		0		
6616	023264	177777	.WORD	END		
6617						
6618	023266	102010	BFSEQ6:	RWD	;REWIND.	(28)
6619	023270	000001		1		
6620	023272	000001		1		
6621	023274	000000		0		
6622	023276	104005		WRT	;WRITE.	(29)
6623	023300	004000		DATCNT		
6624	023302	000001		1		
6625	023304	000001		1		
6626	023306	105005		WRR	;WRITE RETRY.	(30)
6627	023310	004000		DATCNT		
6628	023312	000001		1		
6629	023314	000001		1		
6630	023316	177777	.WORD	END		
6631						
6632	023320	104401	BFSEQ7:	RDR	;READ REV.	(31)
6633	023322	004000		DATCNT		
6634	023324	000001		1		
6635	023326	000001		1		
6636	023330	105401		RNR	;READ NEXT REV.	(32)
6637	023332	004000		DATCNT		
6638	023334	000001		1		
6639	023336	000001		1		
6640	023340	125401		RNF	;READ NEXT FWD.	(33)



6641	023342	004000		DATCNT		
6642	023344	000001		1		
6643	023346	000001		1		
6644	023350	177777	.WORD	END		
6645						
6646	023352	104001	BF SEQ8:	RDF	;READ FWD.	(34)
6647	023354	004000		DATCNT		
6648	023356	000001		1		
6649	023360	000001		1		
6650	023362	105001		RPF	;READ PREVIOUS FWD.	(35)
6651	023364	004000		DATCNT		
6652	023366	000001		1		
6653	023370	000001		1		
6654	023372	125001		RPR	;READ PREVIOUS REV.	(36)
6655	023374	004000		DATCNT		
6656	023376	000001		1		
6657	023400	000001		1		
6658	023402	177777	.WORD	END		
6659						
6660	023404	101012	BF SEQ9: .WORD	CLN	;CLEAN.	(37)
6661	023406	000001		1		
6662	023410	000001		1		
6663	023412	000000		0		
6664	023414	102010		RWD	;REWIND	(38)
6665	023416	000001		1		
6666	023420	000001		1		
6667	023422	000000		0		
6668	023424	177777	.WORD	END	;END OF SEQUENCE.	
6669						
6670	023426	104105	BFSE10:	WTV	;WRITE/VERIFY EVEN LENGTH.	(39)
6671	023430	000012		12		
6672	023432	000001		1		
6673	023434	000000		0		
6674	023436	104105		WTV	;WRITE/VERIFY ODD LENGTH.	(40)
6675	023440	000011		11		
6676	023442	000001		1		
6677	023444	000000		0		
6678	023446	177777	.WORD	END		
6679			.EVEN			
6680						
6681	023450		ENDTST			
6682	023450		L10017:			
6683	023450	104401			TRAP	C\$ETST

```

6684 .SBTTL TEST 2: DATA RELIABILITY.
6685
6686
6687
6688
6689 023452
6690 023452
6691
6692 023452
6693 023452 112737 000001 003441
6694 023460
6695 023460 105037 003440
6696 023464
6697 023464 012702 004000
6698 023470 005302
6699 023472
6700 023472 010237 003356
6701 023476 005137 003356
6702 023502 004737 006444
6703 023506
6704 023506 105737 003452
6705 023512 001404
6706 023514 004737 006470
6707 023520
6708 023520 105037 003452
6709 023524
6710 023524
6711 023524
6712 023524 012721 104105
6713 023530
6714 023530 012721 004000
6715 023534
6716 023534 012702 177740
6717 023540 005102
6718 023542
6719 023542 010221
6720 023544
6721 023544 012721 000007
6722 023550
6723 023550
6724 023550
6725 023550
6726 023550 020127 003550
6727 023554 002012
6728 023556
6729 023556 063737 003360 003362
6730 023564
6731 023564 013702 003362
6732 023570 042702 177741
6733 023574 004772 023732
6734 023600
6735 023600 000763
6736 023602
6737 023602
6738 023602 012711 177777
6739 023606 004737 006510

```

```

      BGNIST
T2::
      LET RANDOM :B= #1           ;SET THE RANDOM OPERATIONS FLAG.
      LET EXPBOT :B= #0           ;CLEAR EXPECT BOT FLAG.
      LET R2 := #DATCNT #1       ;SET UP THE RECORD LENGTH MASK.
      LET LENMSK := LOMP R2      ;ALLOW MAXIMUM BUFFER.
      JSR PC,SETCH                ;CMD 1 - SET CHARACTERISTIC.
      IFB STAFLG NE #0 THEN      ;IF STARTING THEN:
      JSR PC,SETRW                ;CMD2-REWIND
      LET STAFLG :B= #0          ;CLR START FLAG.
      ENDIF
      LET (R1) := #WTV           ;CMD3 - WRITE/ VERIFY.
      LET (R1) := #DATCNT        ;SET BRF TO MAX FOR PATTERN GENERATION.
      LET R2 := COMP #RNOP5C     ;31 OPERATIONS.
      LET (R1) := R2             ;RANDOM PATTERN.
      REPEAT                      ;REPEAT TO EOT:
      WHILE R1 LT #SEQEND DO     ;FILL SEQ TBL WITH RANDOM CMDS.
      LET RANS := RAN5 + RANB
      LET R2 := RANS CLR.BV #177741 ;R2 = RANDOM # (0 36).
      JSR PC,#RANCMD(R2)         ;SET UP A RANDOM CMD - BRF.
      ENDDO
      LET (R1) := #END           ;STORE END OF SEQUENCE CODE IN TABLE.
      JSR PC,EXALL              ;GO EXECUTE ALL CMDS IN SEQUENCE TABLE.

```

6740	023612			LET R1 := #CMDSEQ	; INIT CMD SEQ TBL POINTER,		
6741	023612	012701	003460		MOV	#CMDSEQ,	
6742	023616			UNTIL R2 NE #0	; REPEAT UNTIL EOT IS REACHED		
6743	023616	005702			TST	R2	
6744	023620	001753			BEQ	504433	
6745	023622			LET ALLEOT :B= ALLEOT + #1	; FLAG ALL UNITS @ EOT		
6746	023622	105237	003450		INCB	ALLEOT	
6747	023626	000240		NOP			
6748	023630	000240		NOP			
6749	023632	000240		NOP			
6750	023634	004737	025156	JSR PC,TSWEOT	; WRITE ONE RECORD BEYOND EOT ON ALL UNIT		
6751					; SO THAT SHORTER READ STOP DISTANCE		
6752					; SHALL POSITION HEAD IN CLEAN IRG GAP		
6753					; READ REV THAT EXTRA REC TO RE POSITION		
6754	023640	004737	023172	JSR PC,RANRD	; SET UP READ REV/FWD CMDS,		
6755	023644			LET CMDSEQ*4 := COMP #RNOPSC	; # OF RECORDS FOR READ REV.		
6756	023644	012737	177740		MOV	#RNOPSC,	
6757	023652	005137	003464		COM	CMDSEQ*4	
6758	023656			LET CMDSEQ*14 := CMDSEQ*4	; # OF RECORDS FOR READ FORWARD.		
6759	023656	013737	003464		MOV	CMDSEQ*4	
6760	023664			LET (R1) := #END	; STORE END OF SEQUENCE CODE IN SEQ TABLE		
6761	023664	012711	177777		MOV	#END,(R1	
6762	023670	004737	006510	JSR PC,EXALL	; GO EXECUTE READ REV/FWD OF LAST N RECOR		
6763	023674			LET ALLEOT :B= #0	; CLEAR ALL UNITS @ EOT FLAG		
6764	023674	105037	003450		CLRB	ALLEOT	
6765	023700			LET RPTFLG :B= #1	; REQUEST PERFORMANCE REPORT DURING REWIN		
6766	023700	112737	000001		MOVB	#1,RPTFL	
6767	023706			LET R1 := #CMDSEQ	; INIT SEQ TBL POINTER,		
6768	023706	012701	003460		MOV	#CMDSEQ,	
6769	023712	004737	006470	JSR PC,SETRW	; STORE REWIND IN SEQ TBL.		
6770	023716			LET (R1) := #END	; STORE END IN SEQ TBL.		
6771	023716	012711	177777		MOV	#END,(R1	
6772	023722	004737	006510	JSR PC,EXALL	; EXECUTE REWIND CMD ON ALL UNITS		
6773							
6774	023726			EXIT TST			
6775	023726	104432			TRAP	C\$EXIT	
6776	023730	000174			.WORD	L10034	
6777							

```

6778 ; ADDRESSES OF SUBROUTINES USED TO SET UP RANDOM OPERATIONS IN
6779 ; THE DATA RELIABILITY TEST.
6780
6781 023732 024060 RANCMD: RANWV ;WRITE/VERIFY.
6782 023734 024046 RANWR ;WRITE.
6783 023736 024046 RANWR ;WRITE.
6784 023740 024046 RANWR ;WRITE.
6785 023742 024046 RANWR ;WRITE.
6786 023744 024046 RANWR ;WRITE.
6787 023746 024046 RANWR ;WRITE.
6788 023750 024046 RANWR ;WRITE.
6789 023752 023772 RANRD ;READ.
6790 023754 023772 RANRD ;READ.
6791 023756 023772 RANRD ;READ.
6792 023760 023772 RANRD ;READ.
6793 023762 023772 RANRD ;READ.
6794 023764 023772 RANRD ;READ.
6795 023766 023772 RANRD ;READ.
6796 023770 023772 RANRD ;READ.
6797
6798
6799
6800
6801
6802 ; SUBROUTINE TO SET UP READ COMMANDS IN SEQUENCE TABLE.
6803 ; INPUTS:
6804 ; OUTPUTS:
6805 ; REGISTERS: R2
6806 ; CALLS:
6807
6808 023772 RANRD: LET (R1) := @RDR ;STORE READ REV CMD.
6809 023772 012721 104401 LET (R1) := @DATCNT ;SET BRF TO MAX FOR READ RANDOM LENGTHS.
6810 023776 012721 004000 LET RANB := RANB + RAN, ;MOV @RDR,(R1)
6811 023776 012721 004000 LET R2 := RANB CLR.BY @R1,OP5C ;MOV @DATCNT,
6812 024002 063737 003362 003360 ADD RANS,RAN ;ADD RANS,RAN
6813 024002 063737 003362 003360 LET R2 := RANB CLR.BY @R1,OP5C ;MOV RANB,R2
6814 024010 013702 003360 BIC @RNOPSC. ;BIC @RNOPSC.
6815 024014 042702 177740 LET (R1) := R2 ;SET RANDOM # OF OPERATIONS.
6816 024020 010221 LET (R1) := @RANP ;RANDOM PATTERN.
6817 024020 010221 LET (R1) := @RANP ;RANDOM PATTERN.
6818 024022 012721 000007 LET (R1) := @RDF ;STORE READ FWD CMD.
6819 024022 012721 104001 LET (R1) := @DATCNT ;SET BRF TO MAX TO READ RANDOM LENGTHS.
6820 024026 012721 104001 LET (R1) := R2 ;SET RANDOM # OF OPERATIONS.
6821 024026 012721 104001 LET (R1) := @RANP ;RANDOM PATTERN.
6822 024032 012721 004000 LET (R1) := @RANP ;RANDOM PATTERN.
6823 024032 012721 004000 LET (R1) := @RANP ;RANDOM PATTERN.
6824 024036 010221 LET (R1) := @RANP ;RANDOM PATTERN.
6825 024036 010221 LET (R1) := @RANP ;RANDOM PATTERN.
6826 024040 012721 000007 LET (R1) := @RANP ;RANDOM PATTERN.
6827 024040 012721 000007 LET (R1) := @RANP ;RANDOM PATTERN.
6828 024044 000207 RTS PC
6829 024044 000207
    
```

```

6830 ; SUBROUTINE TO SET UP A WRITE COMMAND IN THE SEQUENCE TABLE.
6831 ; INPUTS:
6832 ; OUTPUTS:
6833 ; REGISTERS:
6834 ; CALLS:
6835
6836 024046 RANWR: LET (R1)+ := @WRT ;STORE WRITE CMD.
6837 024046 012721 104005 ;STORE BRF, # OF OPERATIONS, PATTERN.
6838 024052 004737 024072 JSR PC,RANW
6839 024056 000207 RTS PC
6840
6841
6842
6843
6844
6845 ; SUBROUTINE TO SET UP A WRITE/VERIFY COMMAND IN THE SEQUENCE TABLE.
6846 ; INPUTS:
6847 ; OUTPUTS:
6848 ; REGISTERS:
6849 ; CALLS:
6850
6851 024060 RANWV: LET (R1)+ := @WTV ;STORE WRITE/VERIFY CMD.
6852 024060 012721 104105 ;STORE BRF, # OF OPERATIONS, PATTERN.
6853 024064 004737 024072 JSR PC,RANW
6854 024070 000207 RTS PC
6855
6856
6857

```

```

6858
6859
6860
6861
6862
6863
6864
6865
6866 024072
6867 024072 012721 004000
6868
6869 024076
6870 024076 063737 003362 003360
6871 024104
6872 024104 013702 003360
6873 024110 042702 177740
6874 024114
6875 024114 010221
6876 024116
6877 024116 012721 000007
6878 024122 000207
6879
6880
6881
6882 024124
6883 024124
6884 024124 104401
6885

; SUBROUTINE TO STORE BR# OF OPERATIONS, PATTERN IN COMMAND
; SEQUENCE TABLE FOR WRITE AND WRITE/VERIFY COMMANDS.
; INPUTS:
; OUTPUTS:
; REGISTERS: R2
; CALLS:
RANW: LET (R1) := #DATCNT ;SET BR# TO MAX FOR PATTERN GENERATION.
;RANDOM BR# WILL BE GENERATED FOR EACH R
MOV #DATCNT,
LET RANB := RANB + RANS ;RANDOM BR# WILL BE GENERATED FOR EACH R
ADD RANS,RAN
LET R2 := RANB CLR.BY #RNOPSC
MOV RANB,R2
BIC #RNOPSC,
LET (R1) : R2 ;SET RANDOM # OF OPERATIONS.
MOV R2,(R1)-
LET (R1) := #RANP ;RANDOM PATTERN.
MOV #RANP,(R
RTS PC ;RETURN.
.EVEN
ENDTST
L10034:
TRAP C$ETST

```

```

6886 .SBTTL TEST 3: WRITE COMPATABILITY/WRITE UTILITY.
6887
6888
6889 ;**
6890 ; TEST TO WRITE RECORDS FROM BOT TO EOT.
6891 ;
6892
6893 T3:: BGNTST
6894
6895 LET RANDOM :B= #1 ;SET THE RANDOM OPERATIONS FLAG.
6896 024126 112737 000001 003441 MOVB #1,RANDC
6897 024134 LET EXPBOT :B= #0 ;CLEAR EXPECT BOT FLAG.
6898 024134 105037 003440 CLRB EXPBOT
6899 024140 LET R2 := #DATCNT #1 ;SET UP THE RECORD LENGTH MASK.
6900 024140 012702 004000 MOV #DATCNT,
6901 024144 005302 DEC R2
6902 024146 LET LENMSK := COMP R2 ;ALLOW MAXIMUM BUFFER.
6903 024146 010237 003356 MOV R2,LENMS
6904 024152 005137 003356 COM LENMSK
6905 024156 004737 006444 JSR PC,SETCH ;CMD 1 = SET CHARACTERISTIC.
6906 024162 004737 006470 JSR PC,SETRW ;CMD2=REWIND
6907 024166 LET STAF LG :B= #0 ;CLEAR START FLAG
6908 024166 105037 003452 CLRB STAF LG
6909 024172 REPEAT ;REPEAT TO EOT.
6910 024172
6911 024172 WHILE R1 LT #SEQEND DO ;WHILE THERE IS MORE ROOM IN SEQ TABLE:
6912 024172
6913 024172 020127 003550
6914 024176 002003
6915 024200 004737 024046 JSR PC,RANWR ;STORE A WRITE CMD IN SEQUENCE TABLE.
6916 024204 ENDDO
6917 024204 000772 BR 50447$
6918 024206
6919 024206 LET (R1) := #END ;STORE END OF SEQUENCE CODE IN TABLE.
6920 024206 012711 177777 MOV #END,(R1
6921 024212 004737 006510 JSR PC,EXALL ;EXECUTE ALL CMDs IN SEQ TBL ON UNITS.
6922 024216 LET R1 := #CMDSEQ ;INIT SEQ TEL POINTER.
6923 024216 012701 003460 MOV #CMDSEQ,
6924 024222 UNTIL R2 NE #0 ;REPEAT UNTIL EOT IS REACHED
6925 024222 005702 TST R2
6926 024224 001762 BEQ 50446$
6927 024226 LET ALLEOT :B= ALLEOT + #1 ;SET ALL UNITS @ EOT FLAG
6928 024226 105237 003450 INCB ALLEOT
6929 024232 000240
6930 024234 000240
6931 024236 000240
6932 024240 004737 025156 JSR PC,TSWEOT ;WRITE ONE RECORD BEYOND EOT ON ALL UNIT
6933 ;SO THAT SHORTER READ STOP DISTANCE
6934 ;SHALL POSITION HEAD IN CLEAN IRG GAP
6935 ;READ REV THAT EXTRA REC TO RE POSITION
6936 024244 LET ALLEOT :B= #0 ;CLEAR ALL UNITS @ EOT FLAG
6937 024244 105037 003450 CLRB ALLEOT
6938 024250 004737 006470 JSR PC,SETRW ;STORE REWIND IN SEQ TBL.
6939 024254 LET (R1) := #END ;STORE END IN SEQ TBL.
6940 024254 012711 177777 MOV #END,(R1
6941 024260 004737 006510 JSR PC,EXALI ;EXECUTE REWIND CMD ON ALL UNITS

```

# H13

HARDWARE TESTS MAC111 30(1046) 06 APR 84 08:51 PAGE 165  
CZTSHD.P11 06 APR 84 08:49 TEST 3: WRITE COMPATABILITY/WRITE UTILITY.

SEQ 0163

6942							
6943	024264		EXIT	TST			
6944	024264	104432				TRAP	C\$EXIT
6945	024266	000002				.WORD	L10035-.
6946							
6947			.EVEN				
6948							
6949	024270		ENDTST				
6950	024270		L10035:				
6951	024270	104401				TRAP	C\$ETST
6952							



```

6953
6954
6955
6956
6957
6958
6959
6960 024272
6961 024272
6962
6963 024272
6964 024272 112737 000001 003441
6965 024300
6966 024300 112737 000001 003440
6967 024306 004737 006444
6968 024312 004737 006470
6969 024316
6970 024316 105037 003452
6971 024322
6972 024322 012721 104001
6973 024326
6974 024326 012721 004000
6975 024332
6976 024332 012721 077777
6977 024336
6978 024336 012721 000007
6979 024342
6980 024342 012711 177777
6981 024346 004737 006510
6982 024352
6983 024352 105237 003450
6984 024356
6985 024356 012701 003460
6986 024362
6987 024362 012721 104401
6988 024366
6989 024366 012721 004000
6990 024372
6991 024372 012721 077777
6992 024376
6993 024376 012721 000007
6994 024402
6995 024402 012711 177777
6996 024406 004737 006510
6997 024412
6998 024412 105037 003450
6999
7000 024416
7001 024416 104432
7002 024420 000002
7003
7004
7005
7006 024422
7007 024422
7008 024422 104401

.SBTTL TEST 4: READ COMPATABILITY/READ UTILITY.
;
; **
; TEST TO READ ENTIRE TAPE FORWARD AND REVERSE.
;

T4::
BGNTST
LET RANDOM :B= #1 ;SET THE RANDOM OPERATIONS FLAG.
MOV B #1,RAND0
LET EXPBOT :B= #1 ;SET EXPECT BOT FLAG.
MOV B #1,EXPBO
JSR PC,SETCH ;CMD 1 = SET CHARACTERISTIC.
JSR PC,SETRW ;CMD2=REWIND.
LET STAF LG :B= #0 ;CLEAR START FLAG
CLRB STAF LG
LET (R1) : = #RDF ;CMD3 = READ FORWARD.
MOV #RDF,(R1
LET (R1) : = #DATCNT ;SET LENGTH TO MAX FOR UNKNOWN LENGTHS.
MOV #DATCNT,
LET (R1) : = #77777 ;SET RECORD COUNT TO MAX FOR WHOLE TAPE.
MOV #77777,(
LET (R1) : = #RANP ;PATTERN = RANDOM.
MOV #RANP,(R
LET (R1) : = #END ;STORE END OF SEQUENCE CODE IN TABLE.
MOV #END,(R1
JSR PC,EXALL ;EXECUTE ALL CMDS IN SEQ TBL ON ALL UNIT
LET ALLEOT :B= ALLEOT * #1 ;FLAG TO ALLOW ALL UNITS AT EOT TO READ
INCB ALLEOT
LET R1 : = #CMDSEQ ;INIT CMD SEQ TBL POINTER.
MOV #CMDSEQ,
LET (R1) : = #RDR ;CMD1 = READ REVERSE.
MOV #RDR,(R1
LET (R1) : = #DATCNT ;SET LENGTH TO MAX FOR UNKNOWN LENGTHS.
MOV #DATCNT,
LET (R1) : = #77777 ;RECORD COUNT = MAX FOR WHOLE TAPE.
MOV #77777,(
LET (R1) : = #RANP ;PATTERN = RANDOM.
MOV #RANP,(R
LET (R1) : = #END ;STORE END OF SEQUENCE CODE IN TABLE.
MOV #END,(R1
JSR PC,EXALL ;GO EXECUTE READ REV. OF ENTIRE TAPE.
LET ALLEOT :B= #0 ;CLEAR ALL UNITS @ EOT FLAG
CLRB ALLEOT

EXIT TST
TRAP C$EXIT
.WORD L10036 .

.EVEN
ENDTST
L10036:
TRAP C$TST
    
```

```

7009 .SBTTL TEST 5: EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.
7010
7011 ;**
7012 ; TEST TO EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.
7013 ;
7014
7015 024424 BGNTST
7016 024424 T5::
7017
7018 024424 LET RANDOM :B = #0 ;CLEAR RANDOM MODE FLAG.
7019 024424 105037 003441 CLR B RANDOM
7020 024430 LET EXPBOT :B = #1 ;SET EXPECT BOT FLAG.
7021 024430 112737 000001 003440 MOV B #1,EXPBO
7022 024436 LET IRE :B = PIRE ;MOVE INHIBIT RFC ERROR REPORT FLAG.
7023 024436 113737 002214 003445 MOV B PIRE,IRE
7024 024444 004737 006444 JSR PC,SETCH ;CMD 1 = SET CHARACTERISTIC.
7025 024450 LET CMDSEQ+2 := CHAR ;MOVE CHAR CODE FROM P TBL TO SEQ TBL.
7026 024450 013737 002216 003462 MOV CHAR,CMD
7027 024456 LET R2 := #CMDD ;R2 POINTS TO CMD2 IN SOFT P TABLE.
7028 024456 012702 002220 MOV #CMDD,R2
7029 024462 004737 025134 JSR PC,PTCMDS ;MOVE CMD 2 FROM P TBL TO SEQ TBL.
7030 024466 004737 025134 JSR PC,PTCMDS ;MOVE CMD 3 FROM P TBL TO SEQ TBL.
7031 024472 004737 025134 JSR PC,PTCMDS ;MOVE CMD 4 FROM P TBL TO SEQ TBL.
7032 024476 004737 025134 JSR PC,PTCMDS ;MOVE CMD 5 FROM P TBL TO SEQ TBL.
7033 024502 004737 025134 JSR PC,PTCMDS ;MOVE CMD 6 FROM P TBL TO SEQ TBL.
7034 024506 004737 025134 JSR PC,PTCMDS ;MOVE CMD 7 FROM P TBL TO SEQ TBL.
7035 024512 004737 025134 JSR PC,PTCMDS ;MOVE END CMD FROM P TBL TO SEQ TBL.
7036 024516 LET JLOOP := #0 ;CLEAR JMP CMD LOOP COUNT.
7037 024516 005037 003370 CLR JLOOP
7038 024522 LET STAF LG :B = #0 ;CLEAR START FLAG
7039 024522 105037 003452 CLR B STAF LG
7040 024526 LET R1 := #CMDSEQ ;INIT SEQUENCE TABLE POINTER.
7041 024526 012701 003460 MOV #CMDSEQ,
7042 024532 3$: WHILE (R1) NE #END DO ;WHILE THERE ARE CMDS LEFT IN SEQUENCE
7043 024532 50451$:
7044 024532 021127 177777 CMP (R1),#EN
7045 024536 001574 BEQ 50452$
7046 024540 022711 000040 CMP #JMP.C,(R1) ;IS THIS A JUMP CMD?
7047 024544 001024 BNE 6$ ;BR IF NOT.
7048 024546 LET R1 := R1 + #2 ;POINT TO BRF.
7049 024546 062701 000002 ADD #2,R1
7050 024552 012137 0033'2 MOV (R1),JLOC ;SAVE BRF (LOCATION).
7051 024556 022137 003370 CMP (R1),JLOOP ;HAS LOOP COUNT BE SATISFIED?
7052 024562 001003 BNE 1$ ;IF NOT, JMP AGAIN.
7053 024564 LET R1 := R1 + #2 ;IF SO, ADJUST SEQ POUNTER
7054 024564 062701 000002 ADD #2,R1
7055 024570 000760 BR 3$ ;AND GO TO NEXT COMMAND.
7056 024572 1$: LET JLOOP := JLOOP + #1 ;UPDATE THE LOOP COUNT.
7057 024572 005237 003370 INC JLOOP
7058 024576 LET R1 := #CMDSEQ ;INIT CMD SEQ TABLE POINTER.
7059 024576 012701 003460 MOV #CMDSEQ,
7060 024602 005337 003372 2$: DEC JLOC ;DECR LOCATION COUNTER.
7061 024606 001751 BEQ 3$ ;IF THIS IS THE RIGHT LOCATION TO JMP TO
7062 024610 LET R1 := R1 + #10 ;IF NOT, UPDATE SEQ POINTER TO NEXT CMD.
7063 024610 062701 000010 ADD #10,R1
7064 024614 000772 BR 2$ ;DO IT AGAIN.
    
```

7065	024616	022711	000020	6\$:	CMP #DLY.C.(R1)	;DELAY?	
7066	024622	001026			BNE 4\$	;BR IF NOT.	
7067	024624				LET R1 := R1 + #4	;R1 = LOCATION OF N COUNT.	
7068	024624	062701	000004			ADD #4,R1	
7069	024630				LET TIME2 := (R1)	;SAVE N COUNT.	
7070	024630	011137	003366			MOV (R1),TIM	
7071	024634			7\$:	DELAY 1	;GO TO SUPER WAIT 1 MSEC.	
7072	024634	012727	000001			MOV #1,(PC).	
7073	024640	000000				.WORD 0	
7074	024642	013727	002116			MOV L\$DLY,(P	
7075	024646	000000				.WORD 0	
7076	024650	005367	177772			DEC 6(PC)	
7077	024654	001375				BNE -4	
7078	024656	005367	177756			DEC -22(PC)	
7079	024662	001367				BNE -20	
7080	024664	005337	003366		DEC TIME2		
7081	024670	001361			BNE 7\$		
7082	024672				LET R1 := R1 + #4	;POINT TO NEXT CMD.	
7083	024672	062701	000004			ADD #4,R1	
7084	024676	000715			BR 3\$	;GO CHECK NEXT CMD.	
7085	024700	004737	007452	4\$:	JSR PC,SETUP	;GO SETUP THE COMMAND BLOCK.	
7086	024704				WHILE NCNT LT NCNT1 DO	;WHILE THERE ARE RECORDS REMAINING:	
7087	024704					50453\$:	
7088	024704	023737	003340	003342		CMP NCNT,NCN	
7089	024712	002103				BGE 50454\$	
7090	024714	004737	007344		JSR PC,CMDAC	;STORE CMD ASCII IN ERROR MSG.	
7091	024720	004737	007004		JSR PC,EXSUB	;ISSUE CMD TO ALL,AWAIT INTS,CHECK STATU	
7092	024724				IF CMDWRD EQ #GES THEN	;IF CMD IS GET STATUS THEN:	
7093	024724	023727	003346	100017		CMP CMDWRD,#	
7094	024732	001002				BNE 50455\$	
7095	024734	004737	015776		JSR PC,PRXST	;PRINT EXTENDED STATUS REGISTERS.	
7096	024740				ENDIF		
7097	024740					50455\$:	
7098	024740	004737	016060		JSR PC,CKHAE	;CHECK HALT AFTER EACH CMD FLAG.	
7099	024744				LET R2 := #1	;SET ALL UNITS AT BOT/EOT.	
7100	024744	012702	000001			MOV #1,R2	
7101	024750	004737	015452		JSR PC,FIRSTU	;FIND FIRST UNIT.	
7102	024754				WHILE DEVTBL(R5) NE #END DO	;WHILE THERE ARE MORE UNITS:	
7103	024754					50456\$:	
7104	024754	026527	002532	177777		CMP DEVTBL(R	
7105	024762	001426				BEQ 50457\$	
7106	024764				IF #MOD.CO SETIN CMDWRD THEN	;IF CMD IS REVERSE THEN:	
7107	024764	032737	000400	003346		BIT #MOD.CO.	
7108	024772	001406				BEQ 50460\$	
7109	024774				IF #X0.BOT NOTSETIN EOTFLG(R5) THEN	;IF NOT AT BOT THEN:	
7110	024774	032765	000002	003426		BIT #X0.BOT.	
7111	025002	001001				BNE 50461\$	
7112	025004				LET R2 := #0	;CLEAR EOT/BOT FLAG.	
7113	025004	005002				CLR R2	
7114	025006				ENDIF		
7115	025006					50461\$:	
7116	025006				ELSE	;ELSE IF CMD IS NOT REVERSE:	
7117	025006	000411				BR 50462\$	
7118	025010					50460\$:	
7119	025010				IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN		
7120	025010	032765	000001	003426		BIT #X0.EOT.	



```

7177
7178
7179
7180
7181
7182
7183
7184
7185
7186 025134
7187 025134 012203
7188 025136 005303
7189 025140 006303
7190 025142
7191 025142 016321 003562
7192 025146
7193 025146 012221
7194 025150
7195 025150 012221
7196 025152
7197 025152 012221
7198 025154 000207

; SUBROUTINE TO MOVE A COMMAND FROM THE SOFTWARE P TABLE TO
; THE COMMAND SEQUENCE TABLE.
; INPUTS: R2 = POINTER TO SOFT 'P' TABLE
; OUTPUTS:
; REGISTERS: R3.
; CALLS:
PTCMDS: LET R3 := (R2)+ #1 SHIFT +1 ;R3 = COMMAND TABLE INDEX.
; MOV (R2)+,R3
; DEC R3
; ASL R3
LET (R1)+ := CMDTBL(R3) ;MOVE COMMAND WORD.
; MOV CMDTBL(R
LET (R1)+ := (R2)+ ;MOVE # OF BYTES.
; MOV (R2)+,(R
LET (R1)+ := (R2)+ ;MOVE # OF OPERATIONS.
; MOV (R2)+,(P
LET (R1)+ := (R2)+ ;MOVE PATTERN CODE.
; MOV (R2)+,(R
RTS PC
    
```

```

7199
7200      :      SUBROUTINE TO WRITE THEN READ REVERSE ONE RECORD BEYOND EOT
7201      :      INPUTS:
7202      :      OUTPUTS:
7203      :      REGISTERS:
7204      :      CALLS:          CMDAC,EXSUB,CKHAE
7205
7206 025156 000240      TSWEOT: NOP
7207 025160 000240      NOP
7208 025162 004737 007004      JSR PC,EXSUB          ;WRITE ONE RECORD BEYOND EOT
7209 025166 004737 016060      JSR PC,CKHAE          ;SO THAT READ SHORTER STOP DISTANCE
7210                                     ;SHALL POSITION HEAD IN CLEAN IRG GAP
7211 025172                                     ;REPOSITION TAPE
7212 025172 013737 003346 003352      LET CMDWRD := CMDWRD          MOV      CMDWRD,P
7213 025200                                     ;BEFORE EXTRA RECORD
7214 025200 012737 104401 003346      LET CMDWRD := #RDR          MOV      #RDR,CMD
7215 025206                                     ;BY READING REVERSE
7216 025206 012737 000004 003354      LET CMDLG := #4           MOV      #4,CMDLG
7217 025214                                     ;THAT RECORD TO ALLOW
7218 025214 013737 003346 002310      LET CMDPKT := CMDWRD CLR.BY #BRF.C      MOV      CMDWRD,C
7219 025222 042737 004000 002310      BIC      #BRF.C,C
7220 025230                                     ;NEXT COMMAND IN THE
7221 025230 013737 002310 003350      LET CMDSAV := CMDPKT      MOV      CMDPKT,C
7222 025236                                     ;TABLE TO BE EXECUTED
7223 025236 013737 003336 002312      LET CMDPKT+CP.ADL :- DATARD      MOV      DATARD,C
7224 025244 004737 007344      JSR PC,CMDAC
7225 025250 004737 007004      JSR PC,EXSUB
7226 025254 004737 016060      JSR PC,CKHAE
7227 025260 000207      RTS PC
7228
7229      .EVEN
7230
7231      ENDTST
7232 025262      L10037:
7233 025262      TRAP      C$ETST
7234 025262 104401
7235
7236 025264      ENDMOD
    
```

```

7237 .TITLE PARAMETER CODING
7238
7239 .SETTL HARDWARE PARAMETER CODING SECTION
7240
7241 025264 BGNMOD
7242
7243
7244 ;
7245 ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
7246 ; THAT ARE USED BY THE SUPERVISOR TO BUILD P TABLES. THE
7247 ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
7248 ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
7249 ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
7250 ; WITH THE OPERATOR.
7251 ;
7252 025264 BGNHRD
7253 025264 000024
7254 025266 LSHARD:: .WORD L10040 LSH-
7255
7256 025266 GPRMA TS4ADR,0,0,16000,177564,YES
7257 025266 000031 .WORD TSCODE
7258 025270 025312 .WORD TS4ADR
7259 025272 160002 .WORD T$LOLIM
7260 025274 177564 .WORD T$MILIM
7261 025276 GPRMD TS4VCT,2,0,777,60,776,YES
7262 025276 001032 .WORD TSCODE
7263 025300 025327 .WORD TS4VCT
7264 025302 000777 .WORD 777
7265 025304 000060 .WORD T$LOLIM
7266 025306 000776 .WORD T$MILIM
7267
7268 025310 EXIT HRD
7269 025310 013004 .WORD TSCODE
7270
7271 025312 051524 051123 040440 TS4ADR: .MLIST BEX
025327 126 041505 047524 TS4VCT: .ASCIZ /TSSR ADDRESS,
.LIST BEX
.EVEN
7272
7273
7274 025336 ENDMRD
7275
7276 025336 L10040: .EVEN

```

7277  
7278  
7279  
7280  
7281  
7282  
7283  
7284  
7285  
7286  
7287  
7288 025336  
7289 025336 000501  
7290 025340  
7291  
7292  
7293 025340  
7294 025340 000130  
7295 025342 026102  
7296 025344 000001  
7297 025346  
7298 025346 000130  
7299 025350 026121  
7300 025352 000400  
7301 025354  
7302 025354 001130  
7303 025356 026150  
7304 025360 000001  
7305 025362  
7306 025362 001130  
7307 025364 026174  
7308 025366 000400  
7309 025370  
7310 025370 002130  
7311 025372 026225  
7312 025374 000001  
7313 025376  
7314 025376 004024  
7315 025400  
7316 025400 002130  
7317 025402 026246  
7318 025404 000400  
7319 025406  
7320 025406 003130  
7321 025410 026276  
7322 025412 000001  
7323 025414  
7324 025414 003130  
7325 025416 026321  
7326 025420 000400  
7327 025422  
7328 025422 004130  
7329 025424 026352  
7330 025426 000001  
7331 025430  
7332 025430 127044

.SBTTL SOFTWARE PARAMETER CODING SECTION

\*\*\*  
; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS  
; THAT ARE USED BY THE SUPERVISOR TO BUILD P TABLES. THE  
; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE  
; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE  
; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS  
; WITH THE OPERATOR.  
;

BGNST  
L\$SOFT::  
GPRML CLRM,0,1,YES  
GPRML HRVM,0,400,YES  
GPRML HAEM,2,1,YES  
GPRML RCVERM,2,400,YES  
GPRML IRECM,4,1,YES  
XFERT NEXTSP  
GPRML BADTM,4,400,YES  
NEXTSP: GPRML DINTM,6,1,YES  
GPRML IREM,6,400,YES  
GPRML CHGM,10,1,YES  
XFERR ENDSP1  
.WORD L1004: L\$S  
.WORD T\$CODE  
.WORD CLRM  
.WORD 1  
.WORD T\$CODE  
.WORD RRVM  
.WORD 400  
.WORD T\$CODE  
.WORD HAEM  
.WORD 1  
.WORD T\$CODE  
.WORD RCVERM  
.WORD 400  
.WORD T\$CODE  
.WORD IRECM  
.WORD 1  
.WORD T\$CODE  
.WORD BADTM  
.WORD 400  
.WORD T\$CODE  
.WORD DINTM  
.WORD 1  
.WORD T\$CODE  
.WORD IREM  
.WORD 400  
.WORD T\$CODE  
.WORD CHGM  
.WORD 1  
.WORD T\$CODE



7333	025432		GPRMD	CHARM,14,0,377,0,777,YES		
7334	025432	006032			.WORD	T\$CODE
7335	025434	026376			.WORD	CHARM
7336	025436	000377			.WORD	377
7337	025440	000000			.WORD	T\$LOLIM
7338	025442	000777			.WORD	T\$HILIM
7339	025444		GPRMD	CMD2M,16,D,37,1,33,YES		
7340	025444	007052			.WORD	T\$CODE
7341	025446	026423			.WORD	CMD2M
7342	025450	000037			.WORD	37
7343	025452	000001			.WORD	T\$LOLIM
7344	025454	000033			.WORD	T\$HILIM
7345	025456		GPRMD	BPCRM,20,D,-1,1,0ATCNT,YES		
7346	025456	010052			.WORD	T\$CODE
7347	025460	026431			.WORD	BPCRM
7348	025462	177777			.WORD	-1
7349	025464	000001			.WORD	T\$LOLIM
7350	025466	004000			.WORD	T\$HILIM
7351	025470		GPRMD	NUMBM,22,D,-1,1,77777,YES		
7352	025470	011052			.WORD	T\$CODE
7353	025472	026443			.WORD	NUMBM
7354	025474	177777			.WORD	1
7355	025476	000001			.WORD	T\$LOLIM
7356	025500	077777			.WORD	T\$HILIM
7357	025502		GPRMD	PATTM,24,D,17,0,10,YES		
7358	025502	012052			.WORD	T\$CODE
7359	025504	026463			.WORD	PATTM
7360	025506	000017			.WORD	17
7361	025510	000000			.WORD	T\$LOLIM
7362	025512	000010			.WORD	T\$HILIM
7363	025514		GPRMD	CMD3M,26,D,37,1,33,YES		
7364	025514	013052			.WORD	T\$CODE
7365	025516	026476			.WORD	CMD3M
7366	025520	000037			.WORD	37
7367	025522	000001			.WORD	T\$LOLIM
7368	025524	000033			.WORD	T\$HILIM
7369	025526		GPRMD	BPCRM,30,D,-1,1,0ATCNT,YES		
7370	025526	014052			.WORD	T\$CODE
7371	025530	026431			.WORD	BPCRM
7372	025532	177777			.WORD	1
7373	025534	000001			.WORD	T\$LOLIM
7374	025536	004000			.WORD	T\$HILIM
7375	025540		GPRMD	NUMBM,32,D,-1,1,77777,YES		
7376	025540	015052			.WORD	T\$CODE
7377	025542	026443			.WORD	NUMBM
7378	025544	177777			.WORD	1
7379	025546	000001			.WORD	T\$LOLIM
7380	025550	077777			.WORD	T\$HILIM
7381	025552		GPRMD	PATTM,34,D,17,0,10,YES		
7382	025552	016052			.WORD	T\$CODE
7383	025554	026463			.WORD	PATTM
7384	025556	000017			.WORD	17
7385	025560	000000			.WORD	T\$LOLIM
7386	025562	000010			.WORD	T\$HILIM
7387	025564		GPRMD	CMD4M,36,D,37,1,33,YES		
7388	025564	017052			.WORD	T\$CODE

7389 025566 026504  
7390 025570 000037  
7391 025572 000001  
7392 025574 000033  
7393 025576  
7394 025576 020052  
7395 025600 026431  
7396 025602 177777  
7397 025604 000001  
7398 025606 004000  
7399 025610  
7400 025610 021052  
7401 025612 026443  
7402 025614 177777  
7403 025616 000001  
7404 025620 077777  
7405 025622  
7406 025622 022052  
7407 025624 026463  
7408 025626 000017  
7409 025630 000000  
7410 025632 000010  
7411 025634  
7412 025634 023052  
7413 025636 026512  
7414 025640 000037  
7415 025642 000001  
7416 025644 000033  
7417 025646  
7418 025646 024052  
7419 025650 026431  
7420 025652 177777  
7421 025654 000001  
7422 025656 004000  
7423 025660  
7424 025660 025052  
7425 025662 026443  
7426 025664 177777  
7427 025666 000001  
7428 025670 077777  
7429 025672  
7430 025672 026052  
7431 025674 026463  
7432 025676 000017  
7433 025700 000000  
7434 025702 000010  
7435 025704  
7436 025704 002004  
7437 025706  
7438 025706 075004  
7439 025710  
7440 025710 027052  
7441 025712 026520  
7442 025714 000037  
7443 025716 000001  
7444 025720 000033

GPRMD BPCRM,40,D,-1,1,DATCNT,YES

GPRMD NUMBM,42,D,1,1,77777,YES

GPRMD PATTM,44,D,17,0,10,YES

GPRMD CMD5M,46,D,37,1,33,YES

GPRMD BPCRM,50,D,1,1,DATCNT,YES

GPRMD NUMBM,52,D,1,1,77777,YES

GPRMD PATTM,54,D,17,0,10,YES

XFER ENDSP2

ENDSP1: XFER ENDSP

ENDSP2: GPRMD CMD6M,56,D,37,1,33,YES

.WORD CMD4M  
.WORD 37  
.WORD T\$LOLIM  
.WORD T\$HILIM  
.WORD T\$CODE  
.WORD BPCRM  
.WORD 1  
.WORD T\$LOLIM  
.WORD T\$HILIM  
.WORD T\$CODE  
.WORD NUMBM  
.WORD -1  
.WORD T\$LOLIM  
.WORD T\$HILIM  
.WORD T\$CODE  
.WORD PATTM  
.WORD 17  
.WORD T\$LOLIM  
.WORD T\$HILIM  
.WORD T\$CODE  
.WORD CMD5M  
.WORD 37  
.WORD T\$LOLIM  
.WORD T\$HILIM  
.WORD T\$CODE  
.WORD BPCRM  
.WORD 1  
.WORD T\$LOLIM  
.WORD T\$HILIM  
.WORD T\$CODE  
.WORD NUMBM  
.WORD 1  
.WORD T\$LOLIM  
.WORD T\$HILIM  
.WORD T\$CODE  
.WORD PATTM  
.WORD 17  
.WORD T\$LOLIM  
.WORD T\$HILIM  
.WORD T\$CODE  
.WORD XFER  
.WORD T\$CODE  
.WORD T\$CODE  
.WORD T\$CODE  
.WORD CMD6M  
.WORD 37  
.WORD T\$LOLIM  
.WORD T\$HILIM

7445	025722		GPRMD	BPCRM,60,D, 1,1,DATCNT,YES		
7446	025722	030052			.WORD	T\$CODE
7447	025724	026431			.WORD	BPCRM
7448	025726	177777			.WORD	1
7449	025730	000001			.WORD	T\$LOLIM
7450	025732	004000			.WORD	T\$HILIM
7451	025734		GPRMD	NUMBM,62,D,-1,1,77777,YES		
7452	025734	031052			.WORD	T\$CODE
7453	025736	026443			.WORD	NUMBM
7454	025740	177777			.WORD	1
7455	025742	000001			.WORD	T\$LOLIM
7456	025744	077777			.WORD	T\$HILIM
7457	025746		GPRMD	PATTM,64,D,17,0,10,YES		
7458	025746	032052			.WORD	T\$CODE
7459	025750	026463			.WORD	PATTM
7460	025752	000017			.WORD	17
7461	025754	000000			.WORD	T\$LOLIM
7462	025756	000010			.WORD	T\$HILIM
7463	025760		GPRMD	CMD7M,66,D,37,1,33,YES		
7464	025760	033052			.WORD	T\$CODE
7465	025762	026526			.WORD	CMD7M
7466	025764	000037			.WORD	37
7467	025766	000001			.WORD	T\$LOLIM
7468	025770	000033			.WORD	T\$HILIM
7469	025772		GPRMD	BPCRM,70,D, 1,1,DATCNT,YES		
7470	025772	034052			.WORD	T\$CODE
7471	025774	026431			.WORD	BPCRM
7472	025776	177777			.WORD	-1
7473	026000	000001			.WORD	T\$LOLIM
7474	026002	004000			.WORD	T\$HILIM
7475	026004		GPRMD	NUMBM,72,D, 1,1,77777,YES		
7476	026004	035052			.WORD	T\$CODE
7477	026006	026443			.WORD	NUMBM
7478	026010	177777			.WORD	-1
7479	026012	000001			.WORD	T\$LOLIM
7480	026014	077777			.WORD	T\$HILIM
7481	026016		GPRMD	PATTM,74,D,17,0,10,YES		
7482	026016	036052			.WORD	T\$CODE
7483	026020	026463			.WORD	PATTM
7484	026022	000017			.WORD	17
7485	026024	000000			.WORD	T\$LOLIM
7486	026026	000010			.WORD	T\$HILIM
7487	026030		GPRMD	CMD8M,76,D,37,1,33,YES		
7488	026030	037052			.WORD	T\$CODE
7489	026032	026534			.WORD	CMD8M
7490	026034	000037			.WORD	37
7491	026036	000001			.WORD	T\$LOLIM
7492	026040	000033			.WORD	T\$HILIM
7493	026042		GPRMD	BPCRM,100,D,-1,1,DATCNT,YES		
7494	026042	040052			.WORD	T\$CODE
7495	026044	026431			.WORD	BPCRM
7496	026046	177777			.WORD	-1
7497	026050	000001			.WORD	T\$LOLIM
7498	026052	004000			.WORD	T\$HILIM
7499	026054		GPRMD	NUMBM,102,D, 1,1,77777,YES		
7500	026054	041052			.WORD	T\$CODE

7501 026056 026443  
7502 026060 177777  
7503 026062 000001  
7504 026064 077777  
7505 026066  
7506 026066 042052  
7507 026070 026463  
7508 026072 000017  
7509 026074 000000  
7510 026076 000010  
7511 026100  
7512 026100  
7513 026100 176004

GPRMD PATM,104,D,17,0,10,YES

ENDSP: XFER JMPMSG

.WORD NUMBM  
.WORD -1  
.WORD T\$LOLIM  
.WORD T\$HILIM  
  
.WORD T\$CODE  
.WORD PATM  
.WORD 17  
.WORD T\$LOLIM  
.WORD T\$HILIM  
  
.WORD T\$CODE

7514  
7515  
026102 046103 040505 020122  
026121 122 051505 052105  
026150 040510 052114 040440  
026174 051120 047111 020124  
026225 111 044116 041111  
026246 040502 020104 040524  
026276 044504 040523 046102  
026321 111 044116 041111  
026352 044103 047101 042507  
026376 044103 051101 041501  
026423 103 042115 031057  
026431 102 043122 041440  
026443 043 047440 020106  
026463 120 052101 042524

.NLIST BEX  
CLRM: .ASCIZ /CLEAR COUNTERS/  
RRVM: .ASCIZ /RESET RANDOM VARIABLES/  
HAEM: .ASCIZ /HALT AFTER EACH CMD/  
RCVERM: .ASCIZ /PRINT RECOVERABLE ERRORS/  
IRECM: .ASCIZ /INHIBIT RECOVERY/  
BADTM: .ASCIZ /BAD TAPE SPOT DETECTION/  
DINTM: .ASCIZ /DISABLE INTERRUPTS/  
IREM: .ASCIZ /INHIBIT RFC ERROR REPORT/  
CHGM: .ASCIZ /CHANGE CMD SEQUENCE/  
CHARM: .ASCIZ /CHARACTERISTICS CODE/  
CMD2M: .ASCIZ "CMD/2"  
BPCRM: .ASCIZ /BRF COUNT/  
NUMBM: .ASCIZ /# OF OPERATIONS/  
PATTM: .ASCIZ /PATTERN/  
.LIST BEX  
.EVEN

7516 026474

7517  
7518 026474  
7519 026474  
7520 026474 023004

JMPMSG: EXIT SFT

.WORD T\$CODE

7521  
7522  
026476 046503 027504 000063  
026504 046503 027504 000064  
026512 046503 027504 000065  
026520 046503 027504 000066  
026526 046503 027504 000067  
026534 046503 027504 000070

.NLIST BEX  
CMD3M: .ASCIZ "CMD/3"  
CMD4M: .ASCIZ "CMD/4"  
CMD5M: .ASCIZ "CMD/5"  
CMD6M: .ASCIZ "CMD/6"  
CMD7M: .ASCIZ "CMD/7"  
CMD8M: .ASCIZ "CMD/8"  
.LIST BEX  
.EVEN

7523  
7524  
7525 026542  
7526  
7527 026542

ENDSFT

.EVEN

7528  
7529  
7530  
7531  
7532

L1004i:

;\*\*\*\*\*  
;\*\*\*\*\*  
; PATCH AREA

7533 026542 000100  
7534  
7535  
7536

PATCH: .BLKW 64.

;\*\*\*\*\*  
;\*\*\*\*\*

7537 026742  
7538  
7539 026742 026756  
7540 026744 000004  
7541 026746  
7542 026746

LASTAD

.EVEN  
.WORD T\$FREE  
.WORD T\$SIZE

L\$LAST: ENDMOD

7543  
7544  
7545  
7546  
7547  
7548  
7549 026746  
7550 026746  
7551 026746 000000  
7552 026750 000002  
7553 026752  
7554 026752 172522  
7555 026754 000224  
7556 026756  
7557 026756  
7558 026756  
7559  
7560 000001

.SBTTL HARD CODED P TBL  
:  
:DIAG IS PRE PARAMETERIZED PER TBL  
:  
BGNSETUP 1  
BGNPTAB  
L10042:  
172522  
224  
ENDPTAB  
L10044:  
ENDSETUP  
.  
.  
.  
.END

.WORD Q  
.WORD L10044 .

ACK.C = 100000 G  
 ADR = 000020 G  
 ALLEOT 003450 G  
 ASSEMB = 000010  
 ATTNM 004335 G  
 AUDRPM 004645 G  
 AUTODM 021522  
 BADTM 026246  
 BADTSW 002206 G  
 BFSEQ 022676  
 BFSEQ0 022722  
 BFSEQ1 022774  
 BFSEQ2 023006  
 BFSEQ3 023100  
 BFSEQ4 023152  
 BFSEQ5 023214  
 BFSEQ6 023266  
 BFSEQ7 023320  
 BFSEQ8 023352  
 BFSEQ9 023404  
 BFSE10 023426  
 BGNFLG = 003404  
 BINC 014366  
 BIT0 = 000001 G  
 BIT00 = 000001 G  
 BIT01 = 000002 G  
 BIT02 = 000004 G  
 BIT03 = 000010 G  
 BIT04 = 000020 G  
 BIT05 = 000040 G  
 BIT06 = 000100 G  
 BIT07 = 000200 G  
 BIT08 = 000400 G  
 BIT09 = 001000 G  
 BIT1 = 000002 G  
 BIT10 = 002000 G  
 BIT11 = 004000 G  
 BIT12 = 010000 G  
 BIT13 = 020000 G  
 BIT14 = 040000 G  
 BIT15 = 100000 G  
 BIT2 = 000004 G  
 BIT3 = 000010 G  
 BIT4 = 000020 G  
 BIT5 = 000040 G  
 BIT6 = 000100 G  
 BIT7 = 000200 G  
 BIT8 = 000400 G  
 BIT9 = 001000 G  
 BOE = 000400 G  
 BORERS 013576 G  
 BPCRM 026431  
 BRCPK 002324 G

BRCNT 003344 G  
 BRFC = 004000 G  
 BTADDR 002544 G  
 BTMSG1 013372  
 BTMSG2 013457  
 BTMSG3 013527  
 BTPT 003436 G  
 BTRPT 016544  
 BTO 002774 G  
 BT1 003046 G  
 BT2 003120 G  
 BT3 003172 G  
 CHAR 002216 G  
 CHARM 026376  
 CHGFLG 002212 G  
 CHGM 026352  
 CHKERR 011456 G  
 CH.EAI = 000040 G  
 CH.ERI = 000020 G  
 CH.ESS = 000200 G  
 CKDATA 015036 G  
 CKDCNT 015446  
 CKDFF 015450  
 CKHAE 016060 G  
 CKHRTN 016146  
 CLN = 101012 G  
 CLRERR 011154 G  
 CLRFLG 002202 G  
 CLRM 026102  
 CMDAC 007344 G  
 CMDASC 003650 G  
 CMD 002220 G  
 CMDLG 003354 G  
 CMDPKM 004102 G  
 CMDPKT 002310 G  
 CMDSAV 003350 G  
 CMDSEQ 003460 G  
 CMDSE2 003470 G  
 CMDTBL 003562 G  
 CMDWRD 003346 G  
 CMD.CO = 000001 G  
 CMD.C1 = 000002 G  
 CMD.C2 = 000004 G  
 CMD.C3 = 000010 G  
 CMD.C4 = 000020 G  
 CMD2M 026423  
 CMD3M 026476  
 CMD4M 026504  
 CMD5M 026512  
 CMD6M 026520  
 CMD7M 026526  
 CMD8M 026534  
 CNTBGN = 002554

CNTEND = 003324  
 CNTLEN = 000550 G  
 CODELM 003772 G  
 COUNT = 050404  
 CP.ADH = 000004 G  
 CP.ADL = 000002 G  
 CP.CMD = 000000 G  
 CP.CNT = 000006 G  
 CRLF 005213 G  
 CRLFSP 005216 G  
 CTCC 003376 G  
 CVC.C = 040000 G  
 C\$AU = 000052  
 C\$AUTO = 000061  
 C\$BRK = 000022  
 C\$BSEG = 000004  
 C\$BSUB = 000002 G  
 C\$CEFG = 000045  
 C\$CLCK = 000062  
 C\$CLEA = 000012  
 C\$CLOS = 000035  
 C\$CLP1 = 000006  
 C\$CVEC = 000036  
 C\$DCLN = 000044  
 C\$DODU = 000051  
 C\$DRPT = 000024  
 C\$DU = 000053  
 C\$EDIT = 000003  
 C\$ERDF = 000055  
 C\$ERHR = 000056  
 C\$ERRO = 000060  
 C\$ERSF = 000054  
 C\$ERSO = 000057  
 C\$ESCA = 000010  
 C\$ESEG = 000005  
 C\$ESUB = 000003  
 C\$ETST = 000001  
 C\$EXIT = 000032  
 C\$GETB = 000026  
 C\$GETW = 000027  
 C\$GMAN = 000043  
 C\$GPHR = 000042  
 C\$GPLO = 000030  
 C\$GPRI = 000040  
 C\$INIT = 000011  
 C\$INLP = 000020  
 C\$MANI 000050  
 C\$MEM = 000031  
 C\$MSG = 000023  
 C\$OPEN = 000034  
 C\$PNTB = 000014  
 C\$PNTF = 000017  
 C\$PNTS = 000016

C\$PNTX = 000015  
 C\$QIO = 000377  
 C\$RDBU = 000007  
 C\$REFG = 000047  
 C\$RESE = 000033  
 C\$REVI = 000003  
 C\$RFLA = 000021  
 C\$RPT = 000025  
 C\$SEFG = 000046  
 C\$SPRI = 000041  
 C\$SVEC = 000037  
 C\$TPRI = 000013  
 DATARD 003336 G  
 DATAW 003334 G  
 DATCNT = 004000 G  
 DEVTBL 002532 G  
 DFPTBL 002174 G  
 DFTSCH = 000040 G  
 DIA = 100006 G  
 DIABLK = 003334 G  
 DIACNT = 000020 G  
 DIAGMC = 000000  
 DINT 002210 G  
 DINTM 026276  
 DLY = 000020 G  
 DLY.C = 000020 G  
 DRI = 100013 G  
 DROPDM 004616 G  
 DROPEL 003446 G  
 DROPN 015774  
 DROPU 015554 G  
 DROPUA 015704  
 DRORTN 015762  
 DTAERM 005224 G  
 DTAER2 004677 G  
 DTAER3 004746 G  
 DTAER4 005010 G  
 DTAER5 005031 G  
 EF.CON = 000036 G  
 EF.NEW = 000035 G  
 EF.PWR = 000034 G  
 EF.RES = 000037 G  
 EF.STA = 000040 G  
 EINC 014374  
 END = 177777 G  
 ENDERF = 003416  
 ENDFLG = 003452  
 ENDSP 026100  
 ENDSP1 025706  
 ENDSP2 025710  
 EOTFLG 003426 G  
 ERVER 002205 G  
 ERLOG 003412 G

ERRREC 003415 G  
 ERS = 100411 G  
 ERSFLG 003451 G  
 FVL = 000004 G  
 EXALL 006510 G  
 EXARTN 007002  
 EXCRTN 010634  
 EXCUTE 010326 G  
 EXPBOT 003440 G  
 EXSUB 007004 G  
 E\$END = 002100  
 E\$LOAD = 000035  
 FATSM 004373 G  
 FIRSTU 015452 G  
 FMT.CO = 000040 G  
 FMT.C1 = 000100 G  
 FTLCNT 003314 G  
 FUNRM 004353 G  
 F\$AU = 000015  
 F\$AUTO = 000020  
 F\$BGN = 000040  
 F\$CLEA = 000007  
 F\$DU = 000016  
 F\$END = 000041  
 F\$HARD = 000004  
 F\$HW = 000013  
 F\$INIT = 000006  
 F\$JMP = 000050  
 F\$MOD = 000000  
 F\$MSG = 000011  
 F\$PROT = 000021  
 F\$PWR = 000017  
 F\$RPT = 000012  
 F\$SEG = 000003  
 F\$SOFT = 000005  
 F\$SRV = 000010  
 F\$SUB = 000002  
 F\$SW = 000014  
 F\$TEST = 000001  
 GCMDA 007416 G  
 GENPAT 010030 G  
 GES = 100017 G  
 GETSTM 005157 G  
 GIT 010322  
 GOWAIT 010636 G  
 GSCPK 002320 G  
 G\$CNTD = 000200  
 G\$DELM = 000372  
 G\$DISP = 000003  
 G\$EXCP = 000400  
 G\$HILI = 000002  
 G\$LOLI = 000001  
 G\$NO = 000000

G\$OFFS= 000400	JMP = 000040 G	L\$SPC 002056 G	MSGPKT 002334 G	PRI01 = 000040 G
G\$OFSI= 000376	JMPMSG 026474	L\$SPCP 002020 G	MSGPK0 002352 G	PRI02 = 000100 G
G\$PRMA= 000001	JMP.C = 000040 G	L\$SPTP 002024 G	MSGPK1 002370 G	PRI03 = 000140 G
G\$PRMD= 000002	J\$JMP = 000167	L\$STA 002030 G	MSGPK2 002406 G	PRI04 = 000200 G
G\$PRML= 000000	LENMSK 003356 G	L\$SW 002202 G	MSGPK3 002424 G	PRI05 = 000240 G
G\$RADA= 000140	LOE = 040000 G	L\$TEST 002114 G	MS.RFC= 000004 G	PRI06 = 000300 G
G\$RADB= 000000	LOG 014102 G	L\$TIML 002014 G	MS.XS0= 000006 G	PRI07 = 000340 G
G\$RADD= 000040	LOT = 000010 G	L\$UNIT 002012 G	MS.XS1= 000010 G	PRXST 015776 G
G\$RADL= 000120	L\$ACP 002110 G	L10000 002200	MS.XS2= 000012 G	PTCMDS 025134
G\$RADO= 000020	L\$APT 002036 G	L10001 002310	MS.XS3= 000014 G	PWRFLG 003453 G
G\$XFER= 000004	L\$AU 021770 G	L10002 005370	NCMD.C= 177740 G	RANB 003360 G
G\$YES = 000010	L\$AUT 002070 G	L10003 006314	NCNT 003340 G	RANBC = 153624 G
HAE 002204 G	L\$AUTO 021232 G	L10004 006322	NCNT1 003342 G	RANCMD 023732
HAEM 026150	L\$CCP 002106 G	L10005 006330	NEXTSP 025406	RANDOM 003441 G
HALTM 004042 G	L\$CLEA 021654 G	L10006 006336	NEXTU 015520 G	RANP = 000007 G
HELP = 000000	L\$CO 002032 G	L10007 006344	NINUSE= 177774 G	RANRD 023772
HOE = 100000 G	L\$DEPO 002011 G	L10010 017674	NOINTM 004421 G	RANS 003362 G
HRDCNT 003304 G	L\$DESC 002136 G	L10012 021230	NRDM 021616	RANSC = 032561 G
IBE = 010000 G	L\$DESP 002076 G	L10013 021520	NSSRM 004271 G	RANW 024072
IDU = 000040 G	L\$DEVP 002060 G	L10014 021714	NUMBM 026443	RANWR 024046
IER = 020000 G	L\$DISP 002124 G	L10015 021766	NURTY1 005073 G	PANWV 024060
IE.C = 000200 G	L\$DLY 002116 G	L10016 022062	OFLINM 005127 G	RCVERM 026174
INIT10 017704	L\$DTP 002040 G	L10017 023450	ONEFIL= 000001	RDF = 104001 G
INIT15 020132	L\$DTYP 002034 G	L10020 022216	OPFLAG 003456 G	RDR = 104401 G
INIT16 020152	L\$DU 021716 G	L10021 022242	OPP.C = 020000 G	RECCNT 003324 G
INTFLG 003416 G	L\$DUT 002072 G	L10022 022262	O\$APTS= 000000	RECLG 003411 G
INTPRI= 000340 G	L\$DVTY 002164 G	L10023 022302	O\$AU = 000001	RECREG 006312
IRE 003445 G	L\$EF 002052 G	L10024 022322	O\$BGNR= 000001	RECTAP 006346 G
IREC 002211 G	L\$ENVI 002044 G	L10025 022342	O\$BGNS= 000001	RECU 011310 G
IRECM 026225	L\$ETP 002102 G	L10026 022362	O\$DU = 000001	REPEAT= 050224
IREM 026321	L\$EXP1 002046 G	L10027 022402	O\$ERRT= 000000	RERM 004550 G
ISR = 000100 G	L\$EXP4 002064 G	L10030 022422	O\$GNSW= 000001	RETRY = 050222
IXE = 004000 G	L\$EXP5 002066 G	L10031 022442	O\$POIN= 000001	RETRYC 003404 G
I\$AU = 000041	L\$HARD 025266 G	L10032 022500	O\$SETU= 000001	REWRT 013752
I\$AUTO= 000041	L\$HIME 002120 G	L10033 022664	PASCNT 003254 G	RFBC 002654 G
I\$CLN = 000041	L\$HPCP 002016 G	L10034 024124	PATCH 026542 G	RFCEM 004254 G
I\$DU = 000041	L\$HPTP 002022 G	L10035 024270	PATERN 003374 G	RFREC 002754 G
I\$HRD = 000041	L\$HW 002174 G	L10036 024422	PATRO 010114 G	RFUNR 002764 G
I\$INIT= 000041	L\$ICP 002104 G	L10037 025262	PATR1 010152 G	RLEXM 004310 G
I\$MOD = 000041	L\$INIT 017704 G	L10040 025336	PATR2 010172 G	RNF = 125401 G
I\$MSG = 000041	L\$LADP 002026 G	L10041 026542	PATR3 010202 G	RNOPSC= 177740 G
I\$PROT= 000040	L\$LAST 026746 G	L10042 026752	PATR4 010226 G	RNR = 105401 G
I\$PTAB= 000041	L\$LOAD 002100 G	L10044 026756	PATR5 010240 G	RNYM 004504 G
I\$PWR = 000041	L\$LUN 002074 G	MBR = 100012 G	PATR6 010252 G	RPF = 105001 G
I\$RPT = 000041	L\$MREV 002050 G	MEMOM 021126	PATR7 010272 G	RPR = 125001 G
I\$SEG = 000041	L\$NAME 002000 G	MISCFG 003455 G	PATR8 010324 G	RPTCNT 003406 G
I\$SETU= 000041	L\$PRIO 002042 G	MOD.CO= 000400 G	PATBL 010072	RPTFLG 003443 G
I\$SFT = 000041	L\$PROT 017676 G	MOD.C1= 001000 G	PATM 026463	RPT1A 017012
I\$SRV = 000041	L\$PRT 002112 G	MOD.C2= 002000 G	PCMDWD 003352 G	RPT1B 017067
I\$SUB = 000041	L\$REPP 002062 G	MOD.C3= 004000 G	PIRE 002214 G	RPT1C 017140
I\$TST = 000041	L\$REV 002010 G	MOVMSG 011224 G	PNT = 001000 G	RPT1D 017211
JLOC 003372 G	L\$RPT 016150 G	MSGCNT= 000016 G	PRI = 002000 G	RPT1E 017437
JLOOP 003370 G	L\$SOFT 025340 G	MSGPKA 002502 G	PRI00 = 000000 G	RPT1F 017315



RPT1G	017366	TCC5	012552 G	T\$PTAB=	010043	URERM	004572 G	\$F\$L00=	000200
RPT1I	017563	TCC6	012650 G	T\$PTHV=	000001	VFEXC	014504 G	\$F\$NAM=	000160
RPT1J	017467	TCC7	012700 G	T\$PTNU=	000001	VFISU	014750 G	\$F\$NO =	000403
RPT1K	017554	TC2RTN	012062	T\$SAVL=	177777	VFYCNF	003274 G	\$F\$OR =	000320
RRANV	002203 G	TIME1	003364 G	T\$SEGL=	177777	VFYDAT	014402 G	\$F\$RTN=	000300
RRBC	002614 G	TIME2	003366 G	T\$SIZE=	000004	VFYFLG	003442 G	\$F\$SEL=	000140
RRCL =	000020 G	TOERM	004207 G	T\$SUBN=	000000	VFY.C =	000100 G	\$F\$THE	000330
RRREC	002734 G	TOOMM	004460 G	T\$TAGL=	177777	WLKZRO	010206	\$F\$TRU=	000404
RRUNR	002744 G	TRAPD4	003454 G	T\$TAGN=	010045	WRBC	002554 G	\$F\$UNT=	000130
RRVM	026121	TRAP4	021646 G	T\$TEMP=	000000	WRCL =	000020 G	\$F\$WHI=	000120
RTLE	012716 G	TSAM	004436 G	T\$TEST=	000005	WRR =	105005 G	\$F\$YES=	000402
RTLRTN	013042	TSBA =	002452 G	T\$TSTM=	177777	WRREC	002714 G	\$IFLEV=	177777
RWCPK	002330 G	TSC.FC=	177717 G	T\$TSTS=	000001	WRT =	104005 G	\$ISKO =	000001
RWD =	102010 G	TSC.TC=	177761 G	T\$TAU =	010016	WRTY	013044 G	\$ISK1 =	000001
RWERR	003413 G	TSDB	002452 G	T\$TAUT=	010013	WRTYCT	003244 G	\$ISK2 =	000001
RSSAVE	003400 G	TSSR	002462 G	T\$TCLE=	010014	WRTYER	003410 G	\$ISK3 =	000001
SCCNT	003264 G	TSSREG	003402 G	T\$TDAT=	010044	WRTYFG	003407 G	\$ISK4 =	000001
SCERM	004230 G	TSVCT	002472 G	T\$TDU =	010015	WRUNR	002724 G	\$ISK5 =	000001
SCH =	140004 G	TS.A16=	000400 G	T\$THAR=	010040	WSSR	011170 G	\$ISK6 =	000001
SCHBK	002442 G	TS.A17=	001000 G	T\$THW =	010000	WTM =	100011 G	\$LOCTA=	177777
SCHCNT=	000010 G	TS.NBA=	002000 G	T\$THNI=	010012	WTR =	101011 G	\$LSTCN=	177777
SEQEND	003550 G	TS.NXM=	004000 G	T\$THMSG=	010003	WTV =	104105 G	\$LSTIN=	000001
SEICH	006444 G	TS.OFL=	000100 G	T\$THPC =	000001	WTVERM	004164 G	\$LSTST=	177777
SETRW	006470 G	TS.RMR=	010000 G	T\$THPRO=	010011	WTYBRF	013370	\$LSTTA=	000001
SETUP	007452 G	TS.SC =	100000 G	T\$THPTA=	010043	WTYCMD	013364	\$MCALL=	000000
SFF =	105010 G	TS.SPE=	020000 G	T\$THRPT=	010010	WTYWRD	013366	\$NESTL=	177777
SFPTBL	002202 G	TS.SSR=	000200 G	T\$THSOF=	010041	X\$ALWA=	000000	\$NSKO =	000120
SFR =	105410 G	TS.UPE=	040000 G	T\$THSRV=	010007	X\$FALS=	000040	\$NSK1 =	000120
SRF =	104010 G	TS4ADR	025312	T\$THSUB=	010033	X\$OFFS=	000400	\$NSK2 =	000110
SRR =	104410 G	TS4CL	002522 G	T\$THSW =	010001	X\$TRUE=	000020	\$NSK3 =	000110
STAERM	005372 G	TS4INT	002512 G	T\$THTES=	010037	XO.BOT=	000002 G	\$NSK4 =	000110
STAER1	005704	TS4INO	006316 G	T1	022064 G	XO.EOT=	000001 G	\$NSK5 =	000110
STAER2	006062	TS4IN0	006316 G	T1SWB	003447 G	XO.LET=	020000 G	\$NSK6 =	000110
STAER3	006141	TS4IN1	006324 G	T1.1	022074	XO.ONL=	000100 G	\$SAVLE=	177777
STAER4	006177	TS4IN2	006332 G	T1.10	022424	XO.RLL=	010000 G	\$SSKO =	050452
STAER5	006217	TS4IN3	006340 G	T1.11	022444	XO.RLS=	040000 G	\$TAGLE=	177777
STAER6	006217	TS4VCT	025327	T1.12	022530	XO.TMK=	100000 G	\$TAGNU=	050470
STAER7	005776	T\$ARGC=	000003	T1.2	022220	X2.OPM=	100000 G	\$TEMP =	000402
STAF LG	003452 G	T\$CODE=	023004	T1.3	022224	X3.DCK=	000010 G	\$TSKO =	050451
SVCGBL=	000000	T\$ERRN=	000001	T1.4	022244	X3.RNY=	157400 G	\$TSK1 =	050452
SVCINS=	000001	T\$EXCP=	000000	T1.5	022264	ZROPAT	010156	\$TSK2 =	050453
SVC SUB=	000000	T\$FLAG=	000041	T1.6	022304	\$BGNLE=	177777	\$TSK3 =	050454
SVC TAG=	000000	T\$FREE=	026756	T1.7	022324	\$ERFLG=	000400	\$TSK4 =	050467
SVC TST=	000000	T\$GMAN=	000000	T1.8	022344	\$F\$AND=	000310	\$TSK5 =	050466
SWBFLG	003444 G	T\$HILT=	000010	T1.9	022364	\$F\$BAD=	000401	\$TSK6 =	050462
SWB.C =	010000 G	T\$LAST=	000001	T2	022404	\$F\$BLA=	000170	\$TSK7 =	050464
S\$LSYM=	010000	T\$LOLI=	000000	T3	023452 G	\$F\$CAS=	000150	\$ARGC=	000000
TCCRA	011700	T\$LSYM=	010000	T4	024126 G	\$F\$DEC=	000220	\$BYTE=	000403
TCCO	011720 G	T\$LTNO=	000005	T5	024272 G	\$F\$DO =	000340	\$CASE=	000000
TCC1	011736 G	T\$NEST=	177777	TSWROT	025156	\$F\$FAL=	000405	\$DST =	000000
TCC2	011754 G	T\$NSO =	000000	UAM =	000200 G	\$F\$G00=	000400	\$ELOC=	000402
TCC3	012064 G	T\$NS1 =	000005	UNL =	100412 G	\$F\$IF =	000110	\$ERFL=	000000
TCC4	012102 G	T\$NS2 =	000002	UNREC	003414 G	\$F\$INC=	000210	\$FLAG=	000001
		T\$PCNT=	000000						

M14

PARAMETER CODING MACY11 30(1046) 06 APR 84 08:51 PAGE 184  
CZTSHD.P11 06 APR 84 08:49 SYMBOL TABLE

SEQ 0181

\$\$FROM= 000000	\$\$REG = 177777	\$\$RTN2= 000000	\$\$TGS1= 000000	\$\$\$TAG= 050000
\$\$LOC = 025072	\$\$RETU= 000000	\$\$SRC = 000000	\$\$TGS2= 000000	. = 026756
\$\$LOCN 000000	\$\$RTN1= 000000	\$\$TGSV= 000000	\$\$TO = 000000	

. ABS. 026756 000

ERRORS DETECTED: 0

CZTSHD,CZTSHD/SOL/EQ:ONEFILE=SVC.SML,SPMAC.SML,CZTSHD.P11  
RUN-TIME: 132 138 .8 SECONDS  
RUN TIME RATIO: 347/271=1.2  
CORE USED: 31K (62 PAGES)