

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
40
41
42
43
44
45
46
47
48
49

.REM_

IDENTIFICATION

PRODUCT CODE: AC 1822A MC
PRODUCT NAME: CNTSEAO TSV05 DATA RELIABILITY
PRODUCT DATE: 09 APR 84
MAINTAINER: ISS DIAGNOSTIC SERVICES
AUTHOR: DICK GORDON
MODIFIED BY: JAKI BERG

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 ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

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
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107

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-RECOVERY

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

2.0 OPERATING INSTRUCTIONS

2.1 HARDWARE PARAMETERS

2.2 SOFTWARE PARAMETERS

2.2.1 TS05 COMMAND LIST

2.2.2 DATA PATTERNS

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

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
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164

2.4.1 SYSTEM CONFIGURATION
2.4.2 TEST EXECUTION TIMES

3.0 ERROR INFORMATION

3.1 ERROR REPORTING

- 3.1.1 ERROR #1 - COMMAND PACKET ADDRESS IS NOT ON A MODULO 4 BOUNDRY
- 3.1.2 ERROR #2 - TS05 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 - ICC0, UNDEFINED SPECIAL CONDITION
 - 3.1.5.2 ERROR #6 - ICC1, ATTENTION CONDITION
 - 3.1.5.3 ERROR #7 - ICC2, TAPE STATUS ALERT
 - 3.1.5.4 ERROR #8 - ICC3, FUNCTION REJECT
 - 3.1.5.5 ERROR #9 - ICC4, RECOVERABLE ERROR
 - 3.1.5.6 ERROR #10 - ICC5, RECOVERABLE ERROR
 - 3.1.5.7 ERROR #11 - ICC6, UNRECOVERABLE ERROR
 - 3.1.5.8 ERROR #12 - ICC?, 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 - RANDOM/OPERATOR SELECTED COMMAND SEQUENCE

6.0 DEVICE INFORMATION

- 6.1 GENERAL
- 6.2 Q-BUS INTERFACE SPECIFICATIONS
- 6.3 BIT DEFINITIONS FOR TSV05/TS05 REGISTERS
 - 6.3.1 TSV05/TS05 REGISTER SUMMARY
 - 6.3.2 TSV05 STATUS REGISTER (TSSR)
 - 6.3.2.1 TSV05 EXTENDED DATA BUFFER REGISTER (TSDBX)
 - 6.3.3 EXTENDED STATUS REGISTER 0 (XSTAT0)
 - 6.3.4 EXTENDED STATUS REGISTER 1 (XSTAT1)
 - 6.3.5 EXTENDED STATUS REGISTER 2 (XSTAT2)

856

8

NO PATTERN GENERATION.

858
859
860
861
862
863
864
865
866
867
868
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

2.3 EXAMPLES OF SOFTWARE DIALOGUE

CHANGE HW (L) ?
#UNITS (D) ?
TSDB ADDRESS (O) 176000 ?
VECTOR (O) 224 ?
SELECT DRIVE 0-1 (O) ?

IN ADDITION, ON A START, RESTART OR CONTINUE THE SUPERVISOR REQUESTS CHANGES TO THE SOFTWARE OPERATING PARAMETERS, AS FOLLOWS:

CHANGE SW (L) ?

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 ? <CR>
PRINT RECOVERABLE ERRORS (L) N ? Y<CR>
HALT AFTER EACH CMD (L) N ? <CR>
INHIBIT RECOVERY (L) N ? <CR>
BAD TAPE SPOT DETECTION (L) Y ? <CR>
DISABLE INTERRUPTS (L) N ? <CR>
INHIBIT RFC ERROR REPORT (L) N ? <CR>
CHANGE CMD SEQUENCE (L) N ? <CR>
DEFAULT SWITCH SETTINGS (L) Y ? <CR>

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>

230
231
232
233
234
235
236
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
280
281
282
283
284
285
286

IS ALLOCATED BY THE SUPERVISOR ON REQUEST OR CHOSEN BY PROGRAMMER TO RESIDE BETWEEN THE DIAG AND THE SUPERVISOR.

1.1.4 DIAGNOSTIC INFORMATION

1.1.4.1 SCOPE

THIS DIAGNOSTIC CAN TEST ONE CONTROLLER AND UP TO 2 DRIVES. THE 2 DRIVES ARE ASSIGNED LOGICAL DRIVE NUMBERS 0 - 1 BY THE DIAGNOSTIC.

THERE ARE 5 TESTS IN THIS PROGRAM:

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

1.1.4.2 ERROR RECOVERY

ERROR RECOVERY IS PERFORMED ON READ, WRITE AND WRITE TAPE MARK FUNCTIONS UNLESS ERROR RECOVERY IS INHIBITED BY THE OPERATOR AT START UP TIME. THE READ FORWARD/READ REVERSE RETRY LIMIT IS 16 (8 IN THE SAME DIRECTION AND 8 IN THE OPPOSITE DIRECTION). FOR MORE DETAILED INFORMATION ON ERROR RECOVERY PROCEDURES, REFER TO SECTION 3.0 (ERROR REPORTING) OF THIS LISTING.

1.1.4.3 WRITE ERROR RECOVERY

THERE ARE 2 SELECTABLE WRITE-ERROR RECOVERY ALGORITHMS USED BY THIS DIAGNOSTIC:

1. MEDIA/OPERATIONAL SELECTIVE ALGORITHM
2. OPERATIONAL ALGORITHM

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

TO SELECT THE SECOND ALGORITHM:
ANSWER 'Y' TO CHANCE SW (L) ?
ANSWER 'N' TO BAD TAPE SPOT DETECTION (L) Y ?

IF 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

THIS ALGORITHM IDENTIFIES MEDIA RELATED WRITE ERRORS FROM OPERATIONAL ONES.

ALGORITHM

A WRITE RETRY SUBROUTINE IS CALLED BY THE RECOVERABLE ERROR SUBROUTINE WHICH IS ENTERED UPON DETECTION OF A WRITE RECOVERABLE ERROR. THE WRITE RETRY SUBROUTINE ATTEMPTS TO REWRITE THE RECORD IN SAME SPOT ON TAPE 4 TIMES.

972
973
974
975
976
977
978
979
980
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

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:

```
CNTSE 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 XST4
XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX
```

* CAUTION *

INTERPRET THAT "RECORD READ" COUNT WITH CAUTION. IF VERY DIFFERENT FROM RECORD COUNT TRACKED BY THE DIAGNOSTIC, TAPE 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 COUNT IS RESET TO ZERO ALTHOUGH THE TAPE IS NOT REWOUND. THIS IS NECESSARY BECAUSE THERE IS NO ACCURATE WAY TO DETERMINE ON WHAT RECORD COUNT OF WHICH UNIT THE DIAGNOSTIC WAS HALTED BEFORE RESTARTING OR CONTINUING. IT IS SUGGESTED THAT A "PRINT" BE REQUESTED WHEN HALTING DIAG TO GET A PRINT OF THE RECORD COUNT WHEN HALTED.

EXAMPLE OF AN ERROR REPORT:

```
CNTSE SFT ERR 00009 TST 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 XST4
000350 000002 100004 000000 040055
```

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
392
393
394
395
396
397
398
399
400

AND REFLECTS THE SPECIFICATIONS OF THE HARDWARE UNDER TEST.
TO CLEAR CUMULATIVE COUNTS, ANSWER 'Y' TO: CLEAR COUNTERS (L) Y ?.
THE BAD TAPE SPOTS COUNT IS THEN CLEARED WHEN WRITING THE TAPE FROM BOT.
IF TEST 2 IS HALTED, THEN RESTARTED OR CONTINUED, THE RECORD COUNT
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 PRINT OUT FOR A BAD SPOT ON TAPE:

```
CNTSE 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 XST4
000350 000002 100400 000000 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 SPOY AFTER 2 RETRY, 1 REPEAT
```

```
CNTSE 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 XST4
000350 000002 100010 000000 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
WRT RCR RDF
```

1086
1087
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

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 IMK, 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.

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

1143
1144
1145
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
1192
1193
1194
1195
1196
1197
1198
1199

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:HOE. THERE ARE NO OTHER HALTS.

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			
	WRT	RDR	RDF	
RECOVERABLE ERRORS	XXXXX	XXXXX	XXXXX	
UNRECOVERABLE ERRORS	XXXXX	XXXXX	XXXXX	

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
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570

2.0 OPERATING INSTRUCTIONS

THIS SECTION CONTAINS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES.
FOR DETAILED INFORMATION, REFER TO THE XXDP+ USER'S MANUAL.

COMMANDS

THERE ARE ELEVEN LEGAL COMMANDS FOR THE DIAGNOSTIC RUNTIME SERVICES
(SUPERVISOR). THIS SECTION LISTS THE COMMANDS AND GIVES A VERY
BRIEF DESCRIPTION OF THEM. THE XXDP+ USER'S MANUAL HAS MORE DETAILS.

COMMAND	EFFECT
START	START THE DIAGNOSTIC FROM AN INITIAL STATE
RESTART	START THE DIAGNOSTIC WITHOUT INITIALIZING
CONTINUE	CONTINUE AT TEST THAT WAS INTERRUPTED (AFTER :C)
PROCEED	CONTINUE FROM AN ERROR HALT
EXIT	RETURN TO XXDP+ MONITOR (XXDP+ OPERATION ONLY!)
ADD	ACTIVATE A UNIT FOR TESTING (ALL UNITS ARE CONSIDERED TO BE ACTIVE AT START TIME)
DROP	DEACTIVATE A UNIT
PRINT	PRINT STATISTICAL INFORMATION (IF IMPLEMENTED BY THE DIAGNOSTIC)
DISPLAY	TYPE A LIST OF ALL DEVICE INFORMATION
FLAGS	TYPE THE STATE OF ALL FLAGS
ZFLAGS	CLEAR ALL FLAGS

A COMMAND CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. SO
YOU MAY, FOR EXAMPLE, TYPE "STA" INSTEAD OF "START".

OPERATOR COMMANDS

THE TSV05 DIAGNOSTIC IS A SBC-11/21+ DIAGNOSTIC SUPERVISOR COMPATIBLE
PROGRAM. ALL LOADING AND RUNTIME INSTRUCTIONS CAN BE REFERENCED IN THE
PDP-11 XXDP+ PROGRAMMERS MANUAL. THE USER ENTRY IS IN QUOTES.

BOOT THE DIAGNOSTIC XXDP MEDIA

```
CHMDLBO XXDP+ DL MONITOR 28K
BOOTED VIA UNIT 0
ENTER DATE (DD-MMM-YR): " ENTER DATE OR JUST <CR> "
RESTART ADDRESS: 153726
50 HZ? N " <CR> "
LSI? N " Y<CR> "
THIS IS XXDP+. TYPE "H" OR "H/L" FOR DETAILS
R: NTSEAO
NTSEA0BINDRS LOADED
DIAG. RUN-TIME SERVICES REV D. APR 79
CNTSE-A-0
```

1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292

- * 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.

1294
1295
1296
1297
1298
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

- 5.2 TEST 2 - DATA RELIABILITY.
1. THE TAPE IS INITIATED WITH THE FOLLOWING COMMANDS:
SET CHARACTERISTIC 40
REWIND
WRITE 64 RECORDS OF RANDOM LENGTH AND DATA
 2. WRITE AND READ COMMANDS ARE SELECTED AT RANDOM AND ARE EXECUTED A RANDOM NUMBER OF TIMES WITH RANDOM LENGTHS AND RANDOM PATTERN UNTIL END OF TAPE IS REACHED.
 3. AT THE END OF EACH PASS, A REWIND COMMAND IS ISSUED AND A PERFORMANCE REPORT IS PRINTED.
- NOTE: IF A RESTART COMMAND IS USED TO INITIATE TEST 1, THE INITIAL REWIND COMMAND IS NOT ISSUED.
- 5.3 TEST 3 - WRITE COMPATABILITY/WRITE UTILITY.
REWINDS AND WRITES RECORDS OF RANDOM LENGTHS AND RANDOM DATA FROM BOT TO EOT.
- 5.4 TEST 4 - READ COMPATABILITY/READ UTILITY.
REWINDS AND READS ENTIRE TAPE, FORWARD AND REVERSE.
- 5.5 TEST 5 - RANDOM/OPERATOR SELECTED COMMAND SEQUENCE.
A DEFAULT SEQUENCE OF REWIND/WRITE/READ REV/READ AND/REWIND OF ENTIRE TAPE IS EXECUTED WITH RANDOM PATTERN AND RECORD LENGTH OF 2048 BYTES. OPERATOR CAN ENTER SEQUENCE OF COMMANDS UP TO SEVEN IF THEY DON'T WANT DEFAULT SEQUENCE.
- 6.0 DEVICE INFORMATION TABLES

- 6.1 GENERAL

- THE TS05 TAPE SUBSYSTEM CONSISTS OF A TSV05 Q-BUS CONTROLLER CONNECTED TO A TS05 DRIVE. FROM A SOFTWARE VIEWPOINT THIS CONFIGURATION IS UNIQUE (FOR A Q-BUS 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 TS05 SUBSYSTEM TO FETCH THE WORD(S) WITHIN THE COMMAND PACKET. THE WORDS WITHIN THE COMMAND PACKET ARE:

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
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741

TSDB ADDRESS (0) 176000 ?

VECTOR (0) 224 ?

SELECT DRIVE 0-1 (0) ?

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 TSDB(S). IF NO RESPONSE, IT DROPS THE UNIT(S) IMMEDIATELY WITH THE FOLLOWING MESSAGE:

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

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 DRIVE IS CHECKED. A MESSAGE SHALL BE PRINTED EVERY 50 SECONDS TO WARN THE OPERATOR OF DRIVES BEING NOT READY OR OFF-LINE. THESE DRIVES SHALL BE DROPPED AFTER A REASONABLE AMOUNT OF TIME.

2.2 SOFTWARE PARAMETERS

THE FOLLOWING QUESTIONS ARE ASKED WHEN ONE ANSWERS YES TO THE CHANGE SOFTWARE QUESTION ON A START, RESTART, OR CONTINUE. THEY ALLOW FLEXIBILITY 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 ? (SEE NOTE1:)

DEFAULT SWITCH SETTINGS (L) Y ?

100IPS (L) N ?

WRITE BUFFERING (L) N ?

READ BUFFERING (L) N ?

742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798

ANSWERING NO TO THE DEFAULT SWITCH QUESTION WILL CAUSE THE 100 IPS QUESTION TO BE ASKED.

ANSWERING YES TO THE 100 IPS QUESTION WILL INHIBIT THE LAST TWO QUESTIONS.

ANSWERING NO TO THE 100 IPS QUESTION WILL CAUSE THE WRITE BUFFERING QUESTION TO BE ASKED.

ANSWERING YES TO THE WRITE BUFFERING QUESTION WILL INHIBIT THE LAST QUESTION.

ANSWERING NO TO THE WRITE BUFFERING QUESTION WILL CAUSE THE READ BUFFERING QUESTION TO BE ASKED.

NOTE1: THIS QUESTION SHOULD BE ANSWERED (N) UNLESS AN OPERATOR SELECTED SEQUENCE IS TO BE EXECUTED. IF THIS QUESTION WAS ANSWERED Y, THE FOLLOWING QUESTIONS MUST BE ANSWERED OR DEFAULTED WITH A <CR> ONLY:

CHARACTERISTICS CODE (0) 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)
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)

NOTE2: THE PROGRAM AUTOMATICALLY INSERTS A CHARACTERISTIC CODE OF 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

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
844
845
846
847
848
849
850
851
852
853
854
855

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 DIALOGUE.

2.2.1 COMMAND LIST FOR USE IN SOFTWARE DIALOGUE.

CODE	COMMAND	DESCRIPTION
1	DRI	DRIVE INITIATE.
2	RDF	READ FORWARD.
3	RDR	READ REVERSE.
4	WRT	WRITE.
5	WTV	WRITE/VERIFY. IE. WRITE N RECORDS; READ REVERSE AND CHECK N RECORDS OF DATA; READ FORWARD AND CHECK N RECORDS.
6	SRF	SPACE RECORDS FORWARD.
7	SRR	SPACE RECORDS REVERSE.
8	RNR	READ NEXT REVERSE, IE. SPACE FWD, READ REV.
9	RNF	READ NEXT FORWARD, IE. READ FWD, SPACE REV.
10	RPF	READ PREVIOUS FWD, IE. SPACE REV, READ FWD.
11	RPR	READ PREVIOUS REV, IE. READ REV, SPACE FWD.
12	WRR	WRITE RETRY.
13	RWD	REWIND.
14	MBR	MESSAGE BUFFER RELEASE.
15	WTM	WRITE TAPE MARK.
16	WTR	WRITE TAPE MARK RETRY.
17	SFF	SPACE FILES FORWARD.
18	SFR	SPACE FILES REVERSE.
19	GES	GET EXTENDED STATUS.
20	ERS	ERASE 3 INCHES OF TAPE.
21	UNL	UNLOAD.
22	CLN	CLEAN TAPE
23	SCH	SET DEVICE CHARACTERISTIC. WHERE BRF=200, 40, 20, 0. 200 = ENABLE SKIP TAPE MARKS STOP (STOP AT LOGICAL EOT) 40 = ENABLE ATTENTION INTERRUPTS. 20 = ENABLE MESSAGE BUFFER RELEASE INTERRUPTS. SEE TSV05/TS05 PROGRAMMING SPECIFICATION FOR DESCRIPTION.
25	JMP	JUMP TO THE NTH COMMAND IN THE COMMAND SEQUENCE TABLE, WHERE N IS DEFINED IN THE BRF FIELD. THE NUMBER OF JUMPS IS ENTERED IN THE # OF OPERATIONS FIELD
26	DLY	DELAY "N" MILLISECONDS WHERE N IS DEFINED IN THE # OF OPERATIONS.
27	END	END OF COMMAND SEQUENCE.

2.2.2 DATA PATTERN LIST FOR USE IN SOFTWARE DIALOGUE.

PATTERN #	DESCRIPTION.
0	INCREMENTING PATTERN. 0 - 377.
1	ALL "1"'S PATTERN.
2	ALL "0"'S PATTERN.
3	"1" BIT WALKING FROM R TO L IN A FIELD OF "0"'S.
4	"0" BIT WALKING FROM R TO L IF A FIELD OF "1"'S.
5	ALTERNATING "1" AND "0" BITS WITH ALTERNATE BYTES COMPLIMENTED.
6	ALTERNATING BYTES OF 000 AND 377.
7	RANDOM DATA PATTERN.

856

8 NO PATTERN GENERATION.

858
859
860
861
862
863
864
865
866
867
868
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

2.3 EXAMPLES OF SOFTWARE DIALOGUE

```

CHANGE HW (L) ?
#UNITS (D) ?
TSDB ADDRESS (O) 176000 ?
VECTOR (O) 224 ?
SELECT DRIVE 0-1 (O) ?
    
```

IN ADDITION, ON A START, RESTART OR CONTINUE THE SUPERVISOR REQUESTS CHANGES TO THE SOFTWARE OPERATING PARAMETERS, AS FOLLOWS:

```
CHANGE SW (L) ?
```

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 ?                   <CR>
PRINT RECOVERABLE ERRORS (L) N ?                 Y<CR>
HALT AFTER EACH CMD (L) N ?                       <CR>
INHIBIT RECOVERY (L) N ?                          <CR>
BAD TAPE SPOT DETECTION (L) Y ?                  <CR>
DISABLE INTERRUPTS (L) N ?                        <CR>
INHIBIT RFC ERROR REPORT (L) N ?                 <CR>
CHANGE CMD SEQUENCE (L) N ?                      <CR>
DEFAULT SWITCH SETTINGS (L) Y ?                  <CR>
    
```

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>
    
```

915 INHIBIT RFC ERROR REPORT (L) N ? Y<CR>
 916 CHANGE CMD SEQUENCE (L) N ? N<CR>
 917 DEFAULT SWITCH SETTINGS (L) Y ? <CR>
 918
 919

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

920
 921
 922 A) RECEIVE PROMPT (DR>)
 923 B) ENTER STA/TS:5/FLA:IER:ISR:IDU/FOP:1000<CR>
 924 C) ANSWER HARDWARE QUESTIONS.
 925 D) PROCEED WITH THE FOLLOWING DIALOGUE:
 926
 927 CHANGE SW (L) ? Y<CR>
 928 CLEAR COUNTERS (L) N ? Y<CR>
 929 RESET RANDOM VARIABLES (L) N ? N<CR>
 930 PRINT RECOVERABLE ERRORS (L) N ? N<CR>
 931 HALT AFTER EACH CMD (L) N ? N<CR>
 932 INHIBIT RECOVERY (L) N ? N<CR>
 933 BAD TAPE SPOT DETECTION (L) Y ? N<CR>
 934 DISABLE INTERRUPTS (L) N ? Y<CR>
 935 INHIBIT RFC ERROR REPORT (L) N ? Y<CR>
 936 CHANGE CMD SEQUENCE (L) N ? Y<CR>
 937 CHARACTERISTICS CODE (D) 40 ? 40<CR>
 938 CMD/2 (D) 5 ? 13<CR> (REWIND)
 939 BRP COUNT (D) 2048 ? 1<CR>
 940 # OF OPERATIONS (D) 10 ? 1<CR>
 941 PATTERN (D) 7 ? 1<CR>
 942 CMD/3 (D) 5 ? 4<CR> (WRITE)
 943 BRP (D) 2048 ? 1000<CR>
 944 # OF OPERATIONS (D) 10 ? 10000<CR>
 945 PATTERN (D) 7 ? 1<CR>
 946 CMD/4 (D) 5 ? 27<CR> (END)
 947 BRP (D) 2048 ? <+Z>
 948

2.4 EXECUTION TIMES

2.4.1 SYSTEM CONFIGURATION

PDP11/21+
 MEMORY
 CONSOLE
 TSV05/TS05

2.4.2 TEST EXECUTION TIME (2400 FT. TAPE)

THE TIME TO EXECUTE THE FIVE TESTS IS APPROXIAMATELY 4 HOURS.

NOTE: THE EXECUTION TIMES GIVEN IS FOR ONE DRIVE OPERATION.

3.0 ERROR INFORMATION

971

972
973
974
975
976
977
978
979
980
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

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:

```
CNTSE 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
XXXXXXXX XXXXXX XXXXXX XXXXXX X
XXXXXXXX
XXXXXXXX
XXXXXXXX
XST0 XST1 XST2 XST3 XST4
XXXXXXXX XXXXXX XXXXXX XXXXXX XXXXXX
```

* CAUTION *

INTERPRET THAT "RECORD READ" COUNT WITH CAUTION. IF VERY DIFFERENT FROM RECORD COUNT TRACKED BY THE DIAGNOSTIC, TAPE 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 COUNT IS RESET TO ZERO ALTHOUGH THE TAPE IS NOT REWOUND. THIS IS NECESSARY BECAUSE THERE IS NO ACCURATE WAY TO DETERMINE ON WHAT RECORD COUNT OF WHICH UNIT THE DIAGNOSTIC WAS HALTED BEFORE RESTARTING OR CONTINUING. IT IS SUGGESTED THAT A "PRINT" BE REQUESTED WHEN HALTING DIAG TO GET A PRINT OF THE RECORD COUNT WHEN HALTED.

EXAMPLE OF AN ERROR REPORT:

```
CNTSE SFT ERR 00009 TST 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 XST4
000350 000002 100004 000000 040055
```

1029
1030
1031
1032
1033
1034
1035
1036
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

- 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 - TS05 NOT READY:
BEFORE ANY COMMAND IS ISSUED TO THE TS05, THE SUBSYSTEM READY BIT IN THE TSSR 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 TS05 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
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.

1086
1087
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

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.

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

1143
1144
1145
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
1192
1193
1194
1195
1196
1197
1198
1199

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:HOE. THERE ARE NO OTHER HALTS.

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

1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
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

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 TS05 FUNCTIONS.
SUBTEST 1 - SET CHAR, DRIVE INIT, GET STATUS.
 + SET CHARACTERISTIC 200.
 + DRIVE INITIATE.
 + SET CHARACTERISTIC 20.
 + GET STATUS
 + SET CHARACTERISTIC 40.
 + PRINT TS05 MICROCODE LEVEL (PASS 1 ONLY)
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.
 + SPACE 2 FILES REVERSE.
 + SPACE 2 FILES FORWARD.
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.

1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292

- + 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.

1294
1295
1296
1297
1298
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

5.2 TEST 2 - DATA RELIABILITY.

1. THE TAPE IS INITIATED WITH THE FOLLOWING COMMANDS:
SET CHARACTERISTIC 40
REWIND
WRITE 64 RECORDS OF RANDOM LENGTH AND DATA
2. WRITE AND READ COMMANDS ARE SELECTED AT RANDOM AND ARE EXECUTED A RANDOM NUMBER OF TIMES WITH RANDOM LENGTHS AND RANDOM PATTERN UNTIL END OF TAPE IS REACHED.
3. AT THE END OF EACH PASS, A REWIND COMMAND IS ISSUED AND A PERFORMANCE REPORT IS PRINTED.

NOTE: IF A RESTART COMMAND IS USED TO INITIATE TEST 1, THE INITIAL REWIND COMMAND IS NOT ISSUED.

5.3 TEST 3 - WRITE COMPATABILITY/WRITE UTILITY.

REWINDS AND WRITES RECORDS OF RANDOM LENGTHS AND RANDOM DATA FROM BOT TO EOT.

5.4 TEST 4 - READ COMPATABILITY/READ UTILITY.

REWINDS AND READS ENTIRE TAPE, FORWARD AND REVERSE.

5.5 TEST 5 - RANDOM/OPERATOR SELECTED COMMAND SEQUENCE.

A DEFAULT SEQUENCE OF REWIND/WRITE/READ REV/READ END/REWIND OF ENTIRE TAPE IS EXECUTED WITH RANDOM PATTERN AND RECORD LENGTH OF 2048 BYTES. OPERATOR CAN ENTER SEQUENCE OF COMMANDS UP TO SEVEN IF THEY DON'T WANT DEFAULT SEQUENCE.

6.0 DEVICE INFORMATION TABLES

6.1 GENERAL

THE TS05 TAPE SUBSYSTEM CONSISTS OF A TSV05 Q-BUS CONTROLLER CONNECTED TO A TS05 DRIVE. FROM A SOFTWARE VIEWPOINT THIS CONFIGURATION IS UNIQUE (FOR A Q-BUS 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 TS05 SUBSYSTEM TO FETCH THE WORD(S) WITHIN THE COMMAND PACKET. THE WORDS WITHIN THE COMMAND PACKET ARE:

1351
1352
1353
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

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 TSOS TO TRANSFER EXTENDED STATUS INFORMATION TO THE MEMORY AREA POINTED TO BY THE BUFFER ADDRESS OF THE "GET STATUS" COMMAND. THERE ARE FIVE EXTENDED STATUS REGISTERS. SEE .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) .

1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394

6.2 Q BUS INTERFACE SPECIFICATIONS

TSV05/ TS05	INT. VECTOR	QNBUS ADDRESS	REGISTER
FIRST	224	772520 772522	TSBA/TSQB TSSR

1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452

6.3 BIT DEFINITIONS FOR TSV05/TS05 REGISTERS

6.3.1 TSV05/TS05 REGISTER SUMMARY

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
(R/O) TSBA	A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
(W/O) TSDB	P15	P14	P13	P12	P11	P10	P09	P08	P07	P06	P05	P04	P03	P02	P17	P16
(R/O) TSSR	SC	0	SCE	RMR	NXM	NBA	A17	A16	SSR	OFL	FC1	FC0	TC2	TC1	TC0	0
(W/O) TSDBX	BT	0	0	0	P21	P20	P19	P18	(TSDBX EXISTS ONLY WHEN ENABLED BY THE EXTENDED FEATURES SWITCH ON THE M7196)							
XST0	TMK	RLS	LET	RLI	WLE	NEF	ILC	ILA	MOT	ONL	IE	VCK	PED	WLK	BOT	EOT
XST1	DLT	0	COR	0	0	0	0	RBP	0	0	0	0	0	0	UNC	0
XST2	OPM	RCE	0	0	0	WCF	0	0	RL7	RL6	RL5	RL4	RL3	RL2	RL1	RL0
XST3	MICRO DIAGNOSTIC ERROR CODE								0	OPI	REV	TRF	DCK	0	0	RIB
XST4	HSP	RCE	0	0	0	0	0	0	WRITE RETRY COUNT							

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

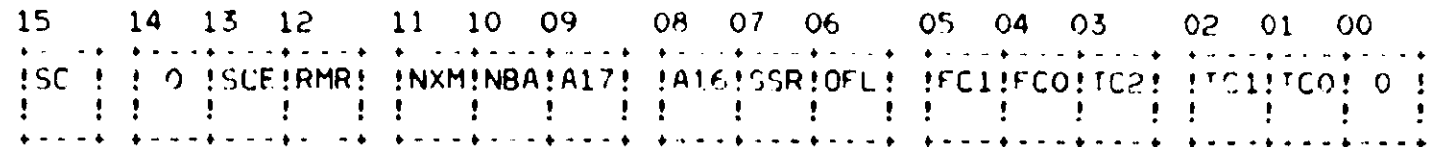
FATAL CLASS CODES (TSSR FC0-FC1):

- 0 - MICRO DIAGNOSTIC FAILURE. SEE ERROR CODE BYTE (XST3) FOR FAILED FUNCTION.
- 1 - RESERVED
- 2 - NOT USED
- 3 - RESERVED FOR FUTURE USE ALWAYS READ AS A 0

1454
1455
1456
1457
1458
1459
1460
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

6.3.2 TSV05 STATUS REGISTER (TSSR)
.....

Q-BUS ADDRESS + 2 READ ONLY



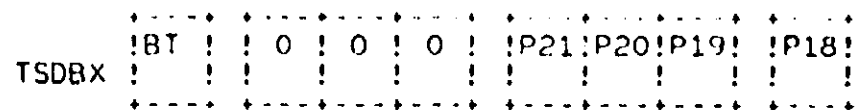
BIT	NAME	TCC	DEFINITION
15	SC	S	SPECIAL CONDITION. WHEN SET, INDICATES THAT THE LAST COMMAND DID NOT COMPLETE WITHOUT INCIDENT. SPECIFICALLY, EITHER AN ERROR WAS DETECTED OR AN EXCEPTION CONDITION OCCURRED. EXCEPTION CONDITIONS CAN BE TAPE MARKS ON READ COMMANDS, REVERSE MOTION AND AT BOT, EOT WHILE WRITING, ETC. MAY ALSO BE SET BY THE ERROR BITS CONTAINED IN THE TSSR REGISTER: SCE, RMR, AND NXM. THE TERMINATION CLASS BITS ARE SOMETHING OTHER THAN 0 (UNLESS RMR IS THE ONLY ERROR - SEE RMR).
14	-	-	RESERVED (ALWAYS A 0)
13	SCE	FC0	SANITY CHECK ERROR-SETS WHEN THE CONTROLLER DETECTS AN ABNORMAL CONDITION WITHIN ITSELF DURING EXECUTION OF IT'S FUNCTIONS AND THE PROBLEM IS SERIOUS ENOUGH THAT A MESSAGE PACKET IS NOT STORED.
12	RMR	S	REGISTER MODIFICATION REFUSED. SET BY THE TSV05 WHEN A COMMAND POINTER IS LOADED INTO TSD8 AND SUB-SYSTEM READY (SSR) IS NOT SET. NOTE THAT THIS BIT CAUSES SPECIAL CONDITION BUT NO TERMINATION CLASS (IN FACT, THE TSV05 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 TSV05 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 TSV05 NEEDS A MESSAGE BUFFER ADDRESS. THIS BIT IS CLEARED DURING THE SET CHARACTERISTICS

1511				COMMAND (IF A GOOD ADDRESS WAS GIVEN).
1512				
1513	09	A17	S	BUS ADDRESS BIT 17. A17 AND A16 (BIT 08) TRACK
1514				THE VALUES OF BITS 17 AND 16 OF THE TSBA
1515				REGISTER. LOADED FROM TSDB BITS 01-00 WHEN TSDB
1516				IS WRITTEN.
1517				
1518	08	A16	S	BUS ADDRESS BIT 16. SEE A17 (BIT 09).
1519				
1520	07	SSR	S	SUB SYSTEM READY. WHEN SET, INDICATES THAT THE
1521				TSV05/TS05 SUBSYSTEM IS NOT BUSY AND IS READY TO
1522				ACCEPT A NEW COMMAND POINTER.
1523				
1524	06	OFL	S,1,3	OFF-LINE. WHEN SET, INDICATES THAT THE TS05 IS
1525				OFF-LINE AND UNAVAILABLE FOR ANY TAPE MOTION
1526				COMMANDS. THIS BIT CAN CAUSE A TERMINATION CLASS
1527				OF 1 (ON ATTN INTERRUPT) OR 3 (RESULTS IN NEF).
1528				
1529	05	FC1	7	FATAL TERMINATION CLASS 01. FC1 AND FC0 (BIT
1530				04) ARE USED TO INDICATE THE TYPE OF FATAL
1531				ERROR WHICH HAS OCCURRED ON THE TS05. THESE
1532				BITS ARE VALID ONLY WHEN SC IS SET AND THE
1533				TERMINATION CLASS CODE BITS ARE ALL SET (111).
1534				
1535	04	FC0	7	FATAL TERMINATION CLASS 00. SEE FC1 (BIT 05).
1536				
1537	03	TC2	S	TERMINATION CLASS BIT 02. THIS BIT, ALONG WITH
1538				THE TC1 AND TC0 BITS, ACT AS AN OFFSET VALUE
1539				WHENEVER AN ERROR OR EXCEPTION CONDITION OCCURS
1540				ON A COMMAND. EACH OF THE EIGHT POSSIBLE
1541				VALUES OF THIS FIELD REPRESENT A PARTICULAR
1542				CLASS OF ERRORS OR EXCEPTIONS. THE CONDITIONS
1543				IN EACH CLASS HAVE SIMILAR SIGNIFICANCE AND, AS
1544				APPLICABLE, RECOVERY PROCEDURES. THE CODE
1545				PROVIDED IN THIS FIELD IS EXPECTED TO BE
1546				UTILIZED AS AN OFFSET INTO A DISPATCH TABLE FOR
1547				HANDLING OF THE CONDITION.
1548				
1549	02	TC1	S	TERMINATION CLASS BIT 01. SEE TC2 (BIT 03).
1550				
1551	01	TC0	S	TERMINATION CLASS BIT 00. SEE TC2 (BIT 03).
1552				
1553	00	-	-	NOT USED. (ALWAYS A 0)
1554				
1555				
1556				
1557				
1558				
1559				
1560				

Q-BUS ADDRESS + 2 - WRITE ONLY
SUBSYSTEM INITIALIZE

1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
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

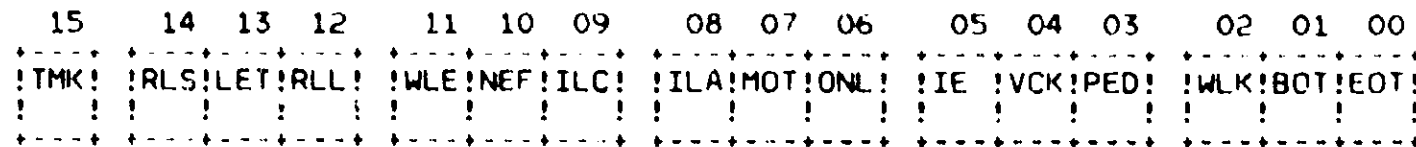
6.3.2.1 TSV05 EXTENDED DATA BUFFER REGISTER (TSDBX)



(TSDBX EXISTS ONLY WHEN
ENABLED BY THE EXTENDED
FEATURES SWITCH ON THE M7196)

BIT	NAME	TCC	DEFINITION
15	BT	-	BOOT COMMAND BIT. WHEN WRITTEN TO A 1, WITH SSR=1, CAUSES THE TAPE TO BE REWOUND TO BOT, THE FIRST TAPE RECORD TO BE SKIPPED, AND THE SECOND RECORD TO BE LOADED INTO CPU MEMORY SPACE STARTING AT LOCATION 0.
14-12			RESERVED (ALWAYS A 0)
11-08	P<21:18>		COMMAND POINTER BITS 21-18. WHEN THE TSDBX IS WRITTEN AND SSR=1, THE DATA IS LOADED INTO BITS 21-18 OF THE INTERNAL TSBA REGISTER.
07-00			RESERVED (ALWAYS A 0)

6.3.3 EXTENDED STATUS REGISTER 0 (XSTAT0)



BIT	NAME	TCC	DEFINITION
15	TMK	S,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.

1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644

13 LET 2

12 RLL 2

11 WLE 3,6

10 NEF 3

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.

RECORD LENGTH LONG. WHEN SET, THIS BIT INDICATES THAT THE RECORD READ WAS LONGER THAN THE BYTE COUNT SPECIFIED.

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.

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 MOTION COMMAND EXCEPT

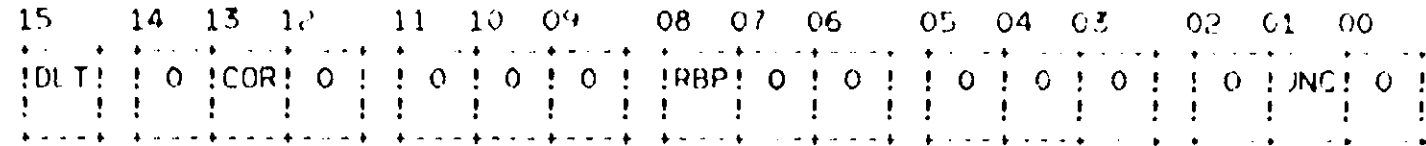
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694

WHEN THE VOLUME CHECK BIT IS SET,
- ANY COMMAND, EXCEPT GET STATUS OR DRIVE INITIALIZE, WHEN THE TS05 IS OFF-LINE.
- ANY WRITE COMMAND WHEN THE TAPE DOES NOT CONTAIN A WRITE ENABLE RING (WRITE LOCK STATUS - WLS).

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 TS05.
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 TS05 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. ALWAYS SET, INDICATES THAT THE TS05 IS CAPABLE OF READING AND WRITING ONLY 1600 BPI PHASE ENCODED DATA.
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.

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

6.3.4 EXTENDED STATUS REGISTER 1 (XSTAT1)

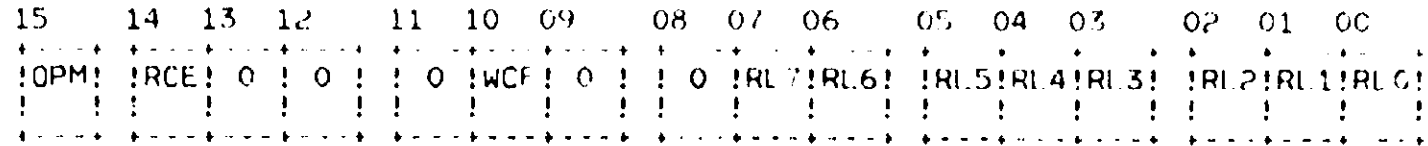


BIT	NAME	TCC	DEFINITION
15	DLT	4	DATA LATE. SET WHEN THE FIFO IS FULL ON A READ OR EMPTY ON A WRITE. THESE CONDITIONS OCCUR WHENEVER THE Q BUS LATENCY EXCEEDS THE DATA TRANSFER RATE OF THE TS05.
14	-	-	NOT USED. (ALWAYS A 0)
13	COR	5	CORRECTABLE DATA. CORRECTABLE DATA ERROR HAS BEEN ENCOUNTERED.
12-09			RESERVED (ALWAYS A 0)
08	RBP	4	READ BUS PARITY ERROR. SET WHEN CONTROLLER DETECTS A PARITY ERROR ON THE READ DATA LINES OF THE TRANSPORT BUS.
07-02 & 00			RESERVED (ALWAYS A 0)
01	UNC	4	UNCORRECTABLE DATA ERROR.

14

1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763

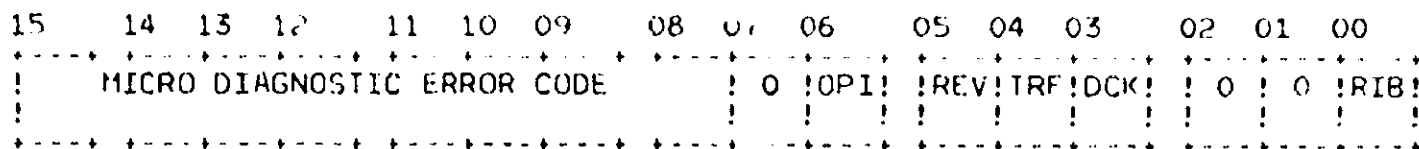
6.3.5 EXTENDED STATUS REGISTER 2 (XSTAT2)



BIT	NAME	TCC	DEFINITION
15	OPM	S	OPERATION IN PROGRESS. (TAPE MOVING)
14	RCE	7,F2	RAM CHECKSUM ERROR. CAUSES FATAL CLASS 2 BECAUSE THE ERROR MIGHT HAVE OCCURRED DURING THE TRANSMISSION OF THE MESSAGE PACKET.
13-11			RESERVED (ALWAYS A 0)
10	WCF	7	WRITE CLOCK FAILURE. SET DURING A WRITE TO INDICATE THAT THE FIFO IS NOT BEING EMPTIED BY THE TRANSPORT.
09-08			RESERVED (ALWAYS A 0)
07-00	RL	-	REVISION LEVEL.
	7-0		

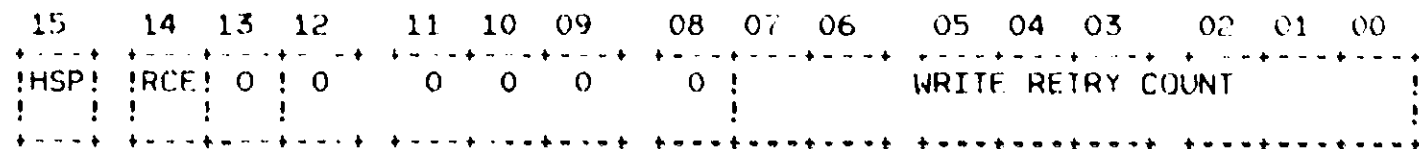
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821

6.3.6 EXTENDED STATUS REGISTER 3 (XSTAT3)



BIT	NAME	TCC	DEFINITION
15 TO 08			MICRO DIAGNOSTIC ERROR CODE. (SEE LIST OF CODES BELOW).
07			RESERVED (ALWAYS A 0)
06	OPI	6	OPERATION INCOMPLETE. SET WHEN A READ, SPACE, OR SKIP OPERATION HAS MOVED 25 FEET OF TAPE WITHOUT DETECTING ANY DATA ON THE TAPE.
05	REV	S	DIRECTION OF CURRENT OPERATION WAS REVERSE (BUT IS 0 IF REWIND OR FORWARD)
04	-	-	RESERVED (ALWAYS A 0)
03	DCK	S,6	DENSITY CHECK. SET WHEN A PE IDENTIFICATION BURST (IDB) WAS NOT DETECTED WHEN MOVING OFF OF BOT.
02-01			RESERVED (ALWAYS A 0)
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.

6.3.7 EXTENDED STATUS REGISTER 4 (XSTAT4)



BIT	NAME	TCC	DEFINITION
-----	------	-----	------------

1822	15	HSP	S	HIGH SPEED. WHEN SET, INDICATES THAT THE TRANSPORT IS OPERATING IN HIGH SPEED MODE.(100IPS) WHEN CLEAR, THE TRANSPORT IS OPERATING IN LOW SPEED MODE.(25IPS)
1823				
1824				
1825				
1826	14	RCE	6	RETRY COUNT EXCEEDED. WHEN SET, INDICATES THAT THE CONTROLLER WAS BUFFERING WRITE DATA AND COULD NOT SUCCESSFULLY OUTPUT THE BUFFERED RECORD WITHIN THE SPECIFIED NUMBER OF RETRIES. CAUSES TAPE POSITION LOST TERMINATION.
1827				
1828				
1829				
1830				
1831	13-8		-	RESERVED (ALWAYS A 0)
1832				
1833	7-0	WRC	S	WRITE RECOUNT COUNT STATISTIC. THIS FIELD INDICATES, WHEN THE CONTROLLER IS BUFFERING WRITE DATA RECORDS, THE TOTAL NUMBER OF CONTROLLER INITIATED RETRIES PERFORMED IN ORDER TO WRITE THE PREVIOUS BUFFERED RECORD. THIS COUNT IS CLEARED AFTER IT IS DISPLAYED.
1834				
1835				
1836				
1837				
1838				
1839				

7.0 DIAGNOSTIC HISTORY

REVISION A - MAR 1982
 - MODIFIED CZTSHC FROM TS11 FOR TSV05

REVISION B - APR 1983
 - UPDATED THE DIAGNOSTIC TO SUPPLY THE CORRECT RECORD NUMBER DURING EXECUTION OF TEST #2.
 REF: CHMIELECKI TO MITCHELL "TSV05 DATA RELIABILITY PROBLEM"; 21-JAN-83.

CVTSEBO => CNTSEAO JAKI BERG 9-APR-1984
 CHANGES WERE MADE TO CVTSEBO TO PRODUCE CNTSEAO FOR THE FALCON-PLUS PROJECT (SBC-11/21+). CHANGES, MARKED BY ";JB REV A-0", ARE:
 - SET THE ODT BREAK VECTOR (LOCATION 140) TO THE STARTING ADDRESS OF FALCON'S ODT ROM (170000-OCTAL).
 - LOWER THE GENERAL INTERRUPT PRIORITY FROM 7 TO 6.
 - CHANGE DEFAULT CSR ADDRESS FROM 172540 TO 176000.

1863

```

1
13
14          .TITLE PROGRAM HEADER AND TABLES
43          .SBTTL PROGRAM HEADER
45          .ENABL ABS,AMA
46          .      2000
48 002000   .      BGNMOD
49
50          ;++
51          ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
52          ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
53          ;--
54
55 002000   .      POINTER BGNRPT,BGNSW,BGNSFT,BGNAU,BGNDU
56
64
65 002000   .      HEADER CNTSE,A,0,5000,1
002000   L$NAME::          ;DIAGNOSTIC NAME
002000   103          .ASCII /C/
002001   116          .ASCII /N/
002002   124          .ASCII /T/
002003   123          .ASCII /S/
002004   105          .ASCII /E/
002005   000          .BYTE 0
002006   000          .BYTE 0
002007   000          .BYTE 0
002010   L$REV::          ;REVISION LEVEL
002010   101          .ASCII /A/
002011   L$DEPO::          ;0
002011   060          .ASCII /O/
002012   L$UNIT::          ;NUMBER OF UNITS
002012   000000       .WORD 0
002014   L$TIML::          ;LONGEST TEST TIME
002014   005000       .WORD 5000
002016   L$HPCP::          ;POINTER TO H.W. QUES.
002016   030002       .WORD L$HARD
002020   L$SPCP::          ;POINTER TO S.W. QUES.
002020   030110       .WORD L$SOFT
002022   L$HPTP::          ;PTR. TO DEF. H.W. PTABLE
002022   002174       .WORD L$HW
002024   L$SPTP::          ;PTR. TO S.W. PTABLE
002024   002204       .WORD L$SW
002026   L$LADP::          ;DIAG. END ADDRESS
002026   032000       .WORD L$LAST
002030   L$STA::          ;RESERVED FOR APT STATS
002030   000000       .WORD 0
002032   L$CO::          .WORD 0
002032   000000       .WORD 0
002034   L$DTYP::          ;DIAGNOSTIC TYPE
002034   000001       .WORD 1
002036   L$APT::          ;APT EXPANSION
002036   000000       .WORD 0
002040   L$DTP::          ;PTR. TO DISPATCH TABLE
002040   002124       .WORD L$DISPATCH
002042   L$PRIO::          ;DIAGNOSTIC RUN PRIORITY
002042   000000       .WORD 0
002044   L$ENVT::          ;FLAGS DESCRIBE HOW IT WAS SETUP

```

PROGRAM HEADER

002044	000000			.WORD	0
002046		L\$EXP1::	;EXPANSION WORD	.WORD	0
002046	000000			.WORD	0
002050		L\$MREV::	;SVC REV AND EDIT 0	.BYTE	C\$REVISION
002050	003			.BYTE	C\$EDIT
002051	003				
002052		L\$EF::	;DIAG. EVENT FLAGS	.WORD	0
002052	000000			.WORD	0
002054	000000			.WORD	0
002056		L\$SPC::		.WORD	0
002056	000000			.WORD	0
002060		L\$DEVP::	; POINTER TO DEVICE TYPE LIST	.WORD	L\$DVTYP
002060	002164			.WORD	L\$DVTYP
002062		L\$REPP::	;PTR. TO REPORT CODE	.WORD	L\$RPT
002062	017570			.WORD	L\$RPT
002064		L\$EXP4::		.WORD	0
002064	000000			.WORD	0
002066		L\$EXP5::		.WORD	0
002066	000000			.WORD	0
002070		L\$AUT::	;PTR. TO ADD UNIT CODE	.WORD	L\$AU
002070	024100			.WORD	L\$AU
002072		L\$DUT::	;PTR. TO DROP UNIT CODE	.WORD	L\$DU
002072	024026			.WORD	L\$DU
002074		L\$LUN::	;LUN FOR EXERCISERS TO FILL	.WORD	0
002074	000000			.WORD	0
002076		L\$DESP::	;POINTER TO DIAG. DESCRIPTION	.WORD	L\$DESC
002076	002136			.WORD	L\$DESC
002100		L\$LOAD::	;GENERATE SPECIAL AUTOLOAD EMT	EMT	E\$LOAD
002100	104035				
002102		L\$ETP::	;POINTER TO ERR TBL	.WORD	0
002102	000000			.WORD	0
002104		L\$ICP::	;PTR. TO INIT CODE	.WORD	L\$INIT
002104	021324			.WORD	L\$INIT
002106		L\$CCP::	;PTR. TO CLEAN-UP CODE	.WORD	L\$CLEAN
002106	023764			.WORD	L\$CLEAN
002110		L\$ACP::	;PTR. TO AUTO CODE	.WORD	L\$AUTO
002110	023342			.WORD	L\$AUTO
002112		L\$PRT::	;PTR. TO PROTECT TABLE	.WORD	L\$PROT
002112	021316			.WORD	L\$PROT
002114		L\$TEST::	;TEST NUMBER	.WORD	0
002114	000000			.WORD	0
002116		L\$DLY::	;DELAY COUNT	.WORD	0
002116	000000			.WORD	0
002120		L\$HIME::	;PTR. TO HIGH MEM	.WORD	0
002120	000000			.WORD	0
66					
72		.SBTTL	DISPATCH TABLE		
73					
74		---			
75		; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.			
76		; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.			
77		---			
78					
79	002122		DISPATCH 5	.WORD	5
	002122	000005		.WORD	5
	002124		L\$DISPATCH::	.WORD	T1
	002124	024204		.WORD	T1

PROGRAM HEADER AND TABLES
DISPATCH TABLE

MACRO M1200 10 MAR 84 08:53 PAGE 16 2

SEG 0041

	002126	025670				.WORD	T2
	002130	026526				.WORD	T3
	002132	026722				.WORD	T4
	002134	027102				.WORD	T5
80							
87							
88					.SBTTL	DESCRIPTIVE TEXT	
89							
90					***		
91					12	LINES OF TEXT PRINTED TO THE OPERATOR TO IDENTIFY THE DIAGNOSTIC AND THE DEVICE UNDER TEST	
92					***		
93							
94	002136				DESCRIPT	<DATA RELIABILITY TEST>	
ST/	002136				L\$DESC::		
	002136	104	101	124		.ASCIZ	/DATA RELIABILITY TE
	002141	101	040	122			
	002144	105	114	111			
	002147	101	102	111			
	002152	114	111	124			
	002155	131	040	124			
	002160	105	123	124			
	002163	000					
95	002164				DEVTYP	<TSV05>	.EVEN
	002164				L\$DVTYP::		
	002164	124	123	126		.ASCIZ	*TSV05*
	002167	060	065	000			
						.EVEN	
96					.SBTTL	DEFAULT HARDWARE P-TABLE	
97							
98					***		
99					;	THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF	
100					;	THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE	
101					;	IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.	
102					***		
103							
104	002172				BGNHW	DFPTBL	
	002172	000003				.WORD	L10000-L\$HW/2
	002174				L\$HW::		
	002174				DFPTBL::		
105							
106							
107	002174	176000			.WORD	176000	;\$DB ADDRESS. ;JB REV A-0
108	002176	000224			.WORD	224	;\$VECTOR ADDRESS.
109	002200	000000			.WORD	0	;\$DRIVE #C FOR DEFAULT
110							
111	002202				ENDHW		
	002202				L10000;		
112					.SBTTL	SOFTWARE P-TABLE	
113							
114					***		
115					;	THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM	
116					;	PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.	
117					***		
118							
119	002202				BGN\$W	SFPTBL	
	002202	000051				.WORD	L10001-L\$SW/2

SOFTWARE P-TABLE

002204		L\$SW::		
002204		SFPTBL::		
120				
127 002204	001	CLRFLG::	.BYTE 1	;CLEAR COUNTERS FLAG.
128 002205	000	RRANV::	.BYTE 0	;RESET RANDOM VARIABLES EACH PASS FLAG.
129 002206	000	MAE::	.BYTE 0	;HALT AFTER EACH COMMAND FLAG.
130 002207	000	ERCVFR::	.BYTE 0	;ENABLE RECOVERABLE ERROR PRINTS FLAG.
131 002210	001	BADTSW::	.BYTE 1	;BAD TAPE SWITCH TO REWRITE ON SAME SPOT & DETECT BAD TAPE
132 002211	000		.BYTE 0	;SPARE
133 002212	000	DINT::	.BYTE 0	;DISABLE INTERRUPTS FLAG.
134 002213	000	IREF::	.BYTE 0	;INHIBIT ERROR RECOVERY FLAG.
135 002214	000	CHGFLG::	.BYTE 0	;CHANGE CMD SEQ TABLE FLAG.
136 002215	000		.BYTE 0	;SPARE.
137 002216	000	PIRE::	.BYTE 0	;INHIBIT RESIDUAL FRAMECOUNT ERROR REPORT FLAG.
138 002217	000		.BYTE 0	;SPARE.
139 002220	000040	CHAR::	CH.EAI	;CHARACTERISTICS CODE (DEFAULT = 40).
140 002222	000015	CMDD::	.WORD 13.	;COMMAND 2 (DEFAULT = REWIND).
141 002224	000001		.WORD 1	;BYTE COUNT
142 002226	000001		.WORD 1	;NUMBER OF OPERATIONS
143 002230	000007		.WORD RAMP	;PATTERN
144 002232	000004		.WORD 4	;COMMAND 3 (DEFAULT = WRITE)
145 002234	004000		.WORD DATCNT	;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
146 002236	076400		.WORD 32000.	;NUMBER OF OPERATIONS (DEFAULT = 32000).
147 002240	000007		.WORD RAMP	;PATTERN (DEFAULT = RANDOM).
148 002242	000003		.WORD 3	;COMMAND 4 (DEFAULT = READ REV).
149 002244	000000		.WORD DATCNT	;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
150 002246	076400		.WORD 32000.	;NUMBER OF OPERATIONS (DEFAULT = 32,000).
151 002250	000007		.WORD RAMP	;PATTERN (DEFAULT = RANDOM).
152 002252	000002		.WORD 2	;COMMAND 5 (DEFAULT = READ FWD).
153 002254	004000		.WORD DATCNT	;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
154 002256	076400		.WORD 32000.	;NUMBER OF OPERATIONS (DEFAULT = 32,000).
155 002260	000007		.WORD RAMP	;PATTERN (DEFAULT = RANDOM).
156 002262	000015		.WORD 13.	;COMMAND 6 (DEFAULT = REWIND).
157 002264	000001		.WORD 1	;BYTE COUNT
158 002266	000001		.WORD 1	;NUMBER OF OPERATIONS
159 002270	000007		.WORD RAMP	;PATTERN
160 002272	000033		.WORD 27.	;END OF CMD SEQ TABLE CODE (DEF) OR CMD 7
161 002274	004000		.WORD DATCNT	;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
162 002276	076400		.WORD 32000.	;NUMBER OF OPERATIONS (DEFAULT = 32000).
163 002300	000007		.WORD RAMP	;PATTERN (DEFAULT = RANDOM).
164 002302	000033		.WORD 27.	;END OF CMD SEQ TABLE CODE (DEF) OR CMD 8
165 002304	004000		.WORD DATCNT	;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
166 002306	076400		.WORD 32000.	;NUMBER OF OPERATIONS (DEFAULT = 32000).
167 002310	000007		.WORD RAMP	;PATTERN (DEFAULT = RANDOM).
168 002312	000001	TS1MD::	.WORD 1	;DEFAULT SWITCH SETTING
169 002314	000000	RDBUF::	.WORD 0	;ENABLE READ BUFFERING
170 002316	000000	WTBUF::	.WORD 0	;ENABLE WRITE BUFFERING
171 002320	000000	HSSW::	.WORD 0	;RUN AT 100IPS SWITCH
172 002322	000000	EXTFEA::	.WORD 0	;EXTENDED FEATURES SOFTWARE SW 0=OFF;1=ON
173 002324	000000	BENBSW::	.WORD 0	;BUFFER ENABLE SOFTWARE SW 0=OFF;1=ON
174				
175 002326				
176				
177 002326		L10001:		
			ENDSW	
			ENDMOD	

SOFTWARE P-TABLE

```

190
191
192 .TITLE GLOBAL AREAS
201 .SBTTL GLOBAL EQUATES SECTION
202 002326 BGNMOD
203
204 ;++
205 ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
206 ; ARE USED IN MORE THAN ONE TEST.
207 ;--
208
209 002326 EQUALS
;
; BIT DIFINITIONS
;
100000 BIT15== 100000
040000 BIT14== 40000
020000 BIT13== 20000
010000 BIT12== 10000
004000 BIT11== 4000
002000 BIT10== 2000
001000 BIT09== 1000
000400 BIT08== 400
000200 BIT07== 200
000100 BIT06== 100
000040 BIT05== 40
000020 BIT04== 20
000010 BIT03== 10
000004 BIT02== 4
000002 BIT01== 2
000001 BIT00== 1
;
001000 BIT9== BIT09
000400 BIT8== BIT08
000200 BIT7== BIT07
000100 BIT6== BIT06
000040 BIT5== BIT05
000020 BIT4== BIT04
000010 BIT3== BIT03
000004 BIT2== BIT02
000002 BIT1== BIT01
000001 BIT0== BIT00
;
; EVENT FLAG DEFINITIONS
; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
;
; BIT POSITION IN SECOND STATUS WORD
000040 EF.START== 32. ; (100000) START COMMAND WAS ISSUED
000037 EF.RESTART== 31. ; (040000) RESTART COMMAND WAS ISSUED
000036 EF.CONTINUE== 30. ; (020000) CONTINUE COMMAND WAS ISSUED
000035 EF.NEW== 29. ; (010000) A NEW PASS HAS BEEN STARTED
000034 EF.PWR== 28. ; (004000) A POWER-FAIL/POWER-UP OCCURRED
;
; PRIORITY LEVEL DEFINITIONS
;
000340 PRI07== 340

```

GLOBAL EQUATES SECTION

```

000300      PRI06== 300
000240      PRI05== 240
000200      PRI04== 200
000140      PRI03== 140
000100      PRI02== 100
000040      PRI01== 40
000000      PRI00== 0
;
;OPERATOR FLAG BITS
;
000004      EVL==      4
000010      LOT==     10
000020      ADR==     20
000040      IDU==     40
000100      ISR==    100
000200      UAM==    200
000400      BOE==    400
001000      PNT==   1000
002000      PRI==   2000
004000      IXE==   4000
010000      JBE==  10000
020000      IER==  20000
040000      LOE==  40000
100000      HOE== 100000

210
218
219      ; REGISTER USAGE.
220      ;
221      ; R0 - PASSES PARAMETERS TO/FROM DIAGNOSTIC SUPERVISOR.
222      ; R1 - COMMAND SEQUENCE TABLE POINTER.
223      ; R2 - GENERAL PURPOSE REGISTER.
224      ; R3 - GENERAL PURPOSE REGISTER.
225      ; R4 - GENERAL PURPOSE REGISTER.
226      ; R5 - CURRENT LOGICAL DEVICE NUMBER X 2.
227      ; R6 - STACK POINTER.
228      ; R7 - PROGRAM COUNTER.
229
230      ;THE FOLLOWING ARE BIT DEFINITIONS FOR THE TSSR REGISTERS.
231
232      100000      TS.SC==100000      ;SPECIAL CONDITION BIT.
233      040000      TS.UPE==40000      ;UNIBUS PARITY ERROR
234      020000      TS.SPE==20000      ;SERIAL BUS PARITY ERROR.
235      010000      TS.RMR==10000      ;REGISTER MODIFICATION REFUSED.
236      004000      TS.NXM==4000      ;NON-EXISTENT MEMORY.
237      002000      TS.NBA==2000      ;NEED BUFFER ADDRESS.
238      001000      TS.A17==1000      ;BUS ADDRESS BIT 17.
239      000400      TS.A16==400      ;BUS ADDRESS BIT 16.
240      000200      TS.SSR==200      ;UNIT READY BIT.
241      000100      TS.OFL==100      ;OFF LINE.
242      177717      TSC.FCC==177717    ;FATAL CLASS CODE MASK.
243      177761      TSC.TCC==177761    ;TERMINATION CLASS CODE MASK.
244
245      ;THE FOLLOWING ARE BIT DEFINITIONS FOR THE COMMAND WORD
246
247      100000      ACK.C==100000      ;ACKNOWLEDGE BIT
248      040000      CVC.C==40000      ;CLEAR VOLUME CHECK.
249      020000      OPP.C==20000      ;OPPOSITE BIT

```

GLOBAL EQUATES SECTION

```

250      010000      SWB.C**10000      ;SWAP BYTE BIT
251      004000      MOD.C3**4000      ;MODE BIT 3
252      004000      BRFC**4000       ;BYTE/RECORD/FILE COUNT FLAG BIT, NOT USED
253      ;BY TS05 BUT USED INTERNALLY BY THIS PROGRAM ONLY.
254      002000      MOD.C2**2000      ;MODE BIT 2
255      001000      MOD.C1**1000      ;MODE BIT 1
256      000400      MOD.C0**400       ;MODE BIT 0
257      000200      IE.C**200        ;INTERRUPT ENABLE
258      000100      FMT.C1**100      ;FORMAT BIT 1
259      000100      VFY.C**100       ;WRITE VERIFY FLAG BIT. INTERNAL USE ONLY.
260      ;NOT USED BY TS05.
261      000040      FMT.C0**40        ;FORMAT BIT 0.
262      000040      JMP.C**40        ;JUMP BIT-TO DIRECT THIS PROGRAM TO JUMP TO
263      ;A CERTAIN LOCATION IN THE COMMAND SEQUENCE
264      ;TABLE. INTERNAL USE ONLY.
265      000020      CMD.C4**20        ;COMMAND BIT 4
266      000020      DLY.C**20        ;INSERT DELAY. INTERNAL USE ONLY.
267      000010      CMD.C3**10       ;COMMAND BIT 3
268      000004      CMD.C2**4        ;COMMAND BIT 2
269      000002      CMD.C1**2        ;COMMAND BIT 1
270      000001      CMD.C0**1        ;COMMAND BIT 0
271
272      ;BIT DEFINITIONS FOR DEVICE CHARACTERISTICS.
273
274      000200      CH.ESS**200       ;ENABLE SKIF TAPE MARKS STOP (STOP AT LOGICAL EOT).
275      000040      CH.EAI**40        ;ENABLE ATTENTION INTERRUPTS.
276      000020      CH.ERI**20       ;ENABLE MESSAGE BUFFER RELEASE INTERRUPTS.
277      000040      DFTSCH**CH.EAI    ;DEFAULT CHARACTERISTICS CODE.
278
279      ;BIT DEFINITIONS FOR EXTENDED CONTROL WORD
280
281      000040      EF.HSS**40        ;ENABLE HIGH SPEED SELECT
282      000030      EF.RWB**30       ;ENABLE BOTH READ & WRITE BUFFERING
283      000020      EF.RBO**20       ;ENABLE READ BUFFERING ONLY
284
285      ;THE FOLLOWING INDICATES THE RELATIVE POSITIONS OF THE STATUS WORDS
286      ;IN THE MESSAGE BUFFER.
287
288      000004      MS.RFC**4         ;RESIDUAL FRAME COUNT.
289      000006      MS.XS0**6        ;EXT STATUS REG 0
290      000010      MS.XS1**10       ;EXT STATUS REG 1
291      000012      MS.XS2**12       ;EXT STATUS REG 2
292      000014      MS.XS3**14       ;EXT STATUS REG 3
293      000016      MS.XS4**16       ;EXT STATUS REG 4
294
295      ;THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0.
296
297      100000      X0.TMK**100000     ;TAPE MARK.
298      040000      X0.RLS**40000     ;RECORD LENGTH SHORT.
299      020000      X0.LET**20000     ;LOGICAL EOT.
300      010000      X0.RLL**10000     ;RECORD LENGTH LONG.
301      000100      X0.ONL**100       ;ON LINE BIT.
302      000004      X0.WLK**4         ;WRITE LOCK BIT
303      000002      X0.BOT**2         ;BOT BIT.
304      000001      X0.EOT**1         ;EOT BIT.
305
306      ;THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2.

```

GLOBAL EQUATES SECTION

```

307
308      100000      X2.OPM==100000      ;OPERATION IN PROGRESS, TAPE MOVING
309      000200      X2.EFE==200        ;EXTENDED FEATURES ENABLED
310      000100      X2.BFE==100        ;BUFFERING ENABLED
311
312      ;THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3.
313
314      000010      X3.DCK==10          ;DENSITY CHECK.
315      157400      X3.RNY==157400      ;CAPSTAN RUNAWAY UDIAG ERROR CODE.
316
317      ;THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4.
318
319      100000      X4.HSS==100000      ;HIGH SPEED SWITCH INDICATING 100IPS
320      040000      X4.RCE== 40000      ;RETRY COUNT EXCEEDED
321
322
323      ;THE FOLLOWING DEFINITIONS SHOW THE RELATIVE POSITIONS OF THE COMMAND
324      ;PACKET ENTRIES.
325
326      000000      CP.CMD==0            ;CMDPKT+0==TS05 COMMAND.
327      000002      CP.ADL==2            ;CMDPKT+2==BUFFER ADDRESS LOW.
328      000004      CP.ADH==4            ;CMDPKT+4==BUFFER ADDRESS HIGH.
329      000006      CP.CNT==6            ;CKDPKT+6==BYTE/FILE/RECORD COUNT
330
331      ;MISCELLANEOUS DEFINITIONS.
332
333      000300      INTPRI==PRI06         ;PRIORITY TO BE USED IN INTERRUPT STATE.
334      000012      SCHCNT==12           ;ARBITRARY BYTE LENGTH FOR CHARACTERISTIC
335      ;BUFFER LENGTH. (EVEN #)
336      000020      MSGCNT==20           ;MESSAGE BUFFER LENGTH IN BYTES. (EVEN #)
337      000020      DIACNT==20           ;DIAGNOSTIC COMMAND BUFFER EXTENT.
338      004000      DATCNT==2048.        ;MAXIMUM RECORD LENGTH IN BYTES.
339      ;THIS COUNT SHOULD BE A MULTIPLE OF 256 TO INSURE
340      ;PROPER READ/WRITE BUFFER ALLOCATION BY THE SUPER.
341      177740      RNOPSC==177740       ;RANDOM # OF OPERATIONS MASK.
342      000007      RANP==7              ;CODE TO SELECT RANDOM PATTERN.
343      000020      RRECL==16.           ;READ RECOVERY ATTEMPT LIMIT.
344      000020      WRECL==16.           ;WRITE RECOVERY ATTEMPT LIMIT.
345      153624      RANBC==153624        ;CONSTANT USED TO RESET RANDOM # GENERATOR BASE.
346      052561      RANSC==32561        ;CONSTANT USED TO RESET RANDOM # SAVE LOCATION.
347      177774      NINUSE==177774      ;NOT IN USE CODE FOR DEVICE STATE TABLE.
348      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
349      ;NOT "COMMAND" BITS.
350
351      ;THE FOLLOWING DEFINES THE COMMAND WORD FOR EACH TS05 COMMAND.
352
353      100013      DRI==  ACK.C!CMD.C3!CMD.C1!CMD.CO
354      ;DRIVE INIT.
355
356      104001      RDF==  ACK.C!BRF.C!CMD.CO
357      ;READ FORWARD
358
359      104401      RDR==  ACK.C!BRF.C!MOD.CO!CMD.CO
360      ;READ REVERSE
361
362      104005      WRT==  ACK.C!BRF.C!CMD.CO!CMD.C2
363      ;WRITE COMMAND

```


GLOBAL EQUATES SECTION

```

421      140006      WSM**  ACK.C!CVC.C!CMD.C2!CMD.C1      ;WRITE SUB-SYS MEM
422
423      100006      DIA**  ACK.C!CMD.C2!CMD.C1      ;DIAGNOSTICS.
424
425      000040      JMP**  JMP.C                      ;JUMP TO "N"TH COMMAND
426
427      000020      DLY**  DLY.C                      ;DELAY "N" MS.
428
429      177777      END**  177777                    ;END OF COMMAND SEQUENCES
430

```

```

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

```

```

431
432
433
434
435
436
437
438
439
440      ;          COMMAND PACKET.
441
442      .          *          <.,+3>&177774      ;MUST BE ON MOD 4 BOUNDRY.
443      CMDPKT::  0          ;1ST WORD IS ISO5 COMMAND.
444      0          ;2ND WORD IS THE BUFFER LOW ADDRESS.
445      0          ;3RD WORD IS THE BUFFER HIGH ADDRESS.
446      0          ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.
447

```

```

448      ;          GET STATUS COMMAND PACKET.
449
450      .          *          <.,+3>&177774      ;MUST BE ON MOD 4 BOUNDRY.
451      GSCP::   .WORD      GES
452

```

```

453
454      ;          MESSAGE BUFFER RELEASE COMMAND PACKET.
455
456      .          *          <.,+3>&177774      ;MUST BE ON MOD 4 BOUNDRY.
457      BRCPK::  .WORD      MBR
458

```

```

459
460      ;          REWIND COMMAND PACKET (USED IN ERROR RECOVERY ONLY)
461
462      .          *          <.,+3>&177774      ;MUST BE ON MOD 4 BOUNDRY.
463      RWCPK::  .WORD      RWD
464      1          .WORD      1
465

```

```

466
467      ;          WORK AREA FOR ANALYSIS OF MESSAGE PACKET CONTENTS.
468
469      MSGPKT::  .BLKW     8,
470
471      ;1ST WORD:: MESSAGE TYPE.
472      ;2ND WORD:: DATA FIELD LENGTH.
473      ;3RD WORD:: RESIDUAL FRAME COUNT.
474      ;4TH WORD:: XSTAT0
475      ;5TH WORD:: XSTAT1
476      ;6TH WORD:: XSTAT2
477

```

GLOBAL DATA SECTION

```

478                                     ;7TH WORD:: XSTAT3
479                                     ;8TH WORD:: XSTAT4
480                                     ;
481                                     ; MESSAGE PACKETS.
482 002374 MSGPK0:: .BLKW 8.                ;MESSAGE PACKET FOR DEVICE #0
483 002414 MSGPK1:: .BLKW 8.                ;MESSAGE PACKET FOR DEVICE #1
484 002434 MSGPK2:: .BLKW 8.                ;MESSAGE PACKET FOR DEVICE #2
485 002454 MSGPK3:: .BLKW 8.                ;MESSAGE PACKET FOR DEVICE #3
486
487                                     ;
488                                     ; SET CHARACTERISTIC BLOCK.
489 002474 002374 SCHBK:: MSGPK0            ;1ST WORD:: MSGPKT ADDR LO(SET UP BY EXECUTE ROUTINE).
490 002476 000000 0                        ;2ND WORD:: MSGPKT ADDR HI.
491 002500 000020 MSGCNT                    ;3RD WORD:: MSG BUFFER LENGTH (BYTES)
492 002502 000040 CH.EAI                    ;4TH WORD:: CHARACTERISTICS WORD(SET BY SETUP ROUTINE).
493 002504 000000 0                        ;5TH WORD:: HSP & BUFFER CONTROL ON EXT'D FEATURES
494
495                                     ;
496                                     ; WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.
497
498 002506 000000 WSMBK:: 0                  ;1ST WORD:: SEL 0
499 002510 000000 0                        ;2ND WORD:: SEL 2
500 002512 000000 0                        ;3RD WORD:: SEL 4
501
502                                     ;
503                                     ; TS05 REGISTER ADDRESSES.
504 002514 TSDB:: .BLKW 4                    ;TS05 DATA BUFFER ADDRESSES.
505 002524 TSSR:: .BLKW 4                    ;TS05 STATUS REGISTER ADDRESSES.
506 002534 TSVCT:: .BLKW 4                  ;TS05 VECTOR ADDRESSES.
507 002514 TSBA==TSDB                       ;DATA BUFFER ADDRESS REGISTER.
508
509                                     ;
510                                     ; ADDRESSES OF MESSAGE PACKETS.
511
512 002544 002374 MSGPKA:: MSGPK0            ;DEVICE 0.
513 002546 002414 MSGPK1                    ;DEVICE 1.
514 002550 002434 MSGPK2                    ;DEVICE 2.
515 002552 002454 MSGPK3                    ;DEVICE 3.
516
517                                     ;
518                                     ; ADDRESSES OF INTERRUPT HANDLING ROUTINES.
519 002554 010034 TSSINT:: TSSIN0            ;DEVICE 0.
520 002556 010042 TSSIN1                    ;DEVICE 1.
521 002560 010050 TSSIN2                    ;DEVICE 2.
522 002562 010056 TSSIN3                    ;DEVICE 3.
523
524                                     ;
525                                     ; TS05 CODE LEVELS, WILL BE STORED AFTER SCH CMD IN BASIC FUNCTION TEST
526 002564 000000 TS5CL:: 0                 ;DEVICE 0
527 002566 000000 0                         ;DEVICE 1
528 002570 000000 0                         ;DEVICE 2
529 002572 000000 0                         ;DEVICE 3
530
531                                     ;
532                                     ; TS05 EXT. FEA & BUF. ENA SW'S, WILL BE STORED AFTER SCH CMD IN BASIC FUNCTION TEST
533 002574 000000 TS5SW:: 0                 ;DEVICE 0
534 002576 000000 0                         ;DEVICE 1

```

GLOBAL DATA SECTION

```

535 002600 000000          0          ;DEVICE 2
536 002602 000000          0          ;DEVICE 3
537
538 ; UNIT NUMBERS OF ALL DEVICES BEING TESTED(1-4).
539 ; WHEN DEVICE IS NOT IN USE, IT'S LOCATION WILL = -5.
540 ; R5 WILL ALWAYS CONTAIN THE PRESENT LOGICAL UNIT NUMBER X 2.
541
542 002604 177774  DEVTBL: .WORD NINUSE
543 002606 177774          .WORD NINUSE
544 002610 177774          .WORD NINUSE
545 002612 177774          .WORD NINUSE
546 002614 177777          .WORD END
547
548
549 ; BAD TAPE TABLE POINTER; USED BY WRITE RETRY ROUTINE
550 ; "WRTY" TO LOG BAD TAPE SPOTS ON UNITS UNDER TEST
551
552 002616 003046  BTADDR: BTO
553 002620 003120          BT1
554 002622 003172          BT2
555 002624 003244          BT3
556 ; COUNTER AREA.
557
558 002626  CNTBGN=.
559 002626  WRBC: .BLKW 20          ;BYTES WRITTEN.
560 002666  RRBC: .BLKW 20          ;BYTES READ REV.
561 002726  RFBC: .BLKW 20          ;BYTES READ FWD.
562 002766  WRREC: .BLKW 4          ;RECOVERABLE WRITE ERRORS.
563 002776  WRUNR: .BLKW 4          ;UNRECOVERABLE WRITE ERRORS.
564 003006  RRREC: .BLKW 4          ;RECOVERABLE READ REV ERRORS.
565 003016  RRUNR: .BLKW 4          ;UNRECOVERABLE READ REV ERRORS.
566 003026  RFREC: .BLKW 4          ;RECOVERABLE READ FWD ERRORS.
567 003036  RFUNR: .BLKW 4          ;UNRECOVERABLE READ FWD ERRORS.
568 003046  BTO: .BLKW 21          ;UNIT 0 BAT TAPE SPOTS LOG
569 003120  BT1: .BLKW 21          ;UNIT 1 BAT TAPE SPOTS LOG
570 003172  BT2: .BLKW 21          ;UNIT 2 BAT TAPE SPOTS LOG
571 003244  BT3: .BLKW 21          ;UNIT 3 BAT TAPE SPOTS LOG
572 003316  WRTYCT: .BLKW 4          ;WRITE RETRY COUNTER
573 003326  PASCNT: .BLKW 4          ;PASS COUNT.
574 003336  SCCNT: .BLKW 4          ;SPECIAL CONDITION COUNT.
575 003346  VFYCNT: .BLKW 4          ;COUNT OF TS05 DATA COMPARE ERRORS.
576 003356  HRDCNT: .BLKW 4          ;COUNT OF HARD ERRORS.
577 003366  FTLCNT: .BLKW 4          ;COUNT OF FATAL ERRORS.
578 003376  CNTEND=.          ;END OF STATISTICAL COUNTERS.
579 003376  RECCNT: .BLKW 4          ;NUMBER OF RECORDS FROM BOT: CLEARED ON REWIND
580 ; AND WHEN RESTARTING OR CONTINUING TEST 2.
581 000550  CNTLEN=*CNTEND-CNTBGN  ;LENGTH OF STATISTICAL COUNTER AREA.
582
583
584 ; THE FOLLOWING ARE THE DEFINITIONS OF VARIABLES
585 ; USED BY THE PROGRAM.
586
587 003406 000000  DATAWT: .WORD 0          ;WRITE BUFFER ADDRESS.
588 003406 003406  DIABLK=*DATAWT  ;WRITE BUFFER ALSO USED FOR DIAG CMD.
589 003410 000000  DATARD: .WORD 0          ;READ BUFFER ADDRESS.
590 003412 000000  NCNT: .WORD 0          ;STORAGE FOR VALUE OF N.
591 003414 000000  NCNT1: .WORD 0          ;TEMP STORAGE FOR VALUE OF N.

```

GLOBAL DATA SECTION

```

572 003416 000000 BRFcnt:: .WORD 0 ; STORAGE FOR BPCR VALUE.
593 003420 177777 CMOWRD:: .WORD END ; CONTAINS COMMAND WORD BEING EXECUTED PRESENTLY.
594 003422 177777 CMOSAV:: .WORD END ; SAVE LOCATION FOR CMD WORD DURING ERROR RECOVERY
595 003424 177777 PCMCWD:: .WORD END ; CONTAINS PREVIOUS COMMAND WORD.
596 003426 000000 CMDLG:: .WORD 0 ; CURRENT COMMAND LOGGING CODE.
597 003430 000000 LENMSK:: .WORD 0 ; RANDOM WRITE LENGTH MASK, TO BE SET UP BY TESTS
598 003432 153624 RANB:: .WORD 153624 ; RANDOM # GENERATOR BASE.
599 003434 032561 RANS:: .WORD 32561 ; RANDOM # SAVE LOCATION.
600 003436 000000 TIME1:: .WORD 0 ; TIME COUNT 1.
601 003440 000000 TIME2:: .WORD 0 ; TIME COUNT 2.
602 003442 000000 JLOOP:: .WORD 0 ; JMP COMMAND LOOP COUNT.
603 003444 000000 JLOC:: .WORD 0 ; JMP COMMAND LOCATION COUNT.
604 003446 000000 PATERN:: .WORD 0 ; PATTERN SELECT CODE.
605 003450 000000 CTCC:: .WORD 0 ; CURRENT TERMINATION CLASS CODE.
606 003452 000000 RSSAVE:: .WORD 0 ; LOCATION FOR SAVING CURRENT DEVICE POINTER.
607 003454 000000 TSSREG:: .WORD 0 ; CURRENT STATUS REGISTER.
608 003456 000000 WTMFLG:: .WORD 0 ; WRITE TAPE MARK FLAG
609
610 ; ERROR FLAG AREA, THESE FLAGS ARE CLEARED DURING INITIALIZATION AND
611 ; AFTER EACH COMMAND IS COMPLETED.
612
613 003460 BGNFLG=,
614 003460 000000 RETRYC:: .WORD 0 ; # OF RECOVERY ATTEMPTS EXECUTED.
615 003462 000 RPTCNT:: .BYTE 0 ; WRITE REPEAT ON SAME SPOT CNTR: 4 PER WRITE RETRY
616 003463 000 WRTYFG:: .BYTE 0 ; WRITE RETRY ON SAME SPOT IN PROGRESS FLAG
617 003464 000 WRTYER:: .BYTE 0 ; WRITE RETRY ON SAME SPOT ERROR FLAG
618 003465 000 RECLOG:: .BYTE 0 ; RECORD COUNT HAS BEEN UPDATED FOR THIS RECORD.
619 003466 000 ERLOG:: .BYTE 0 ; DATA BYTES AND ERRORS HAVE BEEN LOGGED FOR THIS RECORD.
620 003467 000 RWERR:: .BYTE 0 ; READ/WRITE ERROR HAS OCCURED.
621 003470 000 UNREC:: .BYTE 0 ; UNRECOVERABLE ERROR HAS OCCURED.
622 003471 000 ERRREC:: .BYTE 0 ; ERROR RECOVERY MODE.
623 .EVEN
624 003472 ENDERF=,
625
626
627 ; ADDITIONAL FLAGS, THESE FLAGS ARE CLEARED DURING INITIALIZATION.
628
629 003472 INTFLG:: .BLKW 4 ; INTERRUPT OCCURRED FLAGS FOR EACH DEVICE.
630 003502 EOTFLG:: .BLKW 4 ; EOT/BOT FLAGS FOR EACH DEVICE (XSTATO).
631 003512 000000 BTPT:: .WORD 0 ; BAD TAPE SPOT POINTER TO BTO-BT3 VIA BTADDR
632 003514 000 EXPBCT:: .BYTE 0 ; BOT IS EXPECTED, DO NOT ABORT ON BOT/FUNC RTI.
633 003515 000 RANDOM:: .BYTE 0 ; RANDOM EVERYTHING FLAG.
634 003516 000 VFYFLG:: .BYTE 0 ; SET DURING WRITE/VERIFY COMMAND.
635 003517 000 RPTFLG:: .BYTE 0 ; PERFORMANCE REPORT HAS BEEN REQUESTED.
636 003520 000 SWBFLG:: .BYTE 0 ; ENABLES SWAP BYTE FUNCTION WHEN NOT EQUAL TO ZERO.
637 003521 000 IRE:: .BYTE 0 ; INHIBIT RESIDUAL FRAME COUNT ERROR REPORT.
638 003522 000 DROPED:: .BYTE 0 ; CURRENT UNIT HAS BEEN DROPPED
639 003523 000 T1SWB:: .BYTE 0 ; TEST1 SWAP BYTES FLAG
640 003524 000 ALLEOT:: .BYTE 0 ; ALL UNITS @ EOT FLAG
641 003525 000 ERSFLG:: .BYTE 0 ; ERASE FLAG: DO ERASE AFTER A SPACE REV TO DELETE
642 ; BADLY WRITTEN RECORD. 1 TO 4 ERASES LEAVING
643 ; A 3 TO 12 INCH GAP MAY RESULT.
644 .EVEN
645 003526 ENDFLG=,
646
647 ; ADDITIONAL FLAGS, THESE FLAGS ARE CLEARED ONLY AFTER BEING CHECKED.
648

```

GLOBAL DATA SECTION

```

649 003526      000      STAF LG:: .BYTE 0      ;START FLAG - SET BY INIT CODE IF STARTING.
650 003527      000      PWRFLG:: .BYTE 0      ;POWER FAILURE FLAG - SET ONLY DURING INIT.
651 003530      000      TRAPD4:: .BYTE 0      ;TRAPED AT 4 FLAG
652 003531      000      MISCFG:: .BYTE 0      ;MISCELLANEOUS FLAG
653 003532      000000    TSUNT:: .WORD 0      ;NUMBER OF THE UNIT UNDER TEST PLUS HSSP&BUF
654 003534      000000    TSNP:: .WORD 0      ;FOR PRINT OUT UNIT # ONLY
655
656              ;      OPERATOR FLAG SETTINGS PASSED BY DIAG. SUPERVISOR IN A 16 BIT WORD
657              ;      SEE GLOBAL EQUATES SECTION FOR FLAG BIT LIST
658
659 003536      000000    OPFLAG:: .WORD 0      ;READ ONLY OPERATOR FLAG WORD
660              .EVEN
661
662              ;THE FOLLOWING IS THE COMMAND SEQUENCE TABLE. THE TABLE
663              ;HAS DEFAULT VALUES AT PROGRAM LOAD AS SHOWN. THESE VALUES
664              ;CAN BE UPDATED BY A TEST OR BY OPERATOR INPUT.
665
666 003540      140004    CMDSEQ:: .WORD SCH      ;SET CHARACTERISTICS.
667 003542      000040    .WORD CH.EAI
668 003544      000001    .WORD 1
669 003546      000000    .WORD 0
670 003550      102010    CMDSEQ2:: .WORD RWD      ;REWIND.
671 003552      000001    .WORD 1      ;BYTE COUNT.
672 003554      000001    .WORD 1      ;ONCE.
673 003556      000007    .WORD RANP     ;PATTERN.
674 003560      104005    .WORD WRT      ;WRITE.
675 003562      004000    .WORD DATCNT   ;MAX BUFFER LENGTH.
676 003564      076400    .WORD 32000.   ;32,000 RECORDS.
677 003566      000007    .WORD RANP     ;RANDOM PATTERN.
678 003570      104401    .WORD RDR      ;READ REV.
679 003572      004000    .WORD DATCNT   ;MAX BUFFER LENGTH.
680 003574      076400    .WORD 32000.   ;32,000 RECORDS
681 003576      000007    .WORD RANP     ;RANDOM PATTERN.
682 003600      104001    .WORD RDF      ;READ FWD.
683 003602      004000    .WORD DATCNT   ;MAX BUFFER LENGTH.
684 003604      076400    .WORD 32000.   ;32,000 RECORDS.
685 003606      000007    .WORD RANP     ;RANDOM PATTERN.
686 003610      102010    .WORD RWD      ;REWIND.
687 003612      000001    .WORD 1      ;BYTE COUNT.
688 003614      000001    .WORD 1      ;ONCE.
689 003616      000007    .WORD RANP     ;PATTERN.
690 003620      .BLKW 40. ;EXTENSION TO DOUBLE BUFFER SIZE
691 003740      177777    SEQEND:: .WORD END      ;SOFT END OF SEQUENCE TABLE.
692 003742      177777    .WORD END
693 003744      177777    .WORD END
694 003746      177777    .WORD END
695 003750      177777    .WORD END      ;HARD END OF SEQUENCE TABLE.
696
697              ;THE FOLLOWING IS THE TS05 COMMAND TABLE
698 003752      100013    CMDTBL:: .WORD DRI     ;DRIVE INIT.
699 003754      104001    .WORD RDF      ;READ FORWARD.
700 003756      104401    .WORD RDR      ;READ REVERSE.
701 003760      104005    .WORD WRT      ;WRITE
702 003762      104105    .WORD WTV      ;WRITE/VERIFY. (WRITE ALL RECORDS, RDR AND
703              ;CHECK DATA ON ALL RECORDS, RDF AND
704              ;CHECK DATA ON ALL RECORDS.)
705 003764      104010    .WORD .ZF      ;SPACE "N" RECORDS FORWARD.

```

GLOBAL DATA SECTION

706	003766	104410	.WORD	SRR	SPACE "N" RECORDS REVERSE.
707	003770	105401	.WORD	RNR	READ NEXT REVERSE. I.E., SPACE FWD, READ REVERSE.
708	003772	125401	.WORD	RNF	READ NEXT FORWARD, I.E., READ FORWARD, SPACE REVERSE.
709	003774	105001	.WORD	RPF	READ PREVIOUS FORWARD, I.E., SPACE REVERSE, READ FORWARD
710	003776	125001	.WORD	RPR	READ PREVIOUS REVERSE. I.E., READ REVERSE, SPACE FORWARD
711	004000	105005	.WORD	WRR	WRITE RETRY.
712	004002	107010	.WORD	RWD	REWIND.
713	004004	100012	.WORD	MBR	MESSAGE BUFFER RELEASE
714	004006	100011	.WORD	WTM	WRITE TAPE MARK
715	004010	101011	.WORD	WTR	WRITE TAPE MARK RETRY.
716	004012	105010	.WORD	SFF	SPACE "N" FILES FORWARD.
717	004014	105410	.WORD	SFR	SPACE "N" FILES REVERSE.
718	004016	100017	.WORD	GES	GET EXTENDED STATUS.
719	004020	100411	.WORD	ERS	ERASE 3 INCHES OF TAPE.
720	004022	100412	.WORD	UNL	REWIND AND UNLOAD.
721	004024	101012	.WORD	CLN	CLEAR TAPE.
722	004026	14C004	.WORD	SCH	SET CHARACTERISTICS.
723	004030	100006	.WORD	DIA	DIAGNOSTIC COMMAND.
724	004032	000040	.WORD	JMP	JUMP TO THE NTH COMMAND IN THE SEQUENCE.
725	004034	000020	.WORD	DLY	DELAY "N" MS.
726	004036	177777	.WORD	END	END OF COMMAND TABLE

THE FOLLOWING TABLE CONTAINS THE ASCII FOR EACH COMMAND.

730	004040	104	122	111	CMDASC:: .ASCII /DRI/	DRIVE INIT.
731	004043	122	104	106	.ASCII /RDF/	READ FORWARD.
732	004046	122	104	122	.ASCII /RDR/	READ REVERSE.
733	004051	127	122	124	.ASCII /WRT/	WRITE
734	004054	127	124	126	.ASCII /WTV/	WRITE/VERIFY. (WRITE ALL RECORDS, RDR AND CHECK DATA
735						ON ALL RECORDS, RDF AND CHECK DATA ON ALL RECORDS.)
736	004057	123	122	106	.ASCII /SRF/	SPACE "N" RECORDS FORWARD.
737	004062	123	122	122	.ASCII /SRR/	SPACE "N" RECORDS REVERSE.
738	004065	122	116	122	.ASCII /RNR/	READ NEXT REVERSE. I.E., SPACE FWD READ REVERSE.
739	004070	122	116	106	.ASCII /RNF/	READ NEXT FORWARD, I.E., READ FORWARD, SPACE REVERSE.
740	004073	122	120	106	.ASCII /RPF/	READ PREVIOUS FORWARD, I.E., SPACE REVERSE, READ FORWARD
741	004076	122	120	122	.ASCII /RPR/	READ PREVIOUS REVERSE. I.E., READ REVERSE, SPACE FORWARD
742	004101	127	122	122	.ASCII /WRR/	WRITE RETRY.
743	004104	122	127	104	.ASCII /RWD/	REWIND.
744	004107	115	102	122	.ASCII /MBR/	MESSAGE BUFFER RELEASE
745	004112	127	124	115	.ASCII /WTM/	WRITE TAPE MARK
746	004115	127	124	122	.ASCII /WTR/	WRITE TAPE MARK RETRY.
747	004120	123	106	106	.ASCII /SFF/	SPACE "N" FILES FORWARD.
748	004123	123	106	122	.ASCII /SFR/	SPACE "N" FILES REVERSE.
749	004126	107	105	123	.ASCII /GES/	GET EXTENDED STATUS.
750	004131	105	122	123	.ASCII /ERS/	ERASE 3 INCHES OF TAPE.
751	004134	125	116	114	.ASCII /UNL/	REWIND AND UNLOAD.
752	004137	103	114	116	.ASCII /CLN/	CLEAR TAPE.
753	004142	123	103	110	.ASCII /SCH/	SET CHARACTERISTICS. WHERE BRF=200, 40, 20, 0.
754						SEE TSV05/TS05 PROGRAMMING SPECIFICATION FOR DESCRIPTION.
755	004145	104	111	101	.ASCII /DIA/	DIAGNOSTICS. SEE TSV05/TS05 PROGRAMMING SPECIFICATION
756						FOR DESCRIPTION. ODT MUST BE USED TO LOAD DIAGNOSTIC DATA
757						INTO THE WRITE BUFFER BEFORE THIS CMD IS ISSUED.
758	004150	112	115	120	.ASCII /JMP/	JUMP TO THE NTH COMMAND IN THE COMMAND
759						SEQUENCE TABLE, WHERE N IS DEFINED IN
760						THE # OF OPERATIONS.
761	004153	104	114	131	.ASCII /DLY/	DELAY "N" MS, WHERE N IS DEFINED IN
762						THE # OF OPERATIONS.

GLOBAL DATA SECTION

```

763 004156      105      116      104      .ASCII /END/          ;END OF COMMAND SEQUENCE.
764              .EVEN
765
766
767
768              .SBTTL GLOBAL TEXT SECTION
769
770
771              ;**
772              ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
773              ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
774              ; MORE THAN ONE TEST.
775              ;--
776
777
778              ;
779              ; FORMAT STATEMENTS USED IN PRINT CALLS
780              ;
781
782              .NLIST BEX
783
784
785
786
787
788
789 004162      045      116      045 CODELM:: .ASCIZ /%AUNIT %D1%A TSV05 CODE LEVEL %O3%N%N/
790 004231      045      116      045 SWSET:: .ASCIZ /%AUNIT %D1%A TSV05 SWITCH SETTINGS %O3%N%N/
791              .EVEN
792 004306      130      130      150 HALTM:: .ASCIZ /XXX CMD - TYPE <CR> TO CONTINUE/
793 004346      103      115      104 CMDPKM:: .ASCIZ /CMD PACKET ADR NOT ON MODULO 4 BOUNDARY: RELOAD!/
794              .EVEN
795 004430      104      101      124 WTVRM:: .ASCIZ /DATA COMPARE ERROR/
796 004453      116      117      040 TOERM:: .ASCIZ /NO TSV05 RESPONSE/
797 004475      125      116      104 SCERM:: .ASCIZ /UNDEFINED SPEC COND/
798 004521      122      106      103 RFCERM:: .ASCIZ /RFC NON ZERO/
799 004536      124      123      126 NSSRM:: .ASCIZ /TSV05 NOT READY/
800 004556      122      105      124 RLEXM:: .ASCIZ /RETRY LIMIT EXCEEDED/
801 004603      104      122      111 ATYRM:: .ASCIZ /DRIVE OFF LINE/
802 004622      106      125      116 FUNRM:: .ASCIZ /FUNCTION REJECT/
803 004642      106      101      124 FATSM:: .ASCIZ /FATAL SUBSYSTEM ERROR/
804 004670      116      117      040 NOINTM:: .ASCIZ /NO INTERRUPT/
805 004705      124      101      120 TSAM:: .ASCIZ /TAPE STATUS ALERT/
806 004727      124      117      117 TOOMM:: .ASCIZ /TOO MANY INTERRUPTS/
807 004753      103      101      120 RNYM:: .ASCIZ /CAPSTAN RUNAWAY-GET STATUS RESULTS:/
808 005017      122      105      103 RERM:: .ASCIZ /RECOVERABLE ERROR/
809 005041      125      116      122 URERM:: .ASCIZ /UNRECOVERABLE ERROR/
810 005065      045      116      045 DROPDM:: .ASCIZ /%A DROPPED UNIT %D1%N/
811 005114      045      116      045 AUDRPM:: .ASCIZ /%A ALL UNITS DROPPED%N%N/
812 005146      045      116      045 AUDRUN:: .ASCIZ /%A DIAGNOSTIC ONLY SUPPORTS ONE CONTROLLER%N%N/
813 005226      045      116      045 DTAER2:: .ASCIZ "%A BYTE: %D4%S2%AWAS: %B%S2%AS/B: %B%N"
814 005275      045      104      064 DTAER3:: .ASCIZ "%A BYTES IN ERROR OUT OF %D4%N"
815 005337      045      101      116 DTAER4:: .ASCIZ /%A NO DATA READ%N/
816 005360      045      101      122 DTAER5:: .ASCIZ /%A RECORD TOO LONG: >%D4%A BYTES%N/
817 005422      045      101      122 MURTY1:: .ASCIZ /%A RECOVERED ON RETRY %D2%N/
818 005456      045      101      101 OFLIM:: .ASCIZ /%A DRIVE %D1%A OFF LINE%N/
819 005507      045      101      107 GETSTM:: .ASCIZ /%A GET STATUS CMD RESULTS: %N/
820 005543      045      116      045 NODEV:: .ASCII /%A BUS TRAP AT %O6%N/
821 005570      045      101      111 .ASCIZ /%A INTERFACE BAD OR TSOB NOT SET TO ABOVE ADDRESS%N/
822 005653      040      052      052 UNILK:: .ASCIZ /*****TAPE IS WRITE-LOCKED AND WILL CAUSE ERRORS*****/
823 005741      045      116      000 CRLF:: .ASCIZ /%N/
824 005744      045      116      045 CRLFSP:: .ASCIZ /%NS7/
825              .LIST BEX

```

GLOBAL TEXT SECTION

```

826          .EVEN
827          .SBTTL  GLOBAL ERROR REPORT SECTION
828
829          ***
830          ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX CALLS
831          ; THAT ARE USED IN MORE THAN ONE TEST. IT ALSO INCLUDES THE ASCII MESSAGES
832          ; THAT ARE USED BY THE PRINTB AND PRINTX CALLS..
833          ;--
834
835
836          BGNMSG  DTAERM
005752      DIAERM:
842          DATERM:;PRINTB  @STAER1,TSNP,PASCNT(R5),RECCNT(R5)
005752      016546  003376
005756      016546  003326
005762      013746  003534
005766      012746  006436
005772      012746  000004
005776      010600
006000      104414
006002      062706  000012
843          PRINTB  @STAER7
006006      012746  006530
006012      012746  000001
006016      010600
006020      104414
006022      062706  000004
844          MOV     R2,RECRD          ;SAVE R2
845          MOV     R3,TIME1         ;SAVE R3
846          MOV     R4,TIME2         ;SAVE R4
847          JSR    PC,RECTAP          ;RETRIEVE RECORD READ
848          MOV     RECRD,R2         ;RESTORE R2
849          MOV     R3,RECRD         ;SAVE RECORD READ
850          MOV     TIME1,R3         ;RESTORE R3
851          MOV     TIME2,R4         ;RESTORE R4
852          PRINTB  @STAER6,RECRD    ;PRINT RECORD READ
006066      013746  007066
006072      012746  006562
006076      012746  000002
006102      010600
006104      104414
006106      062706  000006
853          EXIT    MSG
006112      000167
006114      000000
854          .EVEN
855
856          ENDMMSG
L10002:
006116      104423
857          TRAP   C3MSG
858          BGNMSG  STAERM
006120      STAERM:
859          STAERM:;PRINTB  @STAER1,TSNP,PASCNT(R5),RECCNT(R5)
006120      016546  003376
006124      016546  003326
006130      013746  003534

```

15

SEQ 0056

GLOBAL ERROR REPORT SECTION

	006134	012746	006436					MOV	*STAER1,-(SP)
	006140	012746	000004					MOV	*4,-(SP)
	006144	010600						MOV	SP,R0
	006146	104414						TRAP	C:PNTB
	006150	062706	000012					ADD	*12,SP
860	006154			PRINTB	*STAER7				
	006154	012746	006530					MOV	*STAER7,-(SP)
	006160	012746	000001					MOV	*1,-(SP)
	006164	010600						MOV	SP,R0
	006166	104414						TRAP	C:PNTB
	006170	062706	000004					ADD	*4,SP
861	006174	013702	002330	MOV	CMDPKT,R2				
862	006200	042702	177740	BIC	*177740,R2				
863	006204	005302		DEC	R2				
864	006206	005702		TST	R2				;IF CMD IS A READ
865	006210	001016		BNE	50000*				
866	006212	004737	010064	JSR	PC,RECTAP				;THEN RETRIEVE
867	006216	010337	010064	MOV	R3,RECTAP				;AND
868	006222			PRINTB	*STAER6,RECRED				;TYPE RECORD READ
	006222	013746	007066					MOV	RECRED,-(SP)
	006226	012746	006562					MOV	*STAER6,-(SP)
	006232	012746	000002					MOV	*2,-(SP)
	006236	010600						MOV	SP,R0
	006240	104414						TRAP	C:PNTB
	006242	062706	000006					ADD	*6,SP
869	006246			50000*:	PRINTX	*STAER2			
	006246	012746	006616					MOV	*STAER2,-(SP)
	006252	012746	000001					MOV	*1,-(SP)
	006256	010600						MOV	SP,R0
	006260	104415						TRAP	C:PNTX
	006262	062706	000004					ADD	*4,SP
870	006266			PRINTX	*STAER3,CMDPKT,@TSDB(R5),MSGPKT+MS.RFC,TSSREG,L'TCC				
	006266	013746	003450					MOV	CTCC,-(SP)
	006272	013746	003454					MOV	TSSREG,-(SP)
	006276	013746	002360					MOV	MSGPKT+MS.RFC,-(SP)
	006302	017546	002514					MOV	@TSDB(R5),-(SP)
	006306	013746	002330					MOV	CMDPKT,-(SP)
	006312	012746	006675					MOV	*STAER3,-(SP)
	006316	012746	000006					MOV	*6,-(SP)
	006322	010600						MOV	SP,R0
	006324	104415						TRAP	C:PNTX
	006326	062706	000016					ADD	*16,SP
871	006332			PRINTX	*STAER4,CMDPKT+2,CMDPKT+4,CMDPKT+6				
	006332	013746	002336					MOV	CMDPKT+6,-(SP)
	006336	013746	002334					MOV	CMDPKT+4,-(SP)
	006342	013746	002332					MOV	CMDPKT+2,-(SP)
	006346	012746	006733					MOV	*STAER4,-(SP)
	006352	012746	000004					MOV	*4,-(SP)
	006356	010600						MOV	SP,R0
	006360	104415						TRAP	C:PNTX
	006362	062706	000012					ADD	*12,SP
872	006366			PRINTX	*STAER5,MSGPKT+MS.XS0,MSGPKT+MS.XS1,MSGPKT+MS.XS2,MSGPKT+MS.XS3,MSGPKT+MS.XS				
	006366	013746	002372					MOV	MSGPKT+MS.XS4,-(SP)
	006372	013746	002370					MOV	MSGPKT+MS.XS3,-(SP)
	006376	013746	002366					MOV	MSGPKT+MS.XS2,-(SP)
	006402	013746	002364					MOV	MSGPKT+MS.XS1,-(SP)
	006406	013746	002362					MOV	MSGPKT+MS.XS0,-(SP)

GLOBAL ERROR REPORT SECTION

```

006412 012746 006753
006416 012746 000006
006422 010600
006424 104415
006426 062706 000016
873 006432          EXIT   MSG
006432 000167
006434 000432
874
875
876 006436      045      101      130  STAER1: .NLIST  BEX
877          .ASCIZ  /#AXXX CMD FAILED - UNIT #D1#S3#APASS:#D5#S3#ARECORD:#D5#N/
878 006530      045      101      120  STAER7: .ASCIZ  /#APREVIOUS CMD WAS XXX /
879 006562      045      123      061  STAER6: .ASCIZ  /#S11#A* RECORD READ:#D5#A */
880 006616      045      116      045  STAER2: .ASCIZ  /#N#ACHDPKT#S2#ATSBA#S4#ARFC#S5#ATSSR#S3#ATCC#N/
881 006675      045      117      066  STAER3: .ASCIZ  /#06#S2#06#S2#06#S2#06#S2#D1#N/
882 006733      045      117      066  STAER4: .ASCII  /#06#N/
883 006740      045      117      066          .ASCII  /#06#N/
884 006745      045      117      066          .ASCIZ  /#06#N/
885 006753      045      101      130  STAER5: .ASCII  /#AXST0#S4#AXST1#S4#AXST2#S4#AXST3#S4#AXST4#N/
886 007027      045      117      066          .ASCIZ  /#06#S2#06#S2#06#S2#06#S2#06#N/
887          .LIST  BEX
888          .EVEN
889 007066 000000      RECRED: .WORD  0          ;RECORD READ FROM TAPE
890
891 007070
007070          ENDMMSG
007070 104423      L10003:
892
893          .SBTTL  GLOBAL SUBROUTINES SECTION
894
895          ;++
896          ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
897          ; THAT ARE USED IN MORE THAN ONE TEST.
898          ;--
899
900
901
902
903          ;+
904          ;ROUTINE TO DO A SOFT INITIALIZE OF THE CONTROLLER
905          ;BY WRITING INTO THE TSSR REGISTER. AFTER THE INIT.
906          ;THE TSSR REGISTER IS TESTED FOR ERRORS. ANY ERRORS
907          ;DETECTED SHOULD BE TREATED AS DEVICE FATAL ERRORS.
908          ;
909          ;INPUTS:
910          ;
911          ;      R5      CURRENT UNIT NUMBER
912          ;
913          ;
914          ;OUTPUTS:
915          ;
916          ;      R0      CONTENTS OF TSSR, IF ERROR
917          ;      CARRY   SET IF INIT WAS OKAY
918          ;              CLEAR IF FATAL ERROR
919          ;
920          ;CALLING SEQUENCE:

```

GLOBAL SUBROUTINES SECTION

```

921      ;      JSR      PC,FIRSTU
922      ;      JSR      PC,SOFINIT
923      ;      BCS      CONTINUE
924      ;      ERRDF      ;REPORT FATAL ERROR
925      ;
926      ;
927
928 007072      SOFINIT.:
929
930 007072 012775 000000 002524      MOV      #0,@TSSR(R5)      ; (SAVREG) SAVE THE REGISTERS
931 007100 004737 012700      JSR      PC,WSSR      ; DO THE INIT.
932 007104 012703 000550      MOV      #360.,R3      ;WAIT FOR UNIT TO BE READY
933 007110 004737 007204      2$:      JSR      PC,WAITF      ; WAIT FOR SSR
934 007114 103416      BCS      3$
935 007116      DELAY      250
          007116 012727 000250      MOV      #250,(PC)+
          007122 000000      .WORD      0
          007124 013727 002116      MOV      L$DLY,(PC)+
          007130 000000      .WORD      0
          007132 005367 177772      DEC      -6(PC)
          007136 001375      BNE      .-4
          007140 005367 177756      DEC      -22(PC)
          007144 001367      BNE      .-20
936 007146 005303      DEC      R3
937 007150 001357      RNE      2$
938 007152 017500 002524      3$:      MOV      @TSSR(R5),R0      ;GET THE TSSR REGISTER
939 007156 010004      MOV      R0,R4      ;TSSR CONTENTS
940 007160 042704 176277      BIC      #C<TS.A17!IS.A16!TS.OFL>,R4
941 007164 052704 002200      BIS      #TS.SSR!TS.NBA,R4      ;R4 HAS EXPECTED CONTENTS
942 007170 020400      CMP      R4,R0      ;ONLY EXPECTED BITS SET ?
943 007172 001402      BEQ      5$      ;BRANCH IF OKAY
944 007174 000241      CLC      ;CLEAR THE CARRY FOR ERROR
945 007176 000401      BR      10$      ;GO TO EXIT
946 007200 000261      5$:      SEC      ;SET THE CARRY BIT
947 007202 000207      10$:     RTS      PC      ;RETURN TO CALLER
948
949
950      ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
951
952      ; INPUTS:
953
954      ;      R5      CURRENT UNIT NUMBER
955
956
957      ; OUTPUTS:
958
959      ;      R0      CONTENTS OF LAST TSSR READ
960      ;      CARRY   SET - READY BIT SET
961      ;      CLR     CLR - TIMEOUT WAITING FOR READY
962
963 007204      WAITF.: BREAK      ; DO A SUPVSR BREAK FIRST.
          007204 104422      TRAP      C$BRK
964 007206 012746 005670      MOV      #3000.,-(SP)      ; 300 MSEC TIMER.
965 007212 017500 002524      2$:      MOV      @TSSR(R5),R0      ;READ THE TSSR REGISTER
966 007216 105700      TSTB     R0      ;TEST FOR READY BIT SET
967 007220 100420      BMI      3$      ; EXIT ON STOP FLAG.
968 007222      DELAY      25      ; WAIT

```

GLOBAL SUBROUTINES SECTION

```

007222 012727 000025
007226 000000
007230 013727 002116
007234 000000
007236 005367 177772
007242 001375
007244 005367 177756
007250 001367
969 007252 005316
970 007254 001356
971 007256 000241
972 007260 000401
973 007262 000261
974 007264 005326
975 007266 000207
976
977
978
979
980
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 007270
1008 007270 010475 002514
1009 007274 004737 007204
1010 007300 103401
1011 007302 000421
1012 007304 005724
1013 007306 011402
1014 007310 011203
1015 007312 032763 000200 000012
1016 007320 001402
1017 007322 005237 002322

MOV #25,(PC)+
.WORD 0
MOV L$DL7,(PC)+
.WORD 0
DEC -6(PC)
BNE .-4
DEC -22(PC)
BNE .-20

DEC (SP) ;REDUCE DELAY COUNT
BNE 2$ ;RETRY UNTIL TIMER EXPIRES
CLC ; C = 0, CONTROLLER STILL RUNNING...
BR 4$ ;...OR HUNG-UP AFTER 300 MSEC.
3$: SEC ; C = 1, CONTROLLER IS STOPPED.
4$: DEC (SP)+ ;RESTORE STACK WITHOUT CHANGING CARRY BIT
RTS PC

;+
;ROUTINE TO ISSUE A WRITE CHARACTERISTICS COMMAND AND CHECK FEATURES
;INPUT:
; R4 ADDRESS OF COMMAND PACKET
; R5 CURRENT UNIT NUMBER
; REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
;OUTPUT:
; R0 TSSR CONTENTS
; CARRY SET - WRITE CHARACTERISTICS COMMAND OK
; CLR - WRITE CHARACTERISTICS FAILED
;IMPLICIT OUTPUT:
; SOFTWARE SWITCHES SET AS FOLLOWS:
; EXTFEA = EXTENDED FEATURES PRESENT
; BENBSW = BUFFER ENABLE SWITCH ON OR OFF
;SIDE EFFECTS:
;-

WRTCHK:
10$: MOV R4,@(SDB(R5)) ;SEND OUT COMMAND
JSR PC,WAIF ;WAIT FOR SSR
BCS 40$ ;BR, IF SSR IS SET AND OK
BR 60$ ;BR IF TROUBLE CARRY = CLEAR
40$: TST (R4)+ ;STEP IT
MOV (R4),R2 ;POINT TO WRT CHARA DATA PACKET
MOV (R2),R3 ;GET ADDRESS OF MESSAGE BUFFER
BIT #X2.EFE,MS.XS2(R3) ;EXTENDED FEATURES BIT SET?
BEQ 45$ ;BR IF NO
INC EXTFEA ;SET EXTENDED FEATURES SW SWITCH

```

GLOBAL SUBROUTINES SECTION

```

1018 007326          45$:
1019 007326 032763 000100 000012      BIT      4X2.BFE,MS.XS2(R3)      ;BUFFER ENABLE SWITCH SET
1020 007334 001402          BEQ      50$                          ;BR, IF SWITCH NOT SET
1021 007336 005237 002324          INC      BENBSW                    ;SET SOFTWARE SWITCH FOR ENABLED
1022 007342          50$:
1023 007342 000261          55$:      SFC                          ;SET CARRY NO TROUBLE
1024 007344 000401          BR      70$                          ;EXIT
1025 007346 000241          60$:      CLC                          ;CARRY CLEAR - ERROR
1026 007350 017500 002524          70$:      MOV      @TSSR(R5),R0        ;RETURN TSSR CONTENTS
1027 007354 000207          RTS      PC                          ;RETURN
1028
1029          ;*
1030          ;
1031          ;ROUTINE TO CHECK WRITE LOCK CONDITION
1032          ;
1033          ;INPUT:
1034          ;
1035          ;      R4      ADDRESS OF COMMAND PACKET
1036          ;      R5      CURRENT UNIT NUMBER
1037          ;
1038          ;-
1039 007356          WLKCHK:
1040 007356 010475 002514          10$:      MOV      R4,@TSDB(R5)        ;SEND OUT COMMAND
1041 007362 004737 007204          JSR      PC,WAITF                ;WAIT FOR SSR
1042 007366 103401          BCS      40$                      ;BR, IF SSR IS SET AND OK
1043 007370 000420          BR      60$                      ;BR IF TROUBLE CARRY = CLEAR
1044 007372 005724          40$:      TST      (R4)+            ;STEP IT
1045 007374 011402          MOV      (R4),R2                ;POINT TO WRT CHARA DATA PACKET
1046 007376 011203          MOV      (R2),R3                ;GET ADDRESS OF MESSAGE BUFFER
1047 007400 032763 000004 000006      BIT      @X0.WLK,MS.XS0(R3)      ;IS UNIT WRITE LOCKED?
1048 007406 001407          BEQ      55$                      ;NO,PROCEED WITH TESTING
1049 007410          ERRHRD 1,UNIWLK            ;TAPE IS WRITE LOCKED
1050          007410 104456          TRAP    C$ERRHRD
1051          007412 000001          .WORD  1
1052          007414 005653          .WORD  UNIWLK
1053          007416 000000          .WORD  0
1054 007420 004737 017200          JSR      PC,DROPU                ;DROP IT
1055 007424 000402          BR      60$                      ;EXIT WITH CARRY=0
1056 007426 000261          55$:      SEC                          ;SET CARRY NO TROUBLE
1057 007430 000401          BR      70$                      ;EXIT
1058 007432 000241          60$:      CLC                          ;CARRY CLEAR = ERROR
1059 007434 000207          70$:      RTS      PC                          ;RETURN
1060
1061          ;*
1062          ;
1063          ;ROUTINE TO ISSUE A WRITE CHARACTERISTICS COMMAND
1064          ;
1065          ;INPUT:
1066          ;
1067          ;      R4      ADDRESS OF COMMAND PACKET
1068          ;      R5      CURRENT UNIT NUMBER
1069          ;      REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
1070          ;
1071          ;OUTPUT:
1072          ;

```

GLOBAL SUBROUTINES SECTION

```

1071      ;      RO      TSSR CONTENTS
1072      ;      CARRY  SET - WRITE CHARACTERISTICS COMMAND OK
1073      ;
1074      ;
1075      ;IMPLICIT OUTPUT:
1076      ;
1077      ;
1078      ;
1079      ;SIDE EFFECTS:
1080      ;
1081      ;
1082      ;-
1083
1084 007436      WRTCHR:
1085 007436 010475 002514 10$:  MOV      R4,@TSSR(R5)      ;SEND OUT COMMAND
1086 007442 004737 007204      JSR      PC,WAITF          ;WAIT FOR SSR
1087 007446 103401          BCS      50$              ;BR, IF SSR IS SET AND OK
1088 007450 000402          BR       60$              ;BR IF TROUBLE CARRY = CLEAR
1089 007452      50$:
1090 007452 000261          SEC      ;SET CARRY NO TROUBLE
1091 007454 000401          BR       70$              ;EXIT
1092 007456 000241      60$:  CLC      ;CARRY CLEAR = ERROR
1093 007460 017500 002524 70$:  MOV      @TSSR(R5),RO      ;RETURN TSSR CONTENTS
1094 007464 000207          RTS      PC              ;RETURN
1095
1096
1097      ;+
1098      ;
1099      ;ROUTINE TO DO SET UP OF RUNNING CONDITIONS
1100      ;
1101      ;INPUTS:
1102      ;
1103      ;      R5      CURRENT UNIT NUMBER
1104      ;
1105      ;
1106      ;OUTPUTS:
1107      ;
1108      ;
1109      ;CALLING SEQUENCE:
1110      ;      JSR      PC,FIRSTU
1111      ;      JSR      PC,SOFINIT
1112      ;      BCS      CONTINUE
1113      ;
1114      ;      ERRDF          ;REPORT FATAL ERROR
1115      ;      JSR      PC,MDSET
1116      ;
1117      ;-
1118 007466      MDSET:  BREAK          ; DO A SUPVSR BREAK FIRST.
1119 007470 104422          ; TRAP          C$BRK
1120 007474 004737 007766      JSR      PC,SETDEF          ;RESTORE DEFAULT
1121 007500 004737 007356      JSR      PC,WLKCHK          ;CHECK WRITE LOCK
1122 007502 103416          BCS      1$              ;C=1 IS O.K.
1123      007502 012727 000001      DELAY  1              ;WAIT
1124      007506 000000          MOV      @1,(PC)+
1125      007510 013727 002116          .WORD  0
1126      007514 000000          MOV      @1,$DLY,(PC)+
1127          .WORD  0

```

GLOBAL SUBROUTINES SECTION

```

007516 005367 177772          DEC      -6(PC)
007522 001375                BNE      .-4
007524 005367 177756          DEC      -22(PC)
007530 001367                BNF      .-20
1123 007532          BREAK          ;BREAK TO SUPER
007532 104422                TRAP     C$BRK
1124 007534          DOCLN          ;DO CLEAN AND ABORT
007534 104444                TRAP     C$DCLN
1125 007536 005737 002312      1$:   TST      TS1MD          ;RUN IN DEFAULT MODE?
1126 007542 001064                BNE      10$          ;YES,RETURN
1127 007544 004737 007766      JSR      PC,SETDEF      ;RESTORE DEFAULT
1128 007550 004737 007270      JSR      PC,WRTCHK      ;GO DO SWITCH CHECK
1129 007554 005737 002320      TST      HSSW          ;DO WE RUN AT 100IPS?
1130 007560 001415                BEQ      3$          ;NO
1131 007562 052737 000040 003532  BIS      %EF,HSS,TSUNT      ;YES,SET THE BIT
1132 007570 005737 002322      TST      EXTFEA        ;ARE WE SET?
1133 007574 001002                BNE      2$          ;YES
1134 007576 004737 007724      JSR      PC,INVRT       ;INVERT THE SWITCH
1135 007602 004737 007766      2$:   JSR      PC,SETDEF      ;NOW SET THE MODES
1136 007606 004737 007436      JSR      PC,WRTCHR      ;DO IT
1137 007612 000443                BR       11$
1138 007614 005737 002316      3$:   TST      WTBUF          ;RUN WITH WRITE BUFFERING?
1139 007620 001415                BEQ      5$          ;NO
1140 007622 052737 000030 003532  BIS      %EF,RWB,TSUNT      ;YES SET THE BITS
1141 007630 005737 002322      TST      EXTFEA        ;ARE WE SET?
1142 007634 001002                BNE      4$          ;YES
1143 007636 004737 007724      JSR      PC,INVRT       ;INVERT THE SWITCH
1144 007642 004737 007766      4$:   JSR      PC,SETDEF      ;NOW SET THE MODES
1145 007646 004737 007436      JSR      PC,WRTCHR      ;DO IT
1146 007652 000423                BR       11$
1147 007654 005737 002314      5$:   TST      RDBUF          ;RUN WITH READ BUFFERING?
1148 007660 001415                BEQ      10$         ;NO
1149 007662 052737 000020 003532  BIS      %EF,RBO,TSUNT      ;YES SET THE BITS
1150 007670 005737 002322      TST      EXTFEA        ;ARE WE SET?
1151 007674 001002                BNE      6$          ;YES
1152 007676 004737 007724      JSR      PC,INVRT       ;INVERT THE SWITCH
1153 007702 004737 007766      6$:   JSR      PC,SETDEF      ;NOW SET THE MODES
1154 007706 004737 007436      JSR      PC,WRTCHR      ;DO IT
1155 007712 000403                BR       11$
1156
1157 007714 013737 003532 002504 10$:   MOV      TSUNT,SCHBK+10      ;AND UNIT #
1158
1159 007722 000207          11$:   RTS      PC          ;RETURN
1160
1161
1162
1163          ; SUBROUTINE TO INVERT SENSE OF EXT'D FEATURES SWITCH
1164
1165          ;INPUTS:
1166
1167
1168
1169          ;OUTPUTS:
1170
1171
1172
1173 007724          INVRT::

```

GLOBAL SUBROUTINES SECTION

```

1174 007724 012737 14000E 002330      MOV      #WSM,CMDPKT+CP.CMD      ;WRT SUB-SYS MEM
1175 007732 012737 002506 002332      MOV      #WSMBK,CMDPKT+CP.ADL    ;MSG BUF ADDR
1176 007740 012737 000006 002336      MOV      #6,CMDPKT+CP.CNT       ;BYTE COUNT
1177 007746 012737 100010 002506      MOV      #100010,WSMBK         ;INVERT THE SWITCH
1178 007754 012704 002330      MOV      #CMDPKT,R4            ;
1179 007760 004737 007436      JSR      PC,WRTCHR              ;DO IT
1180 007764 000207      RTS      PC                      ;RETURN
1181
1182
1183      ; SUBROUTINE TO SETUP DEFAULT SET CHAR CMD
1184      ;
1185      ; INPUTS:
1186      ;
1187      ;
1188      ;
1189      ; OUTPUTS:
1190      ;
1191      ; R4 ADDRESS OF COMMAND PACKET
1192
1193 007766      SETDEF::
1194 007766 012701 140004      MOV      #SCH,R1                ;WRITE CHAR CMD
1195 007772 010137 002330      MOV      R1,CMDPKT+CP.CMD      ;SET UP COMMAND
1196 007776 012737 002474 002332      MOV      #SCHBK,CMDPKT+CP.ADL  ;SET UP ADR LO TO POINT TO MSG BUF(MSGPK0)
1197 010004 012737 000012 002336      MOV      #SCHCNT,CMDPKT+CP.CNT ;SET BUFFER EXTENT
1198 010012 012737 000040 002502      MOV      #DFTSCH,SCHBK+6       ;STORE CHARACTERISTIC CODE IN SCH BLOCK.
1199 010020 013737 003532 002504      MOV      TSUNT,SCHBK+10        ;UNIT #
1200 010026 012704 002330      MOV      #CMDPKT,R4            ;ADDRESS OF CMD PACKET
1201 010032 000207      RTS      PC                      ;RETURN
1202
1203
1204      ; MODULES TO HANDLE TS05 INTERRUPTS.
1205
1206
1207 010034      BGNSRV  TS5IN0
1208 010034 005237 003472      TS5IN0::
1209 010040      INC      INTFLG                ;SET INTERRUPT OCCURRED FLAG.
1210 010040      ENDSRV
1211 010040 000002      L10004:
1212                                     RTI
1213
1214 010042      BGNSRV  TS5IN1
1215 010042 005237 003474      TS5IN1::
1216 010046      INC      INTFLG+2              ;SET INTERRUPT OCCURRED FLAG.
1217 010046      ENDSRV
1218 010046 000002      L10005:
1219                                     RTI
1220
1221 010050      BGNSRV  TS5IN2
1222 010050 005237 003476      TS5IN2::
1223 010054      INC      INTFLG+4              ;SET INTERRUPT OCCURRED FLAG.
1224 010054      ENDSRV
1225 010054 000002      L10006:
1226                                     RTI
1227
1228 010056      BGNSRV  TS5IN3
1229 010056 005237 003500      TS5IN3::
1230 010056      INC      INTFLG+6              ;SET INTERRUPT OCCURRED FLAG.

```

GLOBAL SUBROUTINES SECTION

```

1221 010062          ENDSRV
      010062          L10007:
      010062 000002          RTI
1222
1223      :          SUBROUTINE TO RETRIEVE RECORD COUNT READ FROM TAPE FOR ERROR
1224      :          PRINTS.
1225      :          INPUTS:
1226      :          OUTPUTS: R3 = RECORD COUNT READ
1227      :          REGISTERS: R2, R3, R4
1228      :          CALLS:
1229
1230 010064 032737 000400 003420 RECTAP:;BIT      @MOD.CO,CMDWRD          ;READ REV FETCH
1231 010072 001430          BEQ      50001$
1232 010074 013702 002360          MOV      MSGPKT+MS.RFC,R2          ;FIND LAST READ AD.
1233 010100 063702 003410          ADD      DATARD,R2
1234 010104 032702 000001          BIT      @BIT00,R2          ;ODD AD., REASSEMBLE
1235 010110 001417          BEQ      50002$
1236 010112 005202          INC      R2          ;REC COUNT STARTING
1237 010114 111203          MOV8    (R2),R3          ;WITH UPPER BYTE FETCH
1238 010116 142703 177400          BICB    @177400,R3
1239 010122 000303          SWAB   R3
1240 010124 005302          DEC      R2          ;LET R2 := R2 - #1          ;LOWER BYTE AD.
1241 010126 105737 003520          TSTB   SWBFLG          ;IFB SWBFLG NE #0 THEN
1242 010132 001401          BEQ      50003$
1243 010134 005302          DEC      R2          ;LET R2 := R2 - #1          ;LOWER BYTE AD. ON SWAP
1244
1245 010136          50003$:
1246 010136 111204          MOV8    (R2),R4          ;FETCH LOWER BYTE
1247 010140 142704 177400          BICB    @177400,R4
1248 010144 050403          BIS      R4,R3
1249 010146 000401          BR      50004$
1250 010150          50002$:
1251 010150 011203          MOV      (R2),R3          ;LET R3 := (R2)          ;EVEN AD. FETCH
1252 010152          50004$:
1253 010152 000402          BR      50005$
1254 010154          50001$:
1255 010154 017703 173230          MOV      @DATARD,R3          ;LET R3 := @DATARD          ;READ FWD FETCH
1256
1257 010160          50005$:
1258 010160 000207          RTS     PC
1259
1260      :          SUBROUTINE TO STORE A SET CHARACTERISTIC COMMAND AS
1261      :          THE FIRST ENTRY IN THE SEQUENCE TABLE.
1262      :          INPUTS:
1263      :          OUTPUTS:
1264      :          REGISTERS:
1265      :          CALLS:
1266
1267 010162          SETCH:;
1268 010162 012701 003540          MOV      @CMDSEQ,R1          ;INIT CMD SEQUENCE TABLE POINTER.
1269 010166 012721 140004          MOV      @SCH,(R1)+          ;THIS CODE SETS UP A SET CHARACTERISTIC
1270 010172 012721 000040          MOV      @DFTSCH,(R1)+          ;COMMAND AS THE FIRST COMMAND IN THE
1271 010176 012721 000001          MOV      @1.(R1)+          ;SEQUENCE TABLE.
1272 010202 005721          TST     (R1)+          ;SKIP PATTERN LOCATION.
1273 010204 000207          RTS     PC
1274
1275      :          SUBROUTINE TO STORE A REWIND COMMAND IN THE SEQUENCE TABLE

```

GLOBAL SUBROUTINES SECTION

```

1276 ; INPUTS;
1277 ; OUTPUTS;
1278 ; REGISTERS;
1279 ; CALLS;
1280
1281 010206 012721 102010 SETRW:: MOV #RWD,(R1)+ ;CMD = REWIND.
1282 010212 012721 000001 MOV #1,(R1)+ ;BRF.
1283 010216 012721 000001 MOV #1,(R1)+ ;# OF OPERATIONS.
1284 010222 005721 TST (R1)+ ;SKIP PATTERN.
1285 010224 000207 RTS PC ;RETURN
1286
1287 ; SUBROUTINE TO EXECUTE ALL COMMANDS IN THE SEQUENCE TABLE ON ALL
1288 ; DEVICES.
1289 ; INPUTS:
1290 ; OUTPUTS: R2 = TERMINATION INDICATOR (0=END OF TABLE,1=EOT)
1291 ; REGISTERS:
1292 ; CALLS: CMDAC,SETUP,EXSUB,CKHAE,NEXTU,FIRSTU,VFYDAT.
1293
1294 010226 012701 003540 EXALL:: MOV #CMDSEQ,R1 ;INIT SEQUENCE TABLE POINTER.
1295 010232 50006$:
1296 010232 021127 177777 CMP (R1),#END ;WHILE THERE ARE CMDS IN THE SEQUENCE TABLE.
1297 010236 001530 BEQ 50007$
1298 010240 004737 011172 JSR PC,SETUP ;GO SETUP THE COMMAND BLOCK.
1299 010244 50010$: BREAK ; DO A SUPVSR BREAK FIRST.
1300 010246 023737 003412 003414 CMP NCNT,NCNT1 ;WHILE THERE ARE RECORDS REMAINING: TRAP C$BRK
1301 010254 002116 BGE 50011$
1302 010256 004737 011064 JSR PC,CMDAC ;STORE CMD ASCII IN ERROR MESSAGE.
1303 010262 105737 003515 TSTB RANDOM ;IF IN RANDOM MODE:
1304 010266 001435 BEQ 50012$
1305 010270 023727 003420 104005 CMP CMDWRD,#WRT ;IF CMD IS A WRITE THEN:
1306 010276 001031 BNE 50013$
1307 010300 105737 003516 TSTB VFYFLG ;IF DATA IS NOT TO BE VERIFIED THEN:
1308 010304 001026 BNE 50014$
1309 010306 063737 003434 003432 ADD RANB,RANB ;LET RANB := RANB + RANS ;GENERATE
1310 010314 063737 003432 003434 ADD RANS,RANB ;LET RANS := RANS + RANB ;RANDOM
1311 010322 013737 003434 003416 MOV RANS,BRFCNT ;LET BRFCNT := RANS ;LENGTH
1312 010330 043737 003430 003416 BIC LENMSK,BRFCNT ;MASK RANDOM LENGTH.
1313 010336 023727 003416 000022 CMP BRFCNT,#18. ;DO NOT ALLOW BYTE COUNT OF LESS THAN 18
1314 010344 002003 BGE 50015$
1315 010346 012737 000022 003416 MOV #18.,BRFCNT ;CHANGE COUNT OF 0-17 TO 18.
1316
1317 010354 50015$:
1318 010354 013737 003416 002336 MOV BRFCNT,CMDPKT+CP.CNT ;MOVE BRF TO CMD PACKET.
1319
1320 010362 50014$:
1321
1322 010362 50013$:
1323
1324 010362 50012$:
1325 010362 004737 010524 JSR PC,EXSUB ;ISSUE CMD TO ALL,AWAIT INTS,CHECK STATUS.
1326 010366 004737 017500 JSR PC,CKHAE ;CHECK HALT AFTER EACH CMD FLAG.
1327 010372 012702 000001 MOV #1,R2 ;LET R2 := #1 ;SET ALL UNITS AT BOT/EOT.
1328 010376 004737 017102 JSR PC,FIRSTU ;FIND FIRST UNIT.
1329
1330 010402 50016$:
1331 010402 026527 002604 177777 CMP DEVTBL(R5),#END ;WHILE THERE ARE MORE UNITS:

```

GLOBAL SUBROUTINE'S SECTION

```

1332 010410 001426          BEQ    50017$
1333 010412 032737 000400 003420      BIT    @MOD.CO,CMDWRD      ;IF CMD IS REVERSE THEN:
1334 010420 001406          BEQ    50020$
1335 010422 032765 000002 003502      BIT    @XO,BOT,EOTFLG(R5) ;IF NOT AT BOT THEN:
1336 010430 001001          BNE   50021$
1337 010432 005002          CLR    R2                  ;LET R2 := #0 ;CLEAR EOT/BOT FLAG.
1338
1339          50021$:
1340 010434 000411          BR     50022$              ;ELSE IF CMD IS NOT REVERSE:
1341 010436          50020$:
1342 010436 032765 000001 003502      BIT    @XO,EOT,EOTFLG(R5)
1343 010444 001404          BEQ    50023$
1344 010446 032737 000001 003420      BIT    @CMD.CO,CMDWRD
1345 010454 001001          BNE   50024$
1346 010456          50023$:
1347
1348 010456 005002          CLR    R2                  ;IF NOT AT EOT OR NOT A MOTION CMD THEN:
1349
1350          50024$:
1351
1352 010460          50022$:
1353 010460 004737 017150          JSR    PC,NEXTU          ;FIND NEXT UNIT
1354 010464 000746          BR     50016$
1355 010466          50017$:
1356 010466 020227 000001          CMP    R2,#1              ;IF ALL UNIT ARE AT EOT/BOT THEN:
1357 010472 001001          BNE   50025$
1358 010474 000412          BR     EXARTN          ;RETURN WITH R2 = #1.
1359
1360 010476          50025$:
1361 010476 005237 003412          INC    NCNT              ;LET NCNT := NCNT + #1 ;UPDATE RECORD COUNT.
1362 010502 013737 003420 003424      MOV    CMDWRD,PCMDWD     ;SAVE PREVIOUS COMMAND WORD.
1363
1364 010510 000655          BR     50010$
1365 010512          50011$:
1366 010512 004737 016066          JSR    PC,VFYDAT        ;IF LAST CMD WAS A WRITE VERIFY, THEN GO
1367
1368
1369 010516 000645          BR     50006$
1370 010520          50007$:
1371 010520 005002          CLR    R2                  ;LET R2 := #0 ;SET NORMAL RETURN INDICATOR.
1372 010522 000207          EXARTN: RTS PC          ;RETURN.
1373
1374
1375
1376          ; SUBROUTINE TO ISSUE COMMAND TO ALL DEVICES. WAIT FOR
1377          ; ALL INTERRUPTS, AND CHECK ALL STATUS.
1378          ; INPUTS:
1379          ; OUTPUTS:
1380          ; REGISTERS:
1381          ; CALLS:          EXECUTE,GOWAIT,NEXTU,FIRSTU.
1382
1383 010524 004737 017102          EXSUB: JSR    PC,FIRSTU          ;SET UP FOR FIRST UNIT.
1384 010530          50026$:
1385 010530 026527 002604 177777      CMP    DEVTBL(R5),#END    ;WHILE THERE ARE MORE DEVICES:
1386 010536 001465          BEQ    50027$
1387 010540 032737 000400 003420      BIT    @MOD.CO,CMDWRD      ;IF CMD IS REVERSE THEN:
1388 010546 001421          BEQ    50030$

```

GLOBAL SUBROUTINES SECTION

```

1389 010550 032765 000002 003502 BIT @X0.BOT,EOTFLG(R5) ;IF NOT AT BOT
1390 010556 001014 BNE 50031$
1391 010560 032765 000001 003502 BIT @X0.EOT,EOTFLG(R5) ;BUT IF AT EOT
1392 010566 001406 BEQ 50032$
1393 010570 105737 003524 TSTB ALLEOT ;AND ALL OTHERS AT EOT
1394 010574 001402 BEQ 50033$
1395 010576 004737 012054 JSR PC,EXECUTE ;THEN EXECUTE REV CMD
1396 ;IF NOT ALL AT EOT, FREEZE UNIT(S) AT EOT
1397 010602 50033$: BR 50034$ ;IF NOT AT BOT AND
1398 010602 000402 BR 50034$
1399 010604 50032$: JSR PC,EXECUTE ;NOT AT EOT, EXEC REV CMD
1400 010604 004737 012054 JSR PC,EXECUTE
1401
1402 010610 50034$:
1403
1404 010610 50031$: BR 50035$ ;ELSE IF CMD IS NOT REVERSE:
1405 010610 000435 BR 50035$
1406 010612 50030$: CMP CMDLG,#2
1407 010612 023727 003426 000002 BNE 50036$
1408 010620 001011 BIT @X0.BOT,EOTFLG(R5)
1409 010622 032765 000002 003502 BEQ 50036$
1410 010630 001405
1411 ;CLEAR R SPOT COUNTS WHEN WRITING FROM BOT
1412 010632 016537 002616 003512 MOV BTADDR(R5),BTPT ;LET BTPT := BTADDR(R5)
1413 010640 005077 172646 CLR @BTPT ;LET @BTPT := #0
1414
1415 010644 50036$: BIT @X0.EOT,EOTFLG(R5)
1416 010644 032765 000001 003502 BEQ 50037$
1417 010652 001404 BIT @CMD.CO,CMDWRD
1418 010654 032737 000001 003420 BNE 50040$
1419 010662 001003
1420 010664 50037$: JSR PC,EXECUTE ;IF NOT AT EOT OR NOT A MOTION CMD THEN:
1421 ;ISSUE CMD TO TS05
1422 010664 004737 012054 JSR PC,EXECUTE
1423
1424 010670 000405 BR 50041$
1425 010672 50040$: TSTB ALLEOT ;IFB ALLEOT NE #0 THEN
1426 010672 105737 003524 BEQ 50042$
1427 010676 001402 JSR PC,EXECUTE
1428 010700 004737 012054 JSR PC,EXECUTE
1429
1430 010704 50042$:
1431
1432 010704 50041$:
1433
1434 010704 50035$: JSR PC,NEXTU ;FIND NEXT UNIT IN TEST CYCLE.
1435 010704 004737 017150 JSR PC,NEXTU
1436
1437 010710 000707 BR 50026$
1438 010712 50027$: TSTB RPTFLG ;IF REPORT HAS BEEN REQUESTED THEN:
1439 010712 105737 003517 BEQ 50043$
1440 010716 001403 CLR RPTFLG
1441 010720 105037 003517 DORPT ;CLR THE FLAG.
1442 010724 104424 ;PRINT THE PERFORMANCE REPORT. TRAP C$DRPT
1443 010726 50043$: JSR PC,FIRSTU ;SET UP FOR FIRST UNIT.
1444 010726 004737 017102 JSR PC,FIRSTU

```

GLOBAL SUBROUTINES SECTION

```

1445 010732          50044$:
1446 010732 026527 002604 177777      CMP     DFVTL(R5),0END      ;WHILE THERE ARE MORE DEVICES:
1447 010740 001450          BEQ     50045$
1448 010742 032737 000400 003420      BIT     *MOD.CO,CMDWRD      ;IF CMD IS REVERSE THEN:
1449 010750 001421          BEQ     50046$
1450 010752 032765 000002 003502      BIT     *XO.BOT,EOTFLG(R5) ;IF NOT AT BOT
1451 010760 001014          BNE     50047$
1452 010762 032765 000001 003502      BIT     *XO.EOT,EOTFLG(R5) ;BUT IF AT EOT
1453 010770 001406          BEQ     50050$
1454 010772 105737 003524          TSTB   ALI EOT              ;AND ALL OTHERS AT EOT
1455 010776 001402          BEQ     50051$
1456 011000 004737 012364          JSR    PC,GOWAIT           ;THEN WAIT FOR CMD END
1457                                     ;IF NOT ALL AT EOT, DO NOT WAIT
1458 011004          50051$:
1459                                     ;NOT AT BOT, AND NOT AT EOT
1460 011004 000402          BR     50052$
1461 011006          50050$:
1462 011006 004737 012364          JSR    PC,GOWAIT           ;WAIT FOR INT,CHECK STAT
1463
1464 011012          50052$:
1465
1466
1467 011012          50047$:
1468 011012 000420          BR     50053$              ;ELSE IF CMD IS FORWARD:
1469 011014          50046$:
1470 011014 032765 000001 003502      BIT     *XO.EOT,EOTFLG(R5)
1471 011022 001404          BEQ     50054$
1472 011024 032737 000001 003420      BIT     *CMD.CO,CMDWRD
1473 011032 001003          BNE     50055$
1474 011034          50054$:
1475                                     ;IF NOT AT EOT OR NOT A MOTION CMD THEN:
1476 011034 004737 012364          JSR    PC,GOWAIT           ;WAIT FOR INT,CHECK STATUS.
1477
1478 011040 000405          BR     50056$
1479 011042          50055$:
1480 011042 105737 003524          TSTB   ALLEOT              ;IFB ALLEUT NE #0 THEN
1481 011046 001402          BEQ     50057$
1482 011050 004737 012364          JSR    PC,GOWAIT
1483
1484 011054          50057$:
1485
1486 011054          50056$:
1487
1488 011054          50053$:
1489 011054 004737 017150          JSR    PC,NEXTU           ;FIND NEXT UNIT IN TEST CYCLE.
1490
1491 011060 000724          BR     50044$
1492 011062          50045$:
1493 011062 000207          RTS     PC                ;RETURN.
1494
1495 ; THIS SUBROUTINE STORES THE ASCII FOR THE CURRENT COMMAND AND PREVIOUS
1496 ; COMMAND IN THE STANDARD ERROR MESSAGE. ON ENTRY LOCATION CMDWRD
1497 ; CONTAINS CURRENT CMD AND LOCATION PCMDWD CONTAINS PREVIOUS CMD.
1498 ; INPUTS:
1499 ; OUTPUTS:
1500 ; REGISTERS: R3, R4.
1501 ; CALLS: GCMDB

```

FA

GLOBAL SUBROUTINES SECTION

```

1502
1503 011064 013704 003420 CMDAC:: MOV CMDWRD,R4;LET R4 := CMDWRD ;R4 = CMD BINARY.
1504 011070 004737 011136 JSR PC,GCMDA ;GET CMD ASCII.
1505 011074 112337 006440 MOV (R3)+,STAER1+2 ;MOVE CMD ASCII
1506 011100 112337 006441 MOV (R3)+,STAER1+3 ;
1507 011104 111337 006442 MOV (R3),STAER1+4 ;INTO MSG.
1508 011110 013704 003424 MOV PCMDWD,R4 ;R4 = PREVIOUS CMD BINARY.
1509 011114 004737 011136 JSR PC,GCMDA ;GET CMD ASCII.
1510 011120 112337 006554 MOV (R3)+,STAER7+24 ;MOVE CMD ASCII
1511 011124 112337 006555 MOV (R3)+,STAER7+25 ;
1512 011130 111337 006556 MOV (R3),STAER7+26 ;INTO MSG.
1513 011134 000207 RTS PC ;RETURN. GO EXECUTE NEXT FUNCTION.
1514
1515
1516 ; SUBROUTINE TO FIND THE ASCII EQUIVILENT OF THE COMMAND IN R4.
1517 ; ADDRESS OF ASCII 1ST WORD IS RETURNED IN R3.
1518 ; INPUTS: R4 = PRESENT COMMAND WORD.
1519 ; OUTPUTS: R3 = ADDRESS OF PRESENT COMMAND ASCII.
1520 ; REGISTERS:
1521 ; CALLS:
1522
1523 011136 005003 GCMDA:: CLR R3;LET R3 := #0 ;INIT CMD TBL POINTER.
1524 011140 50060$ CMP CMDTBL(R3),R4 ;UNTIL CURRENT CMD IS FOUND:
1525 011140 026304 003752 BEQ 50061$
1526 011144 001403 ADD #2,R3 ;LET R3 := R3 + #2 ;SEARCH CMD TABLE.
1527 011146 062703 000002 BR 50060$
1528 011152 000772
1529 011154 50061$:
1530 011154 010304 MOV R3,R4 ;LET R4 := R3
1531 011156 006203 ASR R3 ;POINT TO ASCII FOR THAT COMMAND
1532 011160 000240 NOP
1533 011162 060403 ADD R4,R3
1534 011164 062703 004040 ADD #CMDASC,R3
1535 011170 000207 RTS PC ;RETURN.
1536
1537 ; THIS SUBROUTINE LOADS THE TS05 COMMAND PACKET FROM ONE
1538 ; ENTRY IN THE SEQUENCE TABLE.
1539 ; INPUTS:
1540 ; OUTPUTS:
1541 ; REGISTERS: R2, R3.
1542 ; CALLS: GENPAT.
1543
1544 011172 005037 003426 SETUP:: CLR CMDLG ;CLR CMD LOGGING CODE(DISABLES LOGGING)
1545 011176 012137 002330 MOV (R1)+,CMDPKT ;LOAD THE COMMAND WORD.
1546 011202 011137 002336 MOV (R1),CMDPKT+CP,CNT ;LOAD THE BYTE/RECORD/FILE COUNT.
1547 011206 011137 003416 MOV (R1),BRFCNT ;SAVE BRF FOR THIS COMMAND.
1548 011212 013702 002330 MOV CMDPKT,R2 ;GET CMD.
1549 011216 042702 177740 BIC #NCMD,C,R2 ;CLR ALL BUT CMD BITS.
1550 011222 010203 MOV R2,R3 ;SAVE IT TWICE.
1551 011224 062703 000010 SUB #CMD.C3,R3 ;POSITION COMMAND?
1552 011230 001003 BNE 2$ ;BR IF NOT.
1553 011232 011137 002332 MOV (R1),CMDPKT+2 ;MOVE BPCR IN 2ND PKT WORD FOR POSITION CMD.
1554 011236 000464 BR 3$
1555 011240 023727 002330 100011 2$: CMP CMDPKT,#WTM ;IF CMD IS A WRITE TAPE MARK THEN:
1556 011246 001003 BNE 50062$
1557 011250 012737 000002 003426 MOV #2,CMDLG ;WTM LOGGING CODE IS 2.
1558

```

GLOBAL SUBROUTINES SECTION

```

1559 011256          50062$:
1560 011256 010203      MOV      R2,R3
1561 011260 162703 000001  SUB      @CMD.CO,R3      ;IS IT A READ?
1562 011264 001017      BNE      1$             ;BR IF NOT.
1563 011266 013737 003410 002332  MOV      DATARD,CMDPKT+CP,ADL ;IF SO, LOAD THE BUFFER ADDR.
1564 011274 032737 000400 002330  BIT      @MOD.CO,CMDPKT    ;IF CMD IS A READ REV THEN:
1565 011302 001404      BEQ      50063$
1566 011304 012737 000004 003426  MOV      @4,CMDLG         ;LOGGING CODE IS 4.
1567                                ;ELSE - IF CMD IS A READ FWD:
1568 011312 000403      BR       50064$
1569 011314
1570 011314 012737 000006 003426 50063$: MOV      @6,CMDLG         ;LOGGING CODE IS 6.
1571
1572 011322          50064$:
1573 011322 000432      BR       3$             ;CONTINUE.
1574 011324 010203      1$: MOV      R2,R3         ;IS IT
1575 011326 162703 000004      SUB      @CMD.C2,R3      ;A SET CHARACTERISTICS CMD?
1576 011332 001014      BNE      4$             ;BR IF NOT.
1577 011334 012737 002474 002332  MOV      @SCHBK,CMDPKT+CP,ADL ;SET UP ADR LO FOR SET CHAR.
1578 011342 012737 000012 002336  MOV      @SCHCNT,CMDPKT+CP,CNT ;SET BUFFER EXTENT
1579 011350 011137 002502      MOV      (R1),SCHBK+6     ;STORE CHARACTERISTIC CODE IN SCH BLOCK.
1580 011354 013737 003532 002504  MOV      TSUNT,SCHBK+10   ;UNIT
1581 011362 000412      BR       3$             ;CONTINUE.
1582 011364 010203      4$: MOV      R2,R3         ;IS IT
1583 011366 162703 000006      SUB      @CMD.C1!CMD.C2,R3 ;A DIAGNOSTIC (DIA) CMD?
1584 011372 001006      BNE      3$             ;BR IF NOT.
1585 011374 012737 000020 002336  MOV      @DIACNT,CMDPKT+CP,CNT ;LOAD BUFFER EXTENT.
1586 011402 012737 003406 002332  MOV      @DIABLK,CMDPKT+CP,ADL ;LOAD BUFFER ADR LOW.
1587 011410 005721      3$: TST      (R1)+          ;POINT TO N (NUMBER OF TIMES TO EXECUTE THIS INS
1588 011412 012137 003414      MOV      (R1)+,NCNT1     ;SAVE NUMBER OF OPERATIONS
1589 011416 005037 003412      CLR      NCNT           ;CLEAR OPERATION COUNTER.
1590 011422 012137 003446      MOV      (R1)+,PATTERN   ;SAVE PATTERN CODE FOR CURRENT CMD.
1591 011426 010203      MOV      R2,R3         ;IS IT
1592 011430 162703 000005      SUB      @CMD.CO!CMD.C2,R3 ;A WRITE?
1593 011434 001010      BNE      5$             ;BR IF NOT.
1594 011436 013737 003406 002332  MOV      DATAWT,CMDPKT+CP,ADL ;LOAD WRITE BUFFER LO ORDER.
1595 011444 004737 011556      JSR      PC,GENPAT       ;GO GENERATE THE WRITE PATTERN.
1596 011450 012737 000002 003426  MOV      @2,CMDLG        ;WRITE LOGGING CODE IS 2.
1597 011456 032737 000100 002330  5$: BIT      @VFY.C,CMDPKT  ;IF DATA VERIFICATION IS REQUIRED:
1598 011464 001407      BEQ      50065$
1599 011466 112737 000001 003516  MOVB     @1,VFYFLG       ;SET VERIFY FLAG.
1600 011474 042737 000100 002330  BIC      @VFY.C,CMDPKT   ;CLEAR VERIFY BIT(NOT USED BY HARDWARE).
1601                                ;IF DATA VERIFICATION IS NOT REQUIRED:
1602 011502 000402      BR       50066$
1603 011504
1604 011504 105037 003516      50065$: CLRB     VFYFLG         ;CLR VERIFY FLAG.
1605
1606 011510          50066$:
1607 011510 013737 003420 003424  MOV      CMDWRD,PCMDWD    ;SAVE PREVIOUS CMD WORD.
1608 011516 013737 002330 003420  MOV      CMDPKT,CMDWRD   ;SAVE PRESENT CMD WORD.
1609 011524 105737 003520      TSTB     SWBFLG         ;IF SWAP BYTES IS ENABLED:
1610 011530 001403      BEQ      50067$
1611 011532 052737 010000 002330  BIS      @SWB.C,CMDPKT   ;SET SWAP BIT IN COMMAND.
1612
1613 011540          50067$:
1614 011540 042737 004000 002330  BIC      @BRF.C,CMDPKT   ;CLR BRF BIT (INTERNAL ONLY).
1615 011546 013737 02330 003422  MOV      CMDPKT,CMDSAV   ;SAVE 1ST WORD OF COMMAND PACKET.

```

66

GLOBAL SUBROUTINES SECTION

```

1616 011554 000207          RTS      PC              ;RETURN.
1617
1618                      :      THIS SUBROUTINE SETS UP AND CALLS THE APPROPRIATE SUBROUTINE TO GENERATE
1619                      :      THE DESIRED PATTERN FOR THE WRITE AND WRITE/VERIFY COMMANDS.
1620                      :      INPUTS:
1621                      :      OUTPUTS:
1622                      :      REGISTERS:      R2, R3, R4.
1623                      :      CALLS:      PATR0 - PATR7
1624
1625 011556 013703 003446  GENPAT:;MOV      PATERN,R3      ;SETUP PATTERN ROUTINE POINTER
1626 011562 006303          ASL      R3
1627 011564 013704 003416  MOV      BRFCNT,R4      ;SET LENGTH OF WRITE BFR
1628 011570 005204          INC      R4
1629 011572 042704 000001  BIC      #1,R4          ;ROUNDED UP TO NEXT WORD
1630 011576 162704 000002  SUB      #2,R4          ;WITH FIRST WORD RESERVED
1631 011602 013702 003406  MOV      DATAW,R2      ;FOR RECORD COUNT
1632 011606 062702 000002  ADD      #2,R2
1633 011612 004773 011620  JSR      PC,@PATTBL(R3) ;GO GENERATE THE APPROPRIATE PATTERN.
1634 011616 000207          RTS      PC              ;RETURN TO SETUP SUBROUTINE.
1635
1636                      ;TS05 WRITE PATTERN LOOKUP TABLE. USED TO JSR TO THE
1637                      ;CORRECT DATA PATTERN GENERATING ROUTINE.
1638
1639 011620 011642          PATTBL: PATR0
1640 011622 011700          PATR1
1641 011624 011720          PATR2
1642 011626 011730          PATR3
1643 011630 011754          PATR4
1644 011632 011766          PATR5
1645 011634 012000          PATR6
1646 011636 012020          PATR7
1647 011640 012052          PATR8
1648
1649                      ;INCREMENTING PATTERN. 0 - 377.
1650
1651 011642 012703 000400  PATR0:; MOV      #400,R3;LET R3 := #400
1652 011646 162704 000002  1$:  SUB      #2,R4;LET R4 := R4 - #2 ;DECREMENT WORD COUNT.
1653 011652 100411          BMI      2$          ;BR IF DONE.
1654 011654 010322          MOV      R3,(R2);
1655 011656 062703 001002  ADD      #1002,R3      ;STORE DATA WORD.
1656 011662 020327 001000  CMP      R3,#1000      ;UPDATE PATTERN.
1657 011666 001002          BNE     50070$      ;IF PATTERN HAS WRAPPED AROUND THEN:
1658 011670 012703 000400  MOV      #400,R3
1659                      ;INIT THE PATTERN AGAIN.
1660
1661 011674          50070$:
1662 011674 000764          BR      1$          ;DO IT AGAIN.
1663 011676 000207          2$:  RTS      PC              ;RETURN.
1664
1665                      ;ALL ONE'S PATTERN.
1666
1667 011700 012703 177777  PATR1:; MOV      #-1,R3      ;ALL ONES PATTERN;.
1668 011704 162704 000002  ZROPAT: SUB      #2,R4      ;DECREMENT BYTE COUNT.
1669 011710 100402          BMI      1$          ;DONE?,BR IF YES.
1670 011712 010322          MOV      R3,(R2);
1671 011714 000773          BR      ZROPAT      ;IF NOT LOAD NEXT BYTE WITH PATTERN.
1672                      ;DO IT AGAIN.

```


GLOBAL SUBROUTINES SECTION

```

1730      ;      INPUTS:
1731      ;      OUTPUTS:
1732      ;      REGISTERS:      R2, R3.
1733      ;      CALLS:      DROPU, MOVMSG, FIRSTU, NEXTU, WSSR.
1734
1735 012054 012737 177777 003436 EXECUTE:: MOV      0-1, TIME1      ;INIT TIMEOUT COUNTER.
1736 012062      50071$: ;REPEAT      ;WAIT -
1737 012062 005337 003436      DEC      TIME1      ;UPDATE TIMEOUT COUNTER.
1738 012066 005737 003436      TST      TIME1      ;IF TIMED OUT:
1739 012072 001011      BNE      50072$
1740 012074 004737 012734      JSR      PC, MOVMSG      ;MOVE CURRENT PACKET MSG.
1741 012100      ERRDF      2, NSSRM, STAERM      ;REPORT TS05 NOT READY
      012100 104455
      012102 000002
      012104 004536
      012106 006120
1742 012110 004737 017200      JSR      PC, DROPU      ;DROP THE UNIT.
1743 012114 000522      BR      EXCRTN      ;RETURN.
1744
1745 012116      50072$:
1746 012116 032775 000200 002524      BIT      0TS, SSR, 0TSSR(R5)      ;WAIT UNTIL DEVICE IS READY.
1747 012124 001756      BEQ      50071$
1748 012126 023727 003420 140004      CMP      CMDWRD, 0SCH      ;IF WE ARE DOING A SET CHAR CMD THEN:
1749 012134 001022      BNE      50073$
1750 012136 010537 003452      MOV      R5, R5SAVE      ;SAVE CURRENT DEVICE POINTER.
1751 012142 004737 017102      JSR      PC, FIRSTU      ;FIND FIRST UNIT.
1752 012148
1753 012148 026527 002604 177777      50074$:      CMP      DEVTBL(R5), 0END      ;WHILE DEVTBL(R5) NE 0END DO
1754 012154 001405      BEQ      50075$
1755 012156 004737 012700      JSR      PC, WSSR      ;WAIT FOR UNIT READY OR TIME OUT.
1756 012162 004737 017150      JSR      PC, NEXTU      ;FIND NEXT UNIT.
1757
1758 012166 000767      BR      50074$
1759 012170      50075$:
1760 012170 013705 003452      MOV      R5SAVE, R5      ;RESTORE CURRENT DEVICE POINTER.
1761 012174 016537 002544 002474      MOV      MSGPKA(R5), SCHBK      ;SET UP ADR OF MSG PKT IN SCH BLOCK.
1762
1763 012202      50073$:
1764 012202 016503 002544      MOV      MSGPKA(R5), R3      ;ADR OF THIS UNIT'S MSG PACKET.
1765 012206 005002      CLR      R2      ;CLR COUNTER.
1766 012210      50076$:
1767 012210 020227 000020      CMP      R2, 0MSGCNT      ;WHILE THERE ARE MORE LOCATIONS:
1768 012214 001405      BEQ      50077$
1769 012216 012723 177777      MOV      0-1, (R3),      ;INIT THE MSG PACKET WITH ALL 1'S
1770 012222 062702 000002      ADD      02, R2      ;UPDATE COUNTER.
1771
1772 012226 000770      BR      50076$
1773 012230      50077$:
1774 012230 105737 002212      TSTB     DINT      ;ARE INTERRUPTS DISABLED.
1775 012234 001023      BNE      1$      ;BR IF YES.
1776 012236 126527 003472 000001      CMPB     INTFLG(R5), 01      ;IF MORE THAN ONE INTERRUPT HAS OCCURED:
1777 012244 003412      BLE      50100$
1778 012246 017537 002524 003454      MOV      0TSSR(R5), 1SSREG      ;FREEZE THE CURRENT STATUS REG FOR PRINT
1779 012254      ERRDF      15, TOOMM, STAERM      ;REPORT TOO MANY INTERRUPTS.
      012254 104455
      012256 000017
      012260 004727
      TRAP      C$ERDF
      .WORD     15
      .WORD     TOOMM

```

GLOBAL SUBROUTINES SECTION

```

1780 012262 006120
1780 012264 004737 017200 JSR PC,DROPU ;DROP THE UNIT .WORD STAERM
1781 012270 000434 BR EXCRTN ;RETURN - UNIT HAS BEEN DROPPED.
1782
1783 012272 50100$:
1784 012272 005065 003472 CLR INTFLG(R5) ;CLR INTERRUPT FLAG FOR THIS DEV.
1785 012276 052737 000200 002330 BIS #IE.C,CMDPKT ;SET INT ENABLE BIT.
1786 012304 105737 003471 1$: TSTB ERRREC;IFB ERRREC EQ #0 THEN ;IF NOT RETRYING
1787 012310 001005 BNE 50101$
1788 012312 005265 003376 INC RECCNT(R5) ;LET RECCNT(R5) := RECCNT(R5) + #1
1789 012316 016577 003376 171062 MOV RECCNT(R5),@DATAWT ;THEN UPDATE REC COUNT TO WRITE IT ON TAPE
1790
1791 012324 50101$:
1792 012324 012775 002330 002514 MOV #CMDPKT,@TSDB(R5) ;LOAD TSDB WITH CMDPKT ADDRESS
1793 ;THIS INITIATES COMMAND EXECUTION.
1794 012332 032775 000200 002524 BIT #TS.SSR,@TSSR(R5) ;IF READY DID NOT DROP THEN:
1795 012340 001410 BEQ 50102$
1796 012342 004737 012734 JSR PC,MOVMSG ;MOVE CURRENT MESSAGE PACKET TO COMMON.
1797 012346 ERRDF 3,TOERM,STAERM ;REPORT NO TS05 RESPONSE.
012346 104455 TRAP C$ERDF
012350 000003 .WORD 3
012352 004453 .WORD TOERM
012354 006120 .WORD STAERM
1798 012356 004737 017200 JSR PC,DROPU ;DROP THE UNIT
1799
1800 012362 50102$:
1801 012362 000207 EXCRTN: RTS PC ;RETURN.
1802
1803 ; THIS SUBROUTINE WAITS FOR THE TS05 INERRUPT OR DONE BIT TO SET AND ALLOWS THE
1804 ; OPERATOR TO TRANSFER CONTROL TO THE SUPERVISOR.
1805 ; UPON APPEARANCE OF THE INTERRUPT OR DONE, CHECK TSSR FOR STATUS ERRORS,
1806 ; LOG BYTES AND ERRORS AND PERFORM ERROR RECOVERY IF NECESSARY.
1807 ; INPUTS:
1808 ; OUTPUTS:
1809 ; REGISTERS: R2, R3.
1810 ; CALLS: DROPU, MOVMSG, RECUD, CHKERR, LOG, CLRERR.
1811
1812 012364 012737 177777 003436 GOWAIT:; MOV #-1,TIME1 ;INIT TIME OUT COUNTER.
1813 012372 50103$: ;REPEAT ;REPEAT UNTIL INTERRUPT OCCURES:
1814 012372 BREAK ;GO TO THE SUPER TO ALLOW TTY INPUT.
012372 104422 TRAP C$BRK
1815 012374 023727 003420 102010 CMP CMDWRD,@RWD ;IF COMMAND WAS REWIND THEN:
1816 012402 001014 BNE 50104$
1817 012404 DELAY 10. ;WAIT EXTRA MSEC'S EACH LOOP.
012404 012727 000012 MOV #10,.(PC)+
012410 000000 .WORD 0
012412 013727 002116 MOV L$DLY,(PC)+
012416 000000 .WORD 0
012420 005367 177772 DEC -6(PC)
012424 001375 BNE -4
012426 005367 177756 DEC -22(PC)
012432 001367 BNE -20
1818 012434 50104$:
1819 012434 023727 003420 105010 CMP CMDWRD,#SFF ;IF CMDWRD EQ #SFF OR CMDWRD EQ #SFR THEN
1820 012442 001404 BEQ 50105$
1821 012444 023727 003420 105410 CMP CMDWRD,#SFR
1822 012452 001014 BNE 50106$

```

1/4,

GLOBAL SUBROUTINES SECTION

```

1823 012454          50105$:
1824 012454          DELAY 12.          ;ADD DELAY FOR SPACE TAPE MARK COMMANDS
      012454 012727 000014          MOV      #12.,(PC)+
      012460 000000          .WORD 0
      012462 013727 002116          MOV      L$DLY,(PC)+
      012466 000000          .WORD 0
      012470 005367 177772          DEC      -6(PC)
      012474 001375          BNE      .-4
      012476 005367 177756          DEC      -22(PC)
      012502 001367          BNE      .-20
1825 012504          50106$:
1826 012504 105737 002212          TSTB   DINT          ;IF INTERRUPTS ARE ENABLED.
1827 012510 001003          BNE   50107$
1828 012512 016502 003472          MOV   INTFLG(R5),R2 ;FETCH INTERRUPT OCCURRED FLAG.
1829
1830 012516 000406          BR    50110$
1831 012520          50107$:
1832 012520 012703 000200          MOV   #TS.SSR,R3   ;SET UP A MASK FOR THE DONE BIT.
1833 012524 005103          COM   R3
1834 012526 017502 002524          MOV   @TSSR(R5),R2 ;FETCH DONE BIT.
1835 012532 040302          BIC   R3,R2
1836
1837 012534          50110$:
1838 012534 005337 003436          DEC   TIME1        ;UPDATE TIMEOUT COUNTER.
1839 012540 005702          TST   R2            ;REPEAT UNTIL INTERRUPT OR READY OCCURS.
1840 012542 001003          BNE   50111$
1841 012544 005737 003436          TST   TIME1
1842 012550 001310          BNE   50103$
1843 012552          50111$:
1844 012552 005737 003436          TST   TIME1        ;IF TIME OUT HAS OCCURRED:
1845 012556 001022          BNE   50112$
1846 012560 016577 003376 170620          MOV   RECCNT(R5),@DATAWT
1847 012566 005377 170614          DEC   @DATAWT
1848 012572 004737 012734          JSR   PC,MOVMSG    ;MOVE CURRENT MSG PACKET TO COMMON AREA.
1849 012576          ERRDF 4,NOINTM,STAERM ;REPORT NO INTERRUPT.
      012576 104455          TRAP  C$ERRDF
      012600 000004          .WORD 4
      012602 004670          .WORD NOINTM
      012604 006120          .WORD STAERM
1850 012606 004737 017200          JSR   PC,DROPU     ;DROP THE UNIT.
1851 012612 012703 003472          MOV   #ENDERF,R3  ;LET R3 := #ENDERF
1852 012616 004737 012664          JSR   PC,CLRERR   ;CLEAR ALL ERROR FLAGS
1853
1854 012622 000417          BR    50113$
1855 012624          50112$:
1856 012624 004737 012734          JSR   PC,MOVMSG    ;MOVE CURRENT MSG. PACKET TO COMMON AREA.
1857 012630 004737 013020          JSR   PC,RECUD     ;UPDATE THE RECORD COUNT.
1858 012634 004737 013210          JSR   PC,CHKERR   ;CHECK FOR STATUS ERRORS.
1859 012640 105737 003463          TSTB  WRTYFG      ;IFB WRTYFG EQ #0 THEN
1860 012644 001006          BNE   50114$
1861 012646 004737 015566          JSR   PC,LOG       ;LOG BYTES AND ERRORS.
1862 012652 012703 003472          MOV   #ENDERF,R3  ;LET R3 := #ENDERF
1863 012656 004737 012664          JSR   PC,CLRERR   ;CLEAR ALL ERROR FLAGS
1864
1865 012662          50114$:
1866
1867 012662          50113$:

```

GLOBAL SUBROUTINES SECTION

```

1868 012662 000207          RTS      PC              ;RETURN IF DONE.
1869
1870          ;          SUBROUTINE TO CLEAR FLAGS.
1871          ;          INPUTS:          R3 = LWA TO BE CLEARED + 2.
1872          ;          OUTPUTS:
1873          ;          REGISTERS:      R2
1874          ;          CALLS:
1875
1876 012664 012702 003460    CLRERR:; MOV      #BGNFLG,R2      ;LET R2 := #BGNFLG
1877 012670          50115$:;REPEAT
1878 012670 005022          CLR      (R2)+          ;LET (R2)+ := #0
1879 012672 020203          CMP      R2,R3          ;UNTIL R2 EQ R3
1880 012674 001375          BNE      50115$
1881 012676 000207          RTS      PC
1882
1883
1884          ;          SUBROUTINE TO WAIT UNTIL CURRENT UNIT IS READY OR UNTIL TIME OUT.
1885          ;          INPUTS:
1886          ;          OUTPUTS:
1887          ;          REGISTERS:
1888          ;          CALLS:
1889
1890 012700          WSSR:;
1891 012700 012737 177777 003436    MOV      #-1,TIME1          ;INIT TIMEOUT COUNTER.
1892 012706          50116$:;REPEAT UNTIL DEV READY OR TIMEOUT:
1893 012706          BREAK          ;BREAK TO THE SUPERVISOR.
1894 012710 005337 003436          DEC      TIME1          ;UPDATE TIMEOUT COUNTER.
1895 012714 032775 000200 002524    BIT      #TS.SSR,@TSSR(R5) ;UNTIL #TS.SSR SET IN @TSSR(R5) OR TIME1 EQ #0
1896 012722 001003          BNE      50117$
1897 012724 005737 003436          TST      TIME1
1898 012730 001366          BNE      50116$
1899 012732          50117$:
1900 012732 000207          RTS      PC              ;RETURN.
1901
1902
1903
1904          ;          SUBROUTINE TO MOVE THE CURRENT MESSAGE PACKET TO THE COMMON AREA AND
1905          ;          TO UPDATE THE CURRENT TERMINATION CLASS CODE.
1906          ;          INPUTS:
1907          ;          OUTPUTS:
1908          ;          REGISTERS:      R2, R3.
1909          ;          CALLS:
1910
1911 012734 017537 002524 003454    MOVMSG:;MOV      @TSSR(R5),TSSREG      ;FREEZE THE STATUS REG CONTENTS
1912 012742 013702 003454          MOV      TSSREG,R2          ;EXTRACT THE TERMINATION CLASS CODE.
1913 012746 042702 177761          BIC      #TSC.TCC,R2
1914 012752 010237 003450          MOV      R2,CTCC          ;AND SAVE IT
1915 012756 006237 003450          ASR      CTCC
1916 012762 016503 002544          MOV      MSGPKA(R5),R3      ;ADR OF THIS DEVICE'S MSG.
1917 012766 005002          CLR      R2              ;CLR COUNTER.
1918 012770          50120$:
1919 012770 020227 000020          CMP      R2,#MSGCNT        ;WHILE THERE ARE MORE LOCATIONS:
1920 012774 001405          BEQ      50121$
1921 012778 012362 002354          MOV      (R3)+,MSGPKT(R2)  ;MOVE MSG TO COMMON AREA.
1922 013002 062702 000002          ADD      #2,R2            ;UPDATE COUNTER.
1923

```

GLOBAL SUBROUTINES SECTION

```

1924 013006 000770 BR 50120$
1925 013010 50121$:
1926 013010 013737 002362 003502 MOV MSGPKT+MS.XS0,EOTFLG ;MOVE XSTATO TO EOT FLAG.
1927 013016 000207 RTS PC
1928
1929 ; SUBROUTINE TO ADJUST THE RECORD COUNT.
1930 ; INPUTS:
1931 ; OUTPUTS:
1932 ; REGISTERS:
1933 ; CALLS:
1934
1935 013020 105737 003465 RECUD:: TSTB RECLG ;IF RECORD HAS NOT BEEN LOGGED:
1936 013024 001070 BNE 50122$
1937 013026 005365 003376 DEC RECCNT(R5) ;LET RECCNT(R5) := RECCNT(R5) - #1
1938 013032 032737 000001 003450 BIT #BIT0,CTCC ;IF TAPE MOVED
1939 013040 001057 BNE 50123$
1940 013042 032737 100000 002366 BIT #X2.OPM,MSGPKT+MS.XS2
1941 013050 001453 BEQ 50123$
1942 013052 105237 003465 INCB RECLG ;SET RECORD LOGGED.
1943 013056 023727 003420 102010 CMP CMDWRD,#RWD ;IF THIS IS A REWIND CMD:
1944 013064 001003 BNE 50124$
1945 013066 005065 003376 CLR RECCNT(R5) ;CLEAR RECORD COUNT,
1946
1947 013072 000442 BR 50125$
1948 013074 50124$:
1949 013074 032737 004000 003420 BIT #BRF.C,CMDWRD ;IF BRF USED, UPDATE RECORD COUNT.
1950 013102 001436 BEQ 50126$
1951 013104 032737 000400 003420 BIT #MOD.CO,CMDWRD ;IF A FORWARD CMD:
1952 013112 001007 BNE 50127$
1953 013114 032737 000400 003424 BIT #MOD.CO,PCMDWD ;IF PREV CMD WAS A FWD ALSO:
1954 013122 001002 BNE 50130$
1955 013124 005265 003376 INC RECCNT(R5) ;INCREMENT RECORD COUNT.
1956
1957 013130 50130$:
1958 ;IF REVERSE CMD:
1959 013130 000423 BR 50131$
1960 013132 50127$:
1961 013132 032737 000400 003424 BIT #MOD.CO,PCMDWD ;IF PREVIOUS CMD WAS A REV ALSO:
1962 013140 001417 BEQ 50132$
1963 013142 032765 000002 003502 BIT #X0.BOT,EOTFLG(R5) ;WHEN NOT AT BOT THEN
1964 013150 001013 BNE 50133$
1965 013152 105737 003471 TSTB ERRREC ;CHECK THE ERROR RETRY INDICATOR
1966 013156 001406 BEQ 2$ ;BR, IF WE ARE NOT NOW IN ERROR RETRY
1967 013160 105737 003516 TSTB VFYFLG ;CHECK THE WRITE VERIFY INDICATOR
1968 013164 001403 BEQ 2$ ;BR, IF WE ARE NOT IN WRT/VFY MODE
1969 013166 105737 003465 TSTB RECLG ;CHECK IF THIS RECORD HAS BEEN COUNTED
1970 013172 001002 BNE 10$ ;BR, IF HAVE ALREADY BUMPED RECORD CNTR.
1971 013174 005365 003376 2$: DEC RECCNT(R5) ;DECREMENT RECORD COUNT.
1972 013200 10$:
1973
1974 013200 50133$:
1975
1976 013200 50132$:
1977
1978 013200 50131$:
1979
1980

```

GLOBAL SUBROUTINES SECTION

```

1981 013200          50126$:
1982
1983 013200          50125$:
1984
1985 013200          50123$:
1986 013200 016577 003376 170200  MOV      RECCNT(R5),@DATAWT      ;LET @DATAWT := RECCNT(R5)
1987
1988 013206          50122$:
1989 013206 000207      RTS      PC              ;RETURN.
1990
1991      ;      THIS IS THE ERROR CHECK SUBROUTINE, AFTER INTERRUPT THIS
1992      ;      SUBROUTINE IS CALLED TO CHECK THE TSO5 STATUS.
1993      ;      IF SPECIAL COND IS SET THEN THE TCC HANDLING SUBROUTINE IS ENTERED.
1994      ;      IF THE RFC IS NON ZERO FOR A COMMAND REQUIRING A BPCR,
1995      ;      THEN AN ERROR RFC IS REPORTED,
1996      ;      INPUTS:
1997      ;      OUTPUTS:
1998      ;      REGISTERS:      R2, R4.
1999      ;      CALLS:      TCC0-TCC7.
2000
2001 013210 032737 100000 003454  CHKERR:; BIT      @TS.SC,TSSREG      ;IF SPECIAL COND STATUS IS SET THEN:
2002 013216 001441      BEQ      50134$
2003 013220 023727 003450 000002      CMP      CTCC,@2      ;IF TCC IS NOT ? THEN:
2004 013226 001405      BFQ      50135$
2005 013230 105737 003471      TSTB     ERRREC      ;IF NOT IN ERROR RECOVERY:
2006 013234 001002      BNE      50136$
2007 013236 005265 003336      INC      SCCNT(R5)      ;INC SC COUNTER.
2008
2009 013242          50136$:
2010
2011 013242          50135$:
2012 013242 032737 004000 003454      BIT      @TS.NXM,TSSREG      ;WHEN NON-EXISTANT MEMO
2013 013250 001004      BNE      50137$
2014 013252 032737 040000 003454      BIT      @TS.UPE,TSSREG
2015 013260 001412      BEQ      50140$
2016 013262          50137$:
2017 013262 032737 100000 002366      BIT      @X2.GPM,MSGPKT+M5,XS2      ;AND TAPE NOT MOVED
2018 013270 001003      BNF      50141$
2019 013272 012702 000005      MOV      @5,R2      ;SET TCC5 INDEX
2020
2021 013276 000402      BR      50142$
2022 013300          50141$:
2023 013300 012702 000004      MOV      @4,R2      ;TAPE MOVED, SET TCC4 INDEX
2024
2025 013304          50142$:
2026
2027 013304 000402      BR      50143$
2028 013306          50140$:
2029 013306 013702 002450      MOV      CTCC,R2      ;SET DETECTED TCC INDEX
2030
2031 013312          50143$:
2032 013312 006302      ASL      R2      ;CURRENT TCC X 2.
2033 013314 004772 013414      JSR      PC,@TCCRA(R2)      ;GO TO THE TCC HANDLING SUBROUTINE.
2034
2035 013320 000426      BR      50144$
2036 013322          50134$:
2037 013322 032737 004000 003420      BIT      @BRF.C,CMDWRD      ;IF BRF IS USED IN THIS CMD THEN:

```

GLOBAL SUBROUTINES SECTION

```

2038 013330 001422          BEQ      501451
2039 013332 005737 002360    TST      MSGPKT,MS,RFC          ;IF THERE IS AN RFC THEN:
2040 013336 001417          BEQ      501461
2041 013340 105737 003515    TSTB    RANDOM          ;IFB RANDOM EQ 40 ORB VFYFLG NE 40 THEN
2042 013344 001403          BEQ      501471
2043 013346 105737 003516    TSTB    VFYFLG
2044 013352 001411          BEQ      501501
2045 013354          501471:
2046
2047 013354 105737 003521    TSTB    IRE          ;IF NOT IN RANDOM OR IF CMD IS WTV:
2048 013360 001006          BNE      501511          ;IF RFC ERROR REPORTS ARE ALLOWED:
2049 013362 005265 003356    INC      HRDCNT(R5)          ;UPDATE HARD ERROR COUNT
2050 013366          ERRHRD 13,RFCERM,STAERM      ;REPORT RFC ERROR
          013366 104456          TRAP    C1ERRRD
          013370 000015          .WORD  13
          013372 004521          .WORD  RFCERM
          013374 006120          .WORD  STAERM
2051
2052 013376          501511:
2053
2054 013376          501501:
2055
2056 013376          501461:
2057
2058 013376          501451:
2059
2060 013376          501441:
2061 013376 105737 003467    TSTB    RWERR          ;IF A READ/WRITE ERROR HAS OCCURRED THEN:
2062 013402 001403          BEQ      501521
2063 013404 013737 003422 002330  MOV      CMDSAV,CMOPKT      ;RESTORE CMD PACKET AFTER ERROR RECV.
2064
2065 013412          501521:
2066 013412 000207    RTS      PC          ;RETURN.
2067
2068          ; ADDRESSES OF TCC HANDLING ROUTINES FOR TERMINATION CLASS CODES 0 - 7.
2069
2070 013414 013434    TCCRA: TCC0
2071 013416 013452    TCC1
2072 013420 013470    TCC2
2073 013422 013600    TCC3
2074 013424 013616    TCC4
2075 013426 014232    TCC5
2076 013430 014330    TCC6
2077 013432 014472    TCC7
2078
2079          ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 0, UNDEFINED SPECIAL
2080          ; CONDITION ERROR.
2081          ; INPUTS:
2082          ; OUTPUTS:
2083          ; REGISTERS:
2084          ; CALLS:
2085
2086 013434 005265 003356    TCC0:: INC      HRDCNT(R5)          ;UPDATE HARD ERROR COUNT.
2087 013440          ERRHRD 5,SCERM,STAERM      ;REPORT SPECIAL CONDITION ERROR.
          013440 104456          TRAP    C1ERRRD
          013442 000005          .WORD  5
          013444 004475          .WORD  SCERM

```

GLOBAL SUBROUTINES SECTION

```

2088 013446 006120
      013450 000207          RTS PC          ;RETURN.          .WORD STAERM
2089
2090
2091          ;          SUBROUTINE TO HANDLE TERMINATION CLASS CODE 1, ATTENTION CONDITION.
2092          ;          THIS TCC INDICATES THAT THE DRIVE HAS UNDERGONE A STATUS CHANGE
2093          ;          SUCH AS GOING OFFLINE OR COMING ONLINE.
2094          ;          INPUTS:
2095          ;          OUTPUTS:
2096          ;          REGISTERS:      R2,R4
2097          ;          CALLS:          DROPU
2098
2099 013452          TCC1:: ERRDF 6,ATTNM,STAERM          ;REPORT ATTENTION-UNIT OFF LINE.
      013452 104455          TRAP CERRDF
      013454 000006          .WORD 6
      013456 004603          .WORD ATTNM
      013460 006120          .WORD STAERM
2100 013462 004737 017200          JSR PC,DROPU          ;DROP THE UNIT.
2101 013466 000207          RTS PC          ;RETURN.
2102
2103          ;          SUBROUTINE TO HANDLE TERMINATION CLASS CODE 2, TAPE STATUS ALERT.
2104          ;          A STATUS CONDITION HAS BEEN ENCOUNTERED THAT MAY HAVE SIGNIFICANCE
2105          ;          TO THE PROGRAM. BITS OF INTEREST INCLUDE TMK, RLS, LET, RLL, BOT, EOT.
2106          ;          INPUTS:
2107          ;          OUTPUTS:
2108          ;          REGISTERS:
2109          ;          CALLS:
2110
2111 013470 032737 000002 002362 TCC2:: BIT @X0.BOT,MSGPKT+MS.XSO
2112 013476 001404          BEQ 50153#
2113 013500 105737 003514          TSTB EXPBOT
2114 013504 001401          BEQ 50153#
2115
2116 013506 000433          BR TC2RTN          ;IF AT BOT AND BOT IS EXPECTED:
2117          ;RETURN-TCC2 CAUSED BY EXPECTED BOT.
2118 013510          50153#:
2119 013510 032737 170002 002362 BIT @X0.RLS!X0.RLL!X0.TMK!X0.LET!X0.BOT,MSGPKT+MS.XSO
2120          ;IF @X0.RLS!X0.RLL!X0.TMK!X0.LET!X0.BOT SET IN MSGPKT+MS.XSO THEN
2121
2122 013516 001427          BEQ 50154#
2123
2124 013520 105737 003515          TSTB RANDOM          ;IF TCC2 CAUSED BY ANYTHING BUT EOT:
2125 013524 001403          BEQ 50155#          ;IFB RANDOM EQ #0 ORB VFYFLG NE #0 THEN
2126 013526 105737 003516          TSTB VFYFLG
2127 013532 001421          BEQ 50156#
2128 013534          50155#:
2129
2130 013534 105737 003521          TSTB IRE          ;IF NOT IN RANDOM OR IF CMD IS WTV:
2131 013540 001016          BNE 50157#          ;IF RFC ERROR REPORTS ARE ALLOWED:
2132 013542 105737 003471          TSTB ERRREC          ;IF WE ARE IN ERROR RECOVERY THEN:
2133 013546 001403          BEQ 50160#
2134 013550 105237 003470          INCB UNREC          ;SET UNRECOVERABLE FLAG FOR LOG.
2135          ;ELSE - IF NOT IN ERROR RECOVERY:
2136 013554 000402          BR 50161#
2137 013556          50160#:
2138 013556 005265 003336          INC SCCNT,R5          ;INCREMENT THE SPEC COND COUNTER.
2139

```

GLOBAL SUBROUTINES SECTION

```

2140 013562          50161$:
2141 013562 005265 003356      INC      HRDCNT(R5)          ;UPDATE HARD ERROR COUNT.
2142 013566          ERRHRD 7,TSAM,STAERM      ;REPORT TAPE STATUS ALERT.
          013566 104456          TRAP      C$ERRHRD
          013570 000007          .WORD    7
          013572 004705          .WORD    TSAM
          013574 006120          .WORD    STAERM
2143
2144 013576          50157$:
2145
2146 013576          50156$:
2147
2148 013576          50154$:
2149
2150 013576 000207      TC2RTN:  RTS  PC          ;RETURN.
2151
2152
2153          ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 3, FUNCTION REJECT.
2154          ; THE SPECIFIED FUNCTION WAS NOT INITIATED. BITS OF INTEREST ARE
2155          ; RMR, OFL, VCK, BOT, ILC, WLE, ILA, AND NBA.
2156          ; INPUTS:
2157          ; OUTPUTS:
2158          ; REGISTERS:      R2,R4
2159          ; CALLS:          DROPU
2160
2161 013600          TCC3::  ERRDF  8,FUNRM,STAERM      ;REPORT FUNCTION REJECT.
          013600 104455          TRAP      C$ERRDF
          013602 000010          .WORD    8
          013604 004622          .WORD    FUNRM
          013606 006120          .WORD    STAERM
2162 013610 004737 017200      JSR      PC,DROPU          ;DROP THE UNIT.
2163 013614 000207      RTS      PC          ;RETURN.
2164
2165          ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 4, RECOVERABLE ERROR.
2166          ; TAPE POSITION IS ONE RECORD BEYOND WHAT ITS POSITION WAS WHEN
2167          ; THE FUNCTION WAS INITIATED. RECOVERY PROCEDURE IS TO LOG THE
2168          ; ERROR AND ISSUE THE APPROPRIATE RETRY COMMAND.
2169          ; 2 WRITE-ERROR-RECOVERY ALGORITHMS CAN BE SELECTED:
2170          ; THE FIRST ONE, VIA BADTSW SWITCH, DOES DETECT BAD SPOTS ON TAPE.
2171          ; IT CALLS A WRITE RETRY SUBR UNTIL THE RECORD IS RECOVERED
2172          ; OR 20 BAD SPOTS HAVE BEEN LOGGED. ON REACHING 20 BAD
2173          ; SPOTS LOGGED, A BAD TAPE OVERFLOW MSG IS PRINTED AND THE
2174          ; UNIT DROPPED.
2175          ; THE SECOND ALGORITHM ISSUES THE TS05 WRITE RETRY COMMAND
2176          ; UP TO 16 TIMES BEFORE DROPPING THE UNIT OR PROCEEDING
2177          ; WITH THE NEXT RECORD ON RECOVERY.
2178          ; INPUTS:
2179          ; OUTPUTS:
2180          ; REGISTERS:      R2,R4,
2181          ; CALLS:          RTLE, EXECUTE, GOWAIT, DROPU, WRTY
2182
2183 013616 023727 003426 000002  TCC4::  CMP      CMDLG,#2          ;IF CMDLG EQ #2 ANDB BADTSW NE #0 THEN
2184 013624 001125          BNE      50162$
2185 013626 105737 002210          TSTB   BADTSW
2186 013632 001522          BEQ     50162$
2187 013634 105737 003471          TSTB   ERRREC          ;IFB ERRREC EQ #0 ANDB ERCOVER NE #0 THEN
2188 013640 001007          BNE     50163$

```

GLOBAL SUBROUTINES SECTION

```

2189 013642 105737 002207      TSTB  ERVER
2190 013646 001404          BEQ   50163$
2191 013650          ERRSOFT 9,RERM,STAERM ;
      013650 104457
      013652 000011          TRAP  C$ERSOFT
      013654 005017          .WORD 9
      013656 006120          .WORD RERM
                                .WORD STAERM
2192
2193 013660          50163$:
2194 013660 105737 002213      TSTB  IREC ;IFB IREC EQ #0 THEN
2195 013664 001102          BNE  50164$
2196 013666 105237 003471      INCB  ERRREC ;RETRY FLAG FOR EXECUTE SUBR: DON'T UPDATE REC CN
2197 013672 105237 003464      INCB  WRTYER ;REWRITE ERROR FLAG FOR WRTY SUBR
2198 013676 105737 003463      TSTB  WRTYFG ;FIRST RETRY ON THIS RECORD: SUBSEQUENT
2199 013702 001072          BNE  50165$
2200
      ;RETRIES WITH TCC4 ERRORS BY-PASS THIS SECTION
2201 013704 013737 003420 015106      MOV  CMDWRD,WTYWRD ;SAVE WRITE COMMAND PACKET
2202 013712 013737 002330 015104      MOV  CMDPKT,WTYCMD
2203 013720 013737 002336 015110      MOV  CMDPKT*CP,CNT,WTYBRF
2204 013726 105237 003467          INCB  RWERR ;LOG SUBR FLAG: COUNT WRT ERRORS
2205 013732 105237 003463          INCB  WRTYFG ;RETRY IN PROGRESS FLAG
2206
2207 013736          50166$: ;REPEAT
2208 013736 005265 003316          INC  WRTYCT(R5) ;COUNT GLOBAL WRITE RETRIES
2209 013742 005037 003460          CLR  RETRYC ;CLEAR # OF RETRIES PER RECORD
2210 013746 105037 003462          CLR  RPTCNT ;CLEAR # OF REPEATS
2211 013752 004737 014636          JSR  PC,WRTY ;CALL WRITE RETRY
2212 013756 105737 003464          TSTB WRTYER ;REPEAT RETRIES ON SAME RECORD
2213 013762 001404          BEQ  50167$
2214 013764 027727 167522 000050      CMP  #BTPT,#40.
2215 013772 103761          BLO  50166$
2216 013774          50167$:
2217
      ;UNTIL RECOVERED OR 20 BAD SPOTS
2218 013774 027727 167512 000050      CMP  #BTPT,#40. ;WHEN 20 BAD SPOTS LOGGED
2219 014002 103423          BLO  50170$
2220 014004          PRINTB #BTMSG2 ;PRINT BAD TAPE OVERFLOW MSG
      014004 012746 015177          MOV  #BTMSG2,-(SP)
      014010 012746 000001          MOV  #1,-(SP)
      014014 010600          MOV  SP,RO
      014016 104414          TRAP C$PNTB
      014020 062706 000004          ADD  #4,SP
2221 014024 004737 015316          JSR  PC,BOREPS ;ERASE BAD RECORD
2222 014030 005365 003376          DEC  RECCNT(R5)
2223 014034 004737 017200          JSR  PC,DROPU ;DROP UNIT
2224 014040 005065 003376          CLR  RECCNT(R5)
2225 014044 012775 002350 002514      MOV  #RWCPK,#TSD8(R5) ;REWIND UNIT
2226
2227 014052          50170$:
2228 014052 105037 003463          CLR  WRTYFG ;RETRY COMPLETE FLAG
2229 014056 105237 003531          INCB MISCFC ;DO NOT HALT ON THIS CMD FLG
2230 014062 013737 015106 003424      MOV  WTYWRD,PCMDWD ;RESTORE ORIGINAL WRT CMD AFTER RECOVERY
2231
2232 014070          50165$:
2233
2234 014070 000402          BR   50171$
2235 014072          50164$:
2236 014072 105237 003470          INCB UNREC ;LET UNREC :B= UNREC + #1 ;

```

7

GLOBAL SUBROUTINES SECTION

```

2237
2238 014076          50171$:
2239
2240 014076 000454          BR      50172$
2241 014100          50162$:
2242 014100 004737 014510      JSR      PC,RTLE          ;CHECK FOR RETRY LIMIT EXCEEDED.
2243 014104 023727 003426 000002      CMP      CMDLG,#2          ;IF READ CMD THEN:
2244 014112 003411          BLE      50173$
2245 014114 012702 000020      MOV      #RRCL,R2          ;R2=READ RETRY COUNT LIMIT / 2
2246 014120 006202          ASR      R2
2247 014122 023702 003460      CMP      RETRYC,R2          ;IF RETRY COUNT IS MORE THAN HALF LIMIT:
2248 014126 002403          BLT      50174$
2249 014130 052737 0200C0 002330      BIS      #OPP.C,CMDPKT          ;SET OPPOSITE BIT FOR RETRY2.
2250
2251 014136          50174$:
2252
2253 014136          50173$:
2254 014136 005737 003460      TST      RETRYC          ;IF THIS IS THE ORIGINAL ERROR THEN:
2255 014142 001007          BNE      50175$
2256 014144 105737 002207      TSTB     ERVER
2257 014150 001404          BEQ      50175$
2258 014152          ERRSOFT 9,RERM,STAERM ;REPORT RECOVERABLE ERROR
2259
2260 014162          TRAP      C#ERSOFT
2261 014162 005237 003460          .WORD     9
2262 014166 052737 001000 002330      .WORD     RERM
2263 014174 105737 002213          .WORD     STAERM
2264 014200 001011          ;PROVIDED OPERATOR HAS ENABLED THE REPORT
2265 014202 105237 003471          50175$:
2266 014206 0126C          INC      RETRYC          ;UPDATE RETRY COUNT.
2267 014210 012602          BIS      #MOD.C1,CMDPKT ;SET RETRY BIT IN CMD PACKET.
2268 014212 004737 012054          TSTB     IREC          ;IF ERROR RECOVERY ENABLED:
2269 014216 000137 012364          BNE      50176$
2270
2271 014222 000402          INCB     ERRREC          ;SET ERROR RECOVERY FLAG.
2272 014224          MOV      (SP)+,R2          ;POP 2 RTN ADRS FROM STACK.
2273 014224 105237 003470          MOV      (SP)+,R2
2274
2275 014230          JSR      PC,EXCUTE          ;GO EXECUTE THE RETRY COMMAND.
2276
2277 014230          JMP      GOWAIT          ;GO WAIT FOR INTERRUPT + CHECK STATUS.
2278 014230 000207          BR      50177$          ;ELSE IF ERROR RECOVERY IS NOT ENABLED:
2279
2280          50176$:
2281          INCB     UNREC          ;SET UNRECOVERABLE ERROR FLAG.
2282
2283          50177$:
2284
2285          50172$:
2286          RTS PC          ;RETURN
2287
2288          ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 5, RECOVERABLE ERROR.
2289          ; TAPE POSITION HAS NOT CHANGED. RECOVERY PROCEDURE IS TO LOG THE
2290          ; ERROR AND RE-ISSUE THE ORIGINAL COMMAND.
2291          ; INPUTS:
2292          ; OUTPUTS:
2293          ; REGISTERS: R2,R4.
2294          ; CALLS: RTLE, EXCUTE, GOWAIT, DROPU.
2295
2296          TCC5:: JSR      PC,RTLE          ;CHECK FOR RETRY LIMIT EXCEEDED
2297          TST      RETRYC          ;IF THIS IS THE ORIGINAL ERROR THEN:

```

GLOBAL SUBROUTINES SECTION

```

2290 014242 001004          BNE      50200$
2291 014244          ERRSOF 10,RERM,STAERM ;REPORT RECOVERABLE ERROR.
      014244 104457          TRAP    C$ERSOFT
      014246 000012          .WORD  10
      014250 005017          .WORD  RERM
      014252 006120          .WORD  STAERM
2292 014254          50200$:
2293 014254 005237 003460  INC      RETRYC          ;UPDATE RETRY COUNTER.
2294 014260 105737 002213  TSTB    IREC            ;IF ERROR RECOVERY IS ENABLED:
2295 014264 001016          BNE      50201$
2296 014266 105237 003471  INCB    ERRREC          ;SET ERROR RECOVERY FLAG.
2297 014272 005265 003376  INC      RECCNT(R5)      ;UPDATE REC COUNT
2298 014276 016577 003376 167102  MOV     RECCNT(R5),DATAWT ;AND INSERT IT INTO WRT BFR
2299 014304 012602          MOV     (SP)+,R2        ;POP 2 RTN ADRS FROM STACK.
2300 014306 012602          MOV     (SP)+,R2
2301 014310 004737 012054  JSR     PC,EXCUTE        ;GO RE-ISSUE THE COMMAND.
2302 014314 000137 012364  JMP     GOWAIT           ;GO WAIT FOR INTERRUPT + CHECK STATUS.
2303                                ;ELSE IF ERROR RECOVERY IS NOT ENABLED:
2304 014320 000402          BR      50202$
2305 014322          50201$:
2306 014322 105237 003470  INCB    UNREC            ;SET UNRECOVERABLE ERROR FLAG.
2307
2308 014326          50202$:
2309 014326 000207          RTS     PC                ;RETURN.
2310
2311
2312 ;      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 6, UNRECOVERABLE ERROR.
2313 ;      TAPE POSITION HAS BEEN LOST. THE ONLY VALID RECOVERY PROCEDURE
2314 ;      IS TO REWIND AND START OVER AT BOT UNLESS THE TAPE HAS LABELS OR
2315 ;      SEQUENCE NUMBERS. THIS DIAGNOSTIC WILL REWIND AND RETRY THE
2316 ;      COMMAND ONLY IF DENSITY CHECK IS SET, OTHERWISE THE UNIT WILL BE
2317 ;      DROPPED FROM THE TEST SEQUENCE.
2318 ;      INPUTS:
2319 ;      OUTPUTS:
2320 ;      REGISTERS:      R2, R4
2321 ;      CALLS:          RTLE, WSSR, EXCUTE, GOWAIT, DROPU
2322
2323 014330 033737 000010 002370 TCC6:: BIT      X3,DCK,MSGPKT+MS,XS3;IF X3.DCK NOTSETIN MSGPKT+MS,XS3 THEN
2324 014336 001016          BNE      50203$
2325                                ;IF THERE IS NO DENSITY CHECK THEN:
2326 014340 005737 003426  TST     CMDLG            ;IF CMD IS A READ OR WRITE THEN:
2327 014344 001404          BEQ     50204$
2328 014346 105237 003467  INCB    RWERR            ;SET RD/WR ERROR FLAG.
2329 014352 105237 003470  INCB    UNREC            ;SET UNRECOVERABLE ERROR FLAG.
2330
2331 014356          50204$:
2332 014356          ERRDF 11,URERM,STAERM ;REPORT UNRECOVERABLE ERROR.
      014356 104455          TRAP    C$ERDF
      014360 000013          .WORD  11
      014362 005041          .WORD  URERM
      014364 006120          .WORD  STAERM
2333 014366 004737 017200  JSR     PC,DROPU        ;REPORT ERROR + DROP UNIT.
2334                                ;ELSE-IF THERE IS DENSITY CHECK:
2335 014372 000436          BR      50205$
2336 014374          50203$:
2337 014374 004737 014510  JSR     PC,RTLE          ;CHECK FOR RETRY LIMIT EXCEEDED.
2338 014400 005737 003460  TST     RETRYC          ;IF THIS IS THE ORIGINAL ERROR THEN:

```

GLOBAL SUBROUTINES SECTION

```

2339 014404 001004          BNE      50206$
2340 014406          ERRSOFT 11,URERM,STAERM          ;REPORT DENSITY CHECK ERROR
          014406 104457          TRAP      C$ERRSOFT
          014410 000013          .WORD    11
          014412 005041          .WORD    URERM
          014414 006120          .WORD    STAERM
2341
2342 014416          50206$:
2343 014416 005237 003460    INC      RETRYC          ;UPDATE RETRY COUNT.
2344 014422 105737 003521    TSTB   IRE            ;IF ERROR RECOVERY IS ENABLED THEN:
2345 014426 001016          BNE      50207$
2346 014430 105237 003471    INCB   ERRREC          ;SET ERROR RECOVERY FLAG,
2347 014434 012775 002350 002514    MOV    @RWCPK,@TSDB(R5) ;ISSUE A REWIND COMMAND,
2348 014442 004737 012700    JSR    PC,WSSR          ;WAIT FOR SUBSYSTEM READY,
2349 014446 012602          MOV    (SP)+,R2          ;POP 2 RTN ADRS FROM STACK.
2350 014450 012602          MOV    (SP)+,R2
2351 014452 004737 012054    JSR    PC,EXCUTE
2352 014456 000137 012364    JMP    GOWAIT          ;REISSUE THE COMMAND,
2353                                ;WAIT FOR INTERRUPT
2354 014462 000402          BR      50210$          ;ELSE-IF ERR REC DISABLED:
2355 014464          50207$:
2356 014464 105237 003470    INCB   UNREC          ;SET UNRECOVERABLE ERROR FLAG.
2357
2358 014470          50210$:
2359
2360 014470          50205$:
2361 014470 000207          RTS     PC              ;RETURN
2362
2363 ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 7, FATAL SUBSYSTEM
2364 ; ERROR. THE SUBSYSTEM IS INCAPABLE OF PROPERLY PERFORMING
2365 ; COMMANDS OR AT LEAST ITS INTEGRITY IS SERIOUSLY QUESTIONABLE.
2366 ; REFER TO THE FATAL CLASS CODE FIELD IN THE TSSR REGISTER FOR
2367 ; ADDITIONAL INFORMATION ON THE TYPE OF FATAL ERROR.
2368 ; INPUTS:
2369 ; OUTPUTS:
2370 ; REGISTERS: R2, R4
2371 ; CALLS:
2372
2373 014472          TCC7:: ERRDF 12,FATSM,STAERM          ;REPORT FATAL SUBSYSTEM ERROR.
          014472 104455          TRAP      C$ERRDF
          014474 000014          .WORD    12
          014476 004642          .WORD    FATSM
          014500 006120          .WORD    STAERM
2374 014502 004737 017200    JSR    PC,DROPU          ;DROP THE UNIT.
2375 014506 000207          RTS     PC              ;RETURN.
2376
2377
2378 ; SUBROUTINE TO CHECK FOR RETRY LIMIT EXCEEDED. PRINTS ERROR MESSAGE
2379 ; IF EXCEEDED AND DROP UNIT UNLESS COMMAND IS A READ.
2380 ; INPUTS:
2381 ; OUTPUTS:
2382 ; REGISTERS: R2, R4.
2383 ; CALLS: DROPU
2384
2385 014510 005737 003426    RILE:: TST    CMDLG          ;IF CMD IS NOT A READ OR WRITE THEN:
2386 014514 001010          BNE      50211$
2387 014516          ERRDF 11,URERM,STAERM ;REPORT UNRECOVERABLE ERROR.

```

GLOBAL SUBROUTINES SECTION

```

014516 104455 TRAP C$ERDF
014520 000013 .WORD 11
014522 005041 .WORD URERM
014524 006120 .WORD STAERM
2388 014526 004737 017200 JSR PC,DROPU ;DROP THE UNIT.
2389 014532 012602 MOV (SP)+,R2 ;POP RTN ADRS FROM STACK.
2390 014534 000437 BR RTLRTN ;AND RETURN.
2391
2392 014536 50211$:
2393 014536 105237 003467 INCB RWERR ;SET READ/WRITE ERROR FLAG.
2394 014542 023727 003426 000002 CMP CMDLG,#2 ;IF CMD IS A WRT OR WTM:
2395 014550 001016 BNE 50212$
2396 014552 023727 003460 000020 CMP RETRYC,#WRECL ;IF RETRY COUNT HAS REACHED LIMIT:
2397 014560 001011 BNE 50213$
2398 014562 105237 003470 INCB UNREC ;SET UNRECOVERABLE FLAG
2399 014566 ERRDF 14,RLEXM,STAERM ;REPORT RETRY LIMIT EXCEEDED.
014566 104455 TRAP C$ERDF
014570 000016 .WORD 14
014572 004556 .WORD RLEXM
014574 006120 .WORD STAERM
2400 014576 004737 017200 JSR PC,DROPU ;DROP THE UNIT.
2401 014602 012602 MOV (SP)+,R2 ;POP 2 RTN ADRS FROM STACK.
2402 014604 50213$:
2403 ;ELSE - CMD IS A READ:
2404 014604 000413 BR 50214$
2405 014606 50212$:
2406 014606 023727 003460 000020 CMP RETRYC,#RRECL ;IF RETRY COUNT HAS REACHED LIMIT:
2407 014614 001007 BNE 50215$
2408 014616 105237 003470 INCB UNREC ;SET UNRECOVERABLE FLAG
2409 014622 ERRHRD 14,RLEXM,STAERM ;REPORT RECOVERABLE ERROR.
014622 104456 TRAP C$ERHRD
014624 000016 .WORD 14
014626 004556 .WORD RLEXM
014630 006120 .WORD STAERM
2410 014632 012602 MOV (SP)+,R2 ;POP 2 RTN ADRS FROM STACK.
2411 014634 50215$:
2412
2413 014634 50214$:
2414 014634 000207 RTLRTN: RTS PC ;RETURN
2415
2416 ; SUBR TO REWRITE A BAD, BUT RECOVERABLE WRITTEN RECORD.
2417 ; REWRITE RECORD ON SAME SPOT: REPEAT 4 TIMES.
2418 ; IF ALL 4 REPEATS GOOD, RECORD IS RECOVERED
2419 ; AND A RECOVERABLE WRITE ERROR IS LOGGED.
2420 ; IF ANY OF 4 REPEATS BAD, ERASE BAD RECORD, LOG SUSPECTED
2421 ; BAD SPOT, RETRY AGAIN. RETRY 4 TIMES, UP TO 4 REPEATS EACH.
2422 ; IF RECORD NOT GOOD AFTER 4 RETRIES, ERASE IT, EXIT WITH
2423 ; ERROR FLAG WRTYER SET, PRINTING RETRY FAILED.
2424 ; THIS ALL SCHEME IS REENTERED 20 TIMES MAX, IE 20 BAD
2425 ; SPOTS MAX ARE ALLOWED.
2426
2427 ; INPUTS:
2428 ; OUTPUTS:
2429 ; REGISTERS: R3,R4
2430 ; CALLS: BORERS, REWRT
2431
2432 014636 WRTY:: ;BEGIN RETRY ;REPEAT

```

GLOBAL SUBROUTINES SECTION

```

2433
2434 014636          50217$:
2435                  ;BEGIN REPEAT
2436                  ;REPEAT
2437 014636          50221$:
2438 014636 004737 015316 JSR    PC,BORERS      ;BACKSPACE/ERASE ONE RECORD
2439 014642 105037 003464 CLR   WRTYER         ;CLEAR WRITE RETRY ERROR
2440 014646 004737 015472 JSR    PC,REWRT      ;REWRITE RECORD ON SAME SPOT
2441 014652 105237 003462 INCB  RPTCNT         ;COUNT REPEATS
2442 014656 123727 003462 000004 CMP   RPTCNT,#4     ;LIMIT: 4 REPEATS OR RECOVERE
2443 014664 001403      BEQ   50222$
2444 014666 105737 003464 TSTB  WRTYER
2445 014672 001761      BEQ   50221$
2446 014674          50222$:
2447                  ;END REPEAT
2448 014674          50220$:
2449 014674 005237 003460 INC   RETRYC         ;COUNT RETRIES
2450 014700 105737 003464 TSTB  WRTYER
2451 014704 001001      BNE   50223$
2452 014706 000457      BR    50216$      ;EXIT RETRY LOOP IF RECOVERED
2453
2454 014710          50223$:
2455 014710 105737 002207 TSTB  ERCVER        ;IFB ERCVER NE #0 THEN
2456 014714 001415      BEQ   50225$
2457 014716          PRINTB #BTMSG1,RETRYC,<B,RPTCNT> ;PRINT SUSPECTED BAD SPOT
          CLR    -(SP)
          BISB  RPTCNT,(SP)
          MOV   RETRYC,-(SP)
          MOV   #BTMSG1,-(SP)
          MOV   #3,-(SP)
          MOV   SP,R0
          TRAP C$PNTB
          ADD   #10,SP
2458 014750          50225$:
2459 014750 023727 003460 000001 CMP   RETRYC,#1     ;ON FIRST RETRY, LOGG BAD SPOT
2460 014756 001021      BNE   50226$
2461 014760 016537 002616 003512 MOV   BTADDR(R5),BTPT ;BTPT IS BOTH THE BAD SPOT COUNTER
2462 014766 017704 166520 MOV   #BTPT,R4      ;AND THE LOGGING INDEX
2463 014772 052704 000002 ADD   #2,R4
2464 014776 010477 156510 MOV   R4,#BTPT
2465 015002 020427 000050 CMP   R4,#40        ;IF R4 LOS #40. THEN
2466 015006 101005      BHI   50227$
2467 015010 013703 003512 MOV   BTPT,R3      ;STORE FIRST 20 BAD SPOTS
2468 015014 060304      ADD   R3,R4        ;LET R4 := R4 + R3
2469 015016 016514 003376 MOV   RECCNT(R5),(R4) ;LET (R4) := RECCNT(R5)
2470
2471 015022          50227$:
2472
2473 015022          50226$:
2474 015022 105237 003525 INCB  ERSFLG        ;ERASE FLAG TO ERASE BAD RECORD
2475 015026 105037 003467 CLR   RWERR         ;CANCELL "LOG" ERROR FLAG ON FAILING RET
2476 015032 105037 003462 CLR   RPTCNT        ;CLEAR REPEAT COUNT FOR NEXT RETRY
2477
2478 015036          50224$:
2479 015036 023727 003460 000004 CMP   RETRYC,#4     ;LIMIT: 4 RETRIES
2480 015044 001274      BNE   50217$
2481                  ;END RETRY

```

GLOBAL SUBROUTINES SECTION

```

2482 015046
2483 015046 105737 003464
2484 015052 001413
2485 015054 105737 002207
2486 015060 001410
2487 015062
    015062 012746 015247
    015066 012746 000001
    015072 010600
    015074 104414
    015076 062706 000004
2488
2489 015102
2490
2491 015102
2492 015102 000207
2493
2494 015104 000000
2495 015106 000000
2496 015110 000000
2497
2498 015112 045 101 123
    015115 125 123 120
    015120 105 103 124
    015123 040 102 101
    015126 104 040 123
    015131 120 117 124
    015134 040 101 106
    015137 124 105 122
    015142 040 045 104
    015145 061 045 101
    015150 040 122 105
    015153 124 122 131
    015156 054 040 045
    015161 104 061 045
    015164 101 040 122
    015167 105 120 105
    015172 101 124 045
    015175 116 000
2499 015177 045 116 045
    015202 101 102 101
    015205 104 040 124
    015210 101 120 105
    015213 040 117 126
    015216 105 122 106
    015221 114 117 127
    015224 072 040 103
    015227 110 101 116
    015232 107 105 040
    015235 124 101 120
    015240 105 041 045
    015243 116 045 116
    015246 000
2500 015247 045 101 122
    015252 105 124 122
    015255 131 040 106
    015260 101 111 114

50216$:
    TSTB  WRTYER ;IFB WRTYER NE #0 THEN
    BEQ   50230$
    TSTB  ERCOVER ;IFB ERCOVER NE #0 THEN
    BEQ   50231$
    PRINTB #BTMSG3 ;PRINT RETRI FAILED
    MOV   #BTMSG3, -(SP)
    MOV   #1, (SP)
    MOV   SP, R0
    TRAP  C$PNTB
    ADD   #4, SP

50231$:
50230$:
    RTS   PC

WTYCMD: .WORD 0 ;STORAGE FOR WRITE CMD WHILE RETRYING
WTYWRD: .WORD 0 ;STORAGE FOR WRITE CMD WORD WHILE RETRYING
WTYBRF: .WORD 0 ;STORAGE FOR WRITE BPCR WHILE RETRYING

BTMSG1: .ASCIZ /#ASUSPECT BAD SPOT AFTER #D1#A RETRY, #D1#A REPEAT#N/

BTMSG2: .ASCIZ /#N#ABAD TAPE OVERFLOW: CHANGE TAPE!#N#N/

BTMSG3: .ASCIZ /#ARETRY FAILED ON BAD SPOT...ERASED!#N#N/

```

GLOBAL SUBROUTINES SECTION

015263	105	104	040
015266	117	116	040
015271	102	101	104
015274	040	123	120
015277	117	124	056
015302	056	056	105
015305	122	101	123
015310	105	104	041
015313	045	116	000

```

2501                                     .EVEN
2502
2503                                     ; SUBR TO BACKSPACE ONE RECORD
2504                                     ; IF THE ERASE FLAG IS SET, THEN ERASE THAT RECORD
2505                                     ; INPUTS:          ERSFLG 1 = DO ERASE
2506                                     ; OUTPUTS:
2507                                     ; REGISTERS:
2508                                     ; CALLS:          EXECUTE, GOWAIT, CKHAE
2509
2510 015316 013737 003420 003424 BORERS: ;MOV      CMDWRD,PCMDWD ;SET COMMAND TO SPACE REV
2511 015324 012737 104410 003420          ;MOV      *SRR,CMDWRD ;LET CMDWRD := *SRR ;
2512 015332 013737 003420 002330          ;MOV      CMDWRD,CMDPKT ;LET CMDPKT := CMDWRD CLR.BY *BRF.C ;
2513 015340 042737 004000 002330          ;BIC      *BRF.C,CMDPKT
2514 015346 013737 002330 003422          ;MOV      CMDPKT,CMDSAV ;LET CMDSAV := CMDPKT ;
2515 015354 012737 000001 002332          ;MOV      *1,CMDPKT+CP.ADL ;LET CMDPKT+CP.ADL := *1 ;
2516 015362 005037 003426                  ;CLR      CMDLG ;LET CMDLG := *0 ;
2517 015366 004737 011064                  ;JSR      PC,CMDAC ;
2518 015372 004737 012054                  ;JSR      PC,EXECUTE ;
2519 015376 004737 012364                  ;JSR      PC,GOWAIT ;
2520 015402 004737 017500                  ;JSR      PC,CKHAE ;
2521 015406 105737 003525                  ;TSTB    ERSFLG ;WHEN ERASE FLAG IS SET, DO ERASE
2522 015412 001426 50232*
2523 015414 013737 003420 003424          ;MOV      CMDWRD,PCMDWD ;LET PCMDWD := CMDWRD ;
2524 015422 012737 100411 003420          ;MOV      *ERS,CMDWRD ;LET CMDWRD := *ERS ;
2525 015430 013737 003420 002330          ;MOV      CMDWRD,CMDPKT ;LET CMDPKT := CMDWRD ;
2526 015436 013737 002330 003422          ;MOV      CMDPKT,CMDSAV ;LET CMDSAV := CMDPKT ;
2527 015444 004737 011064                  ;JSR      PC,CMDAC ;
2528 015450 004737 012054                  ;JSR      PC,EXECUTE ;
2529 015454 004737 012364                  ;JSR      PC,GOWAIT ;
2530 015460 004737 017500                  ;JSR      PC,CKHAE ;
2531 015464 105037 003525                  ;CLRB    ERSFLG ;LET ERSFLG := *0
2532
2533 015470 50232*
2534 015470 000207 RTS PC
2535
2536                                     ; SUBR TO REWRITE A BADLY WRITTEN RECORD
2537
2538 015472 013737 003420 003424 REWRT: ;MOV      CMDWRD,PCMDWD ;RESTORE WRITE COMMAND PACKET
2539 015500 013737 015106 003420          ;MOV      WTYWRD,CMDWRD ;LET CMDWRD := WTYWRD ;
2540 015506 013737 015104 002330          ;MOV      WTYCMD,CMDPKT ;LET CMDPKT := WTYCMD ;
2541 015514 013737 002330 003422          ;MOV      CMDPKT,CMDSAV ;LET CMDSAV := CMDPKT ;
2542 015522 013737 003406 002332          ;MOV      DATAWT,CMDPKT+CP.ADL ;LET CMDPKT+CP.ADL := DATAWT ;
2543 015530 013737 015110 002336          ;MOV      WTYBRF,CMDPKT+CP.CNT ;LET CMDPKT+CP.CNT := WTYBRF ;
2544 015536 012737 000002 003426          ;MOV      *2,CMDLG ;LET CMDLG := *2 ;
2545 015544 004737 011064                  ;JSR      PC,CMDAC ;
2546 015550 004737 012054                  ;JSR      PC,EXECUTE ;RE-WRITE RECORD
2547 015554 004737 012364                  ;JSR      PC,GOWAIT ;
2548 015560 004737 017500                  ;JSR      PC,CKHAE ;

```

GLOBAL SUBROUTINES SECTION

```

2549 015564 000207      RTS      PC
2550
2551      ;      SUBROUTINE TO LOG BYTES READ/WRITTEN.
2552      ;      ALSO UPDATES READ/WRITE ERROR COUNTERS.
2553      ;      INPUTS:
2554      ;      OUTPUTS:
2555      ;      REGISTERS:      R2, R3, R4.
2556      ;      CALLS:
2557
2558 015566 105737 003466 LOG: :   TSTB   ERLOG      ;IF DATA AND ERRORS HAVE NOT BEEN LOGGED THEN:
2559 015572 001126      BNE     50233$
2560 015574 105237 003466      INCB   ERLOG      ;SET LOG DONE FLAG.
2561 015600 013704 003426      MOV    CMDLG,R4   ;GET CURRENT CMD LOGGING CODE.
2562 015604 005704      TST    R4        ;IF THERE IS A CODE THEN:
2563 015606 001520      BEQ    50234$
2564 015610 162704 000002      SUB    #2,R4     ;ADJUST THE CODE FOR TABLE INDEX.
2565 015614 010502      MOV    R5,R2     ;R2 = ADR OF BYTE COUNT LSW.
2566 015616 066402 016052      ADD    BINC(R4),R2
2567 015622 062702 002626      ADD    #CNTBGN,R2
2568 015626 063712 003416      ADD    BRFCNT,(R2) ;ADD BRFCNT TO LSW.
2569 015632 023737 002360 003416      CMP    MSGPKT+MS.RFC,BRFCNT ;IF THE RFC IS LOWER OR THE SAME AS DRF THEN
2570 015640 101002      BHI    50235$
2571 015642 163712 002360      SUB    MSGPKT+MS.RFC,(R2) ;SUBTRACT RFC FROM EXPECTED BRFC.
2572
2573 015646      50235$:
2574 015646 010203      MOV    R2,R3     ;R3 = ADR OF 2ND WORD.
2575 015650 062703 000010      ADD    #10,R3
2576
2577 015654      50236$: ;WHILE (R2) GT #999. DO
2578 015654 021227 001747      CMP    (R2),#999.
2579 015660 003404      BLE    50237$
2580 015662 162712 001750      SUB    #1000.,(R2) ;UPDATE BYTE COUNT
2581 015666 005213      INC    (R3)      ;LET (R3) := (R3) + #1 ;2ND WORD.
2582
2583 015670 000771      BR     50236$
2584 015672      50237$:
2585 015672 010302      MOV    R3,R2     ;LET R2 := R3 + #10 ;R2 = ADR OF 3RD WORD.
2586 015674 062702 000010      ADD    #10,R2
2587 015700      50240$: ;WHILE (R3) GT #999. DO
2588 015700 021327 001747      CMP    (R3),#999.
2589 015704 003404      BLE    50241$
2590 015706 162713 001750      SUB    #1000.,(R3) ;UPDATE BYTE COUNT
2591 015712 005212      INC    (R2)      ;LET (R2) := (R2) + #1 ;3RD WORD.
2592
2593 015714 000771      BR     50240$
2594 015716      50241$:
2595 015716 010203      MOV    R2,R3     ;LET R3 := R2 + #10 ;R3 = ADR OF 4TH WORD.
2596 015720 062703 000010      ADD    #10,R3
2597 015724      50242$: ;WHILE (R2) GT #999. DO
2598 015724 021227 001747      CMP    (R2),#999.
2599 015730 003404      BLE    50243$
2600 015732 162712 001750      SUB    #1000.,(R2) ;UPDATE BYTE COUNT
2601 015736 005213      INC    (R3)      ;LET (R3) := (R3) + #1 ;4TH WORD.
2602
2603 015740 000771      BR     50242$
2604 015742      50243$:
2605 015742 105737 003467      TSTB   RWERR     ;IF R/W ERROR, UPDATE ERROR COUNT.

```

GLOBAL SUBROUTINES SECTION

```

2606 015746 001440      BEQ      50244$
2507 015750 010502      MOV      R5,R2          ;R2 = ADR OF COUNTER.
2508 015752 066402 016060  ADD      EINC(R4),R2
2609 015756 062702 002766  ADD      @WRREC,R2
2610 015762 105737 003470  TSTB    UNREC          ;IS THE ERROR UNRECOVERABLE?
2611 015766 001404      BEQ      50245$
2612 015770 062702 000010  ADD      @10,R2        ;YES, POINT TO NEXT COUNTER.
2613 015774 005212      INC      (R2)          ;UPDATE THE ERROR COUNTER
2614
2615 015776 000424      BR       50246$        ;ELSE - IF ERROR IS RECOVERABLE:
2616 016000 50245$:
2617 016000 005212      INC      (R2)          ;UPDATE THE ERROR COUNTER
2618 016002 105737 002213  TSTB    IREC          ;IF ERROR RECOVERY IS ENABLED:
2619 016006 001020      BNE     50247$
2620 016010 105737 003522  TSTB    DROPED        ;IF UNIT HAS NOT BEEN DROPPED:
2621 016014 001015      BNE     50250$
2622 016016 105737 002207  TSTB    ERCOVER
2623 016022 001412      BEQ     50250$
2624 016024      PRINTB @NURTY1,RETRYC ;PRINT # OF RETRIES TO RECOVER
      016024 013746 003460      MOV     RETRYC,(SP)
      016030 012746 005422      MOV     @NURTY1,-(SP)
      016034 012746 000002      MOV     @2,-(SP)
      016040 010600      MOV     SP,R0
      016042 104414      TRAP   C$PNTB
      016044 062706 000006      ADD     @6,SP
2625
2626 016050 50250$:
2627
2628 016050 50247$:
2629
2630
2631 016050 50246$:
2632
2633 016050 50244$:
2634
2635 016050 50234$:
2636
2637 016050 50233$:
2638 016050 000207      RTS     PC
2639
2640      ; INDEXES TO BYTE COUNTERS.
2641 016052 000000      BINC:  0              ;WRITE.
2642 016054 000040      40          ;READ REV.
2643 016056 000100      100         ;READ FWD.
2644      ; INDEXES TO READ/WRITE ERROR COUNTERS.
2645 016060 000000      EINC:  0              ;WRITE.
2646 016062 000020      20          ;READ REV.
2647 016064 000040      40          ;READ FWD.
2648
2649      ; IF A WRITE/VERIFY COMMAND IS ISSUED, CONTROL IS THEN
2650      ; TRANSFERRED TO THIS SUBROUTINE TO READ REVERSE, CHECK DATA,
2651      ; READ FORWARD, CHECK DATA, THEN CONTINUE TO NEXT COMMAND.
2652      ; INPUTS:
2653      ; OUTPUTS:
2654      ; REGISTERS:
2655      ; CALLS:          VFEXC.
2656

```

133

GLOBAL SUBROUTINES SECTION

```

2657 016066 105737 003516          VFYDAT:;TSTB  VFYFLG          ;IF DATA IS TO BE VERIFIED:
2658 016072 001426                BEQ 502511
2659 016074 013737 003420 003424  MOV CMDWRD,PCMDWD ;SAVE THE PREVIOUS COMMAND WORD.
2660 016102 012737 104401 003420  MOV #RDR,CMDWRD  ;COMMAND IS READ REV.
2661 016110 012737 000004 003426  MOV #4,CMDLG     ;SET UP CMD LOGGING INDEX.
2662 016116 004737 016152          JSR PC,VFEXC    ;GO READ ALL THE RECORDS REV.
2663 016122 013737 003420 003424  MOV CMDWRD,PCMDWD ;SAVE THE PREVIOUS COMMAND WORD.
2664 016130 012737 104001 003420  MOV #RDF,CMDWRD  ;COMMAND IS READ FWD.
2665 016136 012737 000006 003426  MOV #6,CMDLG     ;SET UP CMD LOGGING INDEX.
2666 016144 004737 016152          JSR PC,VFEXC    ;GO READ ALL RECORDS FWD.
2667
2668 016150          502511:
2669 016150 000207                RTS  PC          ;RETURN.
2670
2671
2672
2673
2674
2675          ; SUBROUTINE TO EXECUTE THE READ AND VERIFY, FORWARD OR REVERSE.
2676          ; INPUTS:
2677          ; OUTPUTS:
2678          ; REGISTERS: R2
2679          ; CALLS: CMDAC, FIRSTU, VFISU, NEXTU, CMHAE.
2680 016152 013737 003420 002330 002330 VFEXC:; MOV CMDWRD,CMDPKT ;COMMAND PACKET = READ REV OR FWD.
2681 016160 042737 004000 002330 002330 BIC #BRF.C,CMDPKT
2682 016166 105737 003520          TSTB SWBFLG     ;IF BYTES ARE TO BE SWAPPED:
2683 016172 001403                BEQ 502521
2684 016174 052737 010000 002330 002330 BIS #SWB.C,CMDPKT ;SET SWAB BIT IN CMD PACKET.
2685
2686 016202          502521:
2687 016202 013737 002330 003422 003422 MOV CMDPKT,CMDSAV ;SAVE COMMAND PACKET 1ST WORD.
2688 016210 013737 003410 002332 002332 MOV DATARD,CMDPKT+CP.ADL ;SAVE BUFFER START ADDRESS.
2689 016216 005037 003412          CLR NCNT       ;CLEAR NUMBER OF OPERATIONS.
2690
2691 016222          502531: ;WHILE NCNT LT NCNT1 DO ;WHILE THERE ARE RECORDS REMAINING:
2692 016222 023737 003412 003414 003414 CMP NCNT,NCNT1
2693 016230 002062                RGE 502541
2694 016232 004737 011064          JSR PC,CMDAC   ;STORE CMD ASCII IN ERROR MSG.
2695 016236 004737 017102          JSR PC,FIRSTU ;SET UP FOR FIRST UNIT.
2696 016242          502551: ;WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE DEVICES REMAINING:
2697 016242 026527 002604 177777 177777 CMP DEVTBL(R5),#END
2698 016250 001442                BEQ 502561
2699 016252 032737 000400 003420 003420 BIT #MOD.CO,CMDWRD ;IF CMD IS REVERSE THEN:
2700 016260 001421                BEQ 502571
2701 016262 032765 000002 003502 003502 BIT #X0.BOT,EOTFLG(R5) ;IF NOT AT BOT
2702 016270 001014                BNE 502601
2703 016272 032765 000001 003502 003502 BIT #X0.EOT,EOTFLG(R5) ;BUT IF AT EOT
2704 016300 001406                BEQ 502611
2705 016302 105737 003524          TSTB ALLUNIT   ;AND ALL OTHERS AT EOT
2706 016306 001402                BEQ 502621
2707 016310 004737 016400          JSR PC,VFISU   ;THEN READ VERIFY
2708
2709 016314          502621: ;IF NOT ALL AT EOT, FREEZE UNIT(S)
2710
2711 016314 000402                BR 502631 ;IF NOT AT BOT AND
2712 016316          502611:
2713 016316 004737 016400          JSR PC,VFISU   ;NOT AT EOT, READ VFY

```

GLOBAL SUBROUTINES SECTION

```

2714
2715 016322          50263$:
2716
2717 016322          50260$:
2718 016322 000412   BR      50264$           ;ELSE IF CMD IS NOT REVERSE:
2719 016324          50257$:
2720 016324 032765 000001 003502   BIT    @X0.EOT,EOTFLG(R5)
2721 016332 001404          BEQ    50265$
2722 016334 032737 000001 003420   BIT    @CMD(C),CMDWRD
2723 016342 001002          BNE    50266$
2724 016344          50265$:
2725
2726 016344 004737 016400   JSR    PC,VFISU           ;IF NOT AT EOT OR NOT A MOTION CMD THEN:
2727                                     ;ISSUE CMD, CHECK STATUS AND DATA.
2728 016350          50266$:
2729
2730 016350          50264$:
2731 016350 004737 017150   JSR    PC,NEXTU           ;GO FIND THE NEXT UNIT.
2732
2733 016354 000732          BR      50255$
2734 016356          50256$:
2735 016356 004737 017500   JSR    PC,CKHAE           ;CHECK FOR HALT AFTER EACH CMD.
2736 016362 005237 003412   INC    NCNT              ;UPDATE THE RECORD COUNT.
2737 016366 013737 003420 003424   MOV    CMDWRD,PCMDWD     ;SAVE PREVIOUS COMMAND WORD.
2738
2739 016374 000712          BR      50253$
2740 016376          50254$:
2741 016376 000207          RTS    PC              ;RETURN.
2742
2743 ; SUBROUTINE TO ISSUE COMMAND, AWAIT INTERRUPT,
2744 ; CHECK STATUS, CHECK DATA.
2745 ; INPUTS:
2746 ; OUTPUTS:
2747 ; REGISTERS:      R2
2748 ; CALLS:          EXECUTE, GOWAIT, CKDATA.
2749
2750 016400 013702 003410   VFISU:: MOV    DATARD,R2           ;INIT READ BUFFER POINTER.
2751 016404 062702 000010   ADD    @B.,R2
2752 016410          50267$: ;WHILE R2 NE DATARD DO           ;UNTIL 8 BYTES HAVE BEEN SET,
2753 016410 020237 003410   CMP    R2,DATARD
2754 016414 001403          BEQ    50270$
2755 016416 012742 177777   MOV    @-1,-(R2)         ;INIT READ BUFFER.
2756
2757 016422 000772          BR      50267$
2758 016424          50270$:
2759 016424 004737 012054   JSR    PC,EXECUTE        ;GO EXECUTE THE COMMAND.
2760 016430 105737 003522   TSTB  DROPED             ;IF UNIT HAS NOT BEEN DROPPED THEN:
2761 016434 001002          BNE    50271$
2762 016436 004737 012364   JSR    PC,GOWAIT        ;GO WAIT FOR DONE BIT.
2763
2764 016442          50271$:
2765 016442 105737 003522   TSTB  DROPED             ;IF UNIT HAS NOT BEEN DROPPED THEN:
2766 016446 001006          BNE    50272$
2767 016450 032765 000002 003502   BIT    @X0.BOT,EOTFLG(R5) ;WHEN NOT REVERSED INTO BOT, THEN
2768 016456 001002          BNE    50273$
2769 016460 004737 016466   JSR    PC,CKDATA        ;GO VERIFY DATA.
2770

```


GLOBAL SUBROUTINES SECTION

```

2809 016636          50300$:
2810                ;REPEAT
2811 016636          50301$:
2812 016636 020237 017076      CMP      R2,CKDCNT      ;REPEAT UNTIL ALL DATA IS COMPARED:
2813 016642 001011          BNE      50302$      ;IF THIS IS THE LAST BYTE THEN:
2814 016644 105737 003520      TSTB   SWBFLG          ;IF BYTE SWAPPING IS ENABLED THEN:
2815 016650 001406          BEQ      50303$
2816 016652 032737 000001 017076  BIT     @BIT00,CKDCNT  ;IF RECORD LENGTH IS ODD THEN:
2817 016660 001002          BNE      50304$
2818 016662 105723          TSTB   (R3),          ;LAST BYTE WILL BE IN
2819 016664 105724          TSTB   (R4),          ;THE UPPER BYTE.
2820
2821 016666          50304$:
2822
2823 016666          50303$:
2824
2825 016666          50302$:
2826 016666 121314          CMPB   (R3),(R4)      ;ARE THEY EQUAL.
2827 016670 001452          BEQ      3$          ;BR IF SO.
2828 016672 005737 017100      TST   CKDFF          ;1 ST TIME THRU?
2829 016676 001010          BNE      2$          ;BR IF NOT.
2830 016700 005265 003346      INC   VFYCNT(R5)    ;INC THE VERIFY ERROR COUNTER.
2831 016704 005265 003356      INC   HRDCNT(R5)    ;INC THE HARD ERROR COUNT.
2832 016710          ERRHRD 17,WTVERM,DTAERM ;REPORT WRITE/VERIFY ERROR.
                TRAP   C$ERRHRD
                .WORD  17
                .WORD  WTVERM
                .WORD  DTAERM
2833 016720 005237 017100      2$:   INC   CKDFF;LET CKDFF := CKDFF + #1 ;INCREMENT # OF BYTES IN ERROR.
2834 016724 111437 003436      MOVB  (R4),TIME1    ;SAVE WAS DATA FOR TYP0UT.
2835 016730 042737 177400 003436  BIC   @177400,TIME1 ;CLEAR GARBAGE.
2836 016736 111337 003440      MOVB  (R3),TIME2    ;SAVE SHOULD BE DATA FOR TYP0UT.
2837 016742 042737 177400 003440  BIC   @177400,TIME2 ;CLEAR GARBAGE.
2838 016750 023727 017100 000013  CMP   CKDFF,#11.    ;IF ERROR BYTE COUNT IS LESS THAN 11:
2839 016756 002017          BGE   50305$
2840 016760          PRINTY @DTAER2,R2,<B,TIME1>,<B,TIME2>;PRINT ACTUAL & EXPECTED DATA
                CLR   -(SP)
                BISB  TIME2,(SP)
                CLR   -(SP)
                BISB  TIME1,(SP)
                MOV   R2,-(SP)
                MOV   @DTAER2,-(SP)
                MOV   #4,-(SP)
                MOV   SP,R0
                TRAP  C$PNTX
                ADD   #12,SP
2841 017016          50305$:
2842
2843 017016 105723          3$:   TSTB   (R3),          ;UPDATE WRITE BUFFER ADDRESS.
2844 017020 105724          TSTB   (R4),          ;UPDATE READ BUFFER ADDRESS.
2845 017022 105722          TSTB   (R2),          ;UPDATE BYTE COUNTER.
2846 017024 020237 017076      CMP   R2,CKDCNT    ;END OF DATA COMPARE REPEAT LOOP.
2847 017030 003702          BLE   50301$
2848 017032 005237 017076      INC   CKDCNT
2849 017036 005737 017100      TST   CKDFF
2850 017042 001414          BEQ   50306$
2851 017044          PRINTB @DTAER3,CKDFF,CKDCNT ;PRINT # OF BYTES IN ERROR.

```

GLOBAL SUBROUTINES SECTION

```

017044 013746 017076
017050 013746 017100
017054 012746 005275
017060 012746 000003
017064 010600
017066 104414
017070 062706 000010
2852
2853 017074 50306$:
2854
2855 017074 50277$:
2856
2857 017074 50275$:
2858 017074 000207 RTS PC ;OTHERWISE, RETURN.
2859
2860 017076 000000 CKDCNT: .WORD 0 ;# OF BYTES TO BE VERIFIED -1.
2861 017100 000000 CKDFF: .WORD 0 ;# OF BYTES IN ERROR COUNTER.
2862
2863 ; SUBROUTINE TO FIND THE FIRST DEVICE IN THE TEST SEQUENCE.
2864 ; INPUTS:
2865 ; OUTPUTS:
2866 ; REGISTERS:
2867 ; CALLS:
2868
2869 017102 105037 003522 FIRSTU:: CLRB DROPED ;CLR UNIT DROPPED FLAG
2870 017106 005005 CLR R5 ;CLR DEVICE POINTER.
2871 017110 026527 002604 177774 50307$: CMP DEVTBL(R5),#NINUSE ;WHILE DEVICES ARE NOT IN USE:
2872 017116 001003 BNE 50310$
2873 017120 062705 000002 ADD #2,R5 ;LET R5 := R5 + #2 ;POINT TO NEXT DEVICE.
2874 017124 000771 BR 50307$
2875 017126 50310$:
2876 017126 026527 002604 177777 CMP DEVTBL(R5),#END ;IF ALL UNITS HAVE BEEN DROPPED THEN:
2877 017134 001001 BNE 50311$
2878 017136 DOCLN ;DO CLEAN CODE AND TERMINATE PASS.
017136 104444 TRAP C$DCLN
2879
2880 017140 50311$:
2881 017140 016537 002604 002074 MOV DEVTBL(R5),L$LUN ;SET UNIT # IN "HEADER" FOR ERROR REPORT
2882 017146 000207 RTS PC ;RETURN WITH 1ST DEVICE IN R5.
2883
2884 ; SUBROUTINE TO FIND THE NEXT UNIT IN THE TEST CYCLE.
2885 ; INPUTS:
2886 ; OUTPUTS:
2887 ; REGISTERS:
2888 ; CALLS:
2889
2890
2891 017150 105037 003522 NEXTU:: CLRB DROPED ;CLR UNIT DROPPED FLAG
2892 ;REPEAT ;REPEAT UNTIL THE NEXT DEVICE IS FOUND.
2893 017154 50312$:
2894 017154 062705 000002 ADD #2,R5 ;UPDATE DEVICE TABLE POINTER.
2895 017160 026527 002604 177774 CMP DEVTBL(R5),#NINUSE ;UNTIL DEVTBL(R5) NE #NINUSE
2896 017166 001772 BEQ 50312$
2897 017170 016537 002604 002074 MOV DEVTBL(R5),L$LUN ;SET UNIT # IN "HEADER" FOR ERROR REPORT
2898 017176 000207 RTS PC ;RETURN.
2899
2900

```

GLOBAL SUBROUTINES SECTION

```

2901      ;      SUBROUTINE TO DROP A DEVICE FROM THE TEST SEQUENCE.
2902      ;      INPUTS:
2903      ;      OUTPUTS:
2904      ;      REGISTERS:
2905      ;      CALLS:          MOVMSG, PRXST, LOG
2906
2907 017200 005265 003366      DROPU:: INC      FTLCNT(R5)          ;INCREMENT THE FATAL ERROR COUNT.
2908 017204 013704 002370      MOV      MSGPKT+MS.XS3,R4      ;GET UDIAG ERROR CODE FROM XSTAT3.
2909 017210 042704 000377      BIC      #377,R4
2910 017214 016503 002544      MOV      MSGPKA(R5),R3        ;ADR OF THIS UNIT'S MSG PACKET.
2911 017220 005002      CLR      R2                    ;LET R2 := #0 ;CLR COUNTER.
2912 017222      50313$: ;WHILE R2 NE #MSGCNT DO          ;WHILE THERE ARE MORE LOCATIONS:
2913 017222 020227 000020      CMP      R2,#MSGCNT
2914 017226 001405      BEQ      50314$
2915 017230 012723 177777      MOV      #-1,(R3)+           ;INIT THE MSC PACKET WITH ALL 1'S
2916 017234 062702 000002      ADD      #2,R2                ;LET R2 := R2 + #2 ;UPDATE COUNTER.
2917
2918 017240 000770      BR      50313$
2919 017242      50314$:
2920 017242 012775 002340 002514      MOV      #GSCPK,@TSDB(R5)      ;INITIATE A GET STATUS COMMAND.
2921 017250 004737 012700      JSR      PC,WSSR              ;WAIT A WHILE FOR SSR=1
2922 017254 004737 012734      JSR      PC,MOVMSG            ;MOVE MSG PACKET TO COMMON AREA.
2923 017260 020427 157400      CMP      R4,#X3.RNY          ;IF WE HAVE A CAPSTAN RUNAWAY THEN:
2924 017264 001005      BNE      50315$
2925 017266      ERRDF 16,RNYM,STAERM      ;REPORT CAPSTAN RUNAWAY WITH TACH CNT.
                                     TRAP      C#ERDF
                                     .WORD    16
                                     .WORD    RNYM
                                     .WORD    STAERM
2926      ;ELSE-IF NOT A RUNAWAY:
2927 017276 000402      BR      50316$
2928 017300      50315$:
2929 017300 004737 017412      JSR      PC,PRXST             ;PRINT EXTENDED STATUS REGISTERS.
2930
2931 017304      50316$:
2932 017304 105737 003465      TSTB     RECLOG               ;IF THE RECORD HAS BEEN LOGGED THEN:
2933 017310 001404      BEQ      50317$
2934 017312 105237 003522      INCB     DROPED              ;SET UNIT DROPPED FLAG.
2935 017316 004737 015566      JSR      PC,LOG              ;LOG DATA BYTES + RD/WR ERRORS.
2936
2937 017322      50317$:
2938 017322      DORPT              ;PRINT PERFORMANCE REPORT
                                     TRAP      C#DRPT
2939 017324 104424 003326      DROPUA: TST      PASCNT(R5)      ;IF PASCNT(R5) NE #0 THEN
2940 017330 001402      BEQ      50320$
2941 017332 005365 003326      DEC      PASCNT(R5)          ;LET PASCNT(R5) := PASCNT(R5) - #1
2942
2943 017336      50320$:
2944 017336 013737 003534 017410      MOV      TSNP,DROPN          ;SAVE # OF UNIT TO BE DROPPED.
2945 017344 013700 003534      MOV      TSNP,R0             ;R0=LOGICAL DEVICE NUMBER
2946 017350      DODU      R0        ;DROP THE UNIT
                                     TRAP      C#DODU
2947      ;EXEC BGNDU-ENDDU CODE IF IDU = 0
2948
2949 017352 026527 002604 177774      CMP      DEVTBL(R5),#NINUSE   ;IF UNIT NOT DROPPED
2950 017360 001410      BEQ      50321$
2951 017362 105737 002213      TSTB     IREC                ;IF RECOVERY IS ENABLED THEN:

```

GLOBAL SUBROUTINES SECTION

```

2952 017366 001005      BNE      50322$
2953 017370 000240      NOP
2954 017372 000240      NOP
2955 017374 000240      NOP
2956 017376 105237 003526  INCB     STAFLG      ;SET START FLAG TO ENABLE REWIND,
2957
2958 017402      50322$:
2959
2960 017402      50321$:
2961 017402 105237 003522  DRORTN: INCB     DROPEL      ;SET UNIT DROPPED FLAG.
2962 017406 000207      RTS      PC          ;RETURN.
2963
2964 017410 000000      DROPN:  .WORD    0          ;# OF UNIT TO BE DROPPED
2965
2966      :      SUBROUTINE TO PRINT EXTENDED STATUS REGISTERS.
2967      :      INPUTS:
2968      :      OUTPUTS:
2969      :      REGISTERS:
2970      :      CALLS:
2971
2972 017412      PRXST:: PRINTX #GETSTM
2973 017412 012746 005507      MOV      #GETSTM, -(SP)
2974 017416 012746 000001      MOV      #1, -(SP)
2975 017422 010600      MOV      SP, R0
2976 017424 104415      TRAP    C$PNTX
2977 017426 062706 000004      ADD     #4, SP
4 2973 017432      PRINTX #STAERS,MSGPKT+MS.XS0,MSGPKT+MS.XS1,MSGPKT+MS.XS2,MSGPKT+MS.XS3,MSGPKT+MS.XS
2974 017432 013746 002372      MOV     MSGPKT+MS.XS4, -(SP)
2975 017436 013746 002370      MOV     MSGPKT+MS.XS3, -(SP)
2976 017442 013746 002366      MOV     MSGPKT+MS.XS2, -(SP)
2977 017446 013746 002364      MOV     MSGPKT+MS.XS1, -(SP)
2978 017452 013746 002362      MOV     MSGPKT+MS.XS0, -(SP)
2979 017456 012746 006753      MOV     #STAERS, -(SP)
2980 017462 012746 000006      MOV     #6, -(SP)
2981 017466 010600      MOV     SP, R0
2982 017470 104415      TRAP    C$PNTX
2983 017472 062706 000016      ADD     #16, SP
2974 017476 000207      RTS PC
2975
2976      :      SUBROUTINE TO HALT AFTER EACH COMMAND.
2977      :      INPUTS:
2978      :      OUTPUTS:
2979      :      REGISTERS:      R3, R4
2980      :      CALLS:
2981
2982 017500 105737 002206      CKHAE:: TSTB     HAE;IFB HAE NE #0 THEN      ;IF HALT FLAG IS SET:
2983 017504 001430      BEQ     50323$
2984 017506 105737 003531      TSTB     MISCFG      ;IFB MISCFG EQ #0 THEN      ;
2985 017512 001023      BNE     50324$
2986 017514      MANUAL      ;IS MANUAL INTERVENTION ALLOWED?
2987 017514 104450      BNCOMPLETE CKHRTN      ;BR IF NOT.      TRAP    C$MANI
2988 017516 103023      MOV     CMDWRD, R4      ;LET R4 := CMDWRD      BCC     CKHRTN
2989 017520 013704 003420      JSR    PC, GCMOA      ;FETCH ADR OF CMD ASCII.      ;COMMAND WORD.
2990 017524 004737 011136      MOVB   (R3)+, HALTM      ;MOVE CMD ASCII
2991 017530 112337 004306      MOVB   (R3)+, HALTM+1    ;LET HALTM+1 := (R3)+
2991 017534 112337 004307

```

GLOBAL SUBROUTINES SECTION

```

2992 017540 111337 004310      MOVB    (R3),HALTM+2      ;INTO MESSAGE.
2993 017544      GMANIL HALTM,TIME1.1,YES ;HALT - WAIT FOR AN OEPRTOR INPUT.
      017544 104443      TRAP    C$GMAN
      017546 000404      BR      10000$
      017550 003436      .WORD   TIME1
      017552 000130      .WORD   T$CODE
      017554 004306      .WORD   HALTM
      017556 000001      .WORD   1
      017560      10000$:
2994 017560      10000$:
2995
2996 017560 000402      BR      50325$
2997 017562      50324$:
2998 017562 105037 003531      CLRB    MISCFG          ;LET MISCFG :B= #0
2999
3000 017566      50325$:
3001
3002 017566      50323$:
3003 017566 000207      CKHRTN: RTS    PC          ;RETURN
3004      .EVEN
3005
3006 017570      ENDMOD
3007
3008
3009      .TITLE MISCELLANEOUS SECTIONS
3010      .SBTTL REPORT CODING SECTION
3011
3012
3013      ;++
3014      ; THE REPORT CODING SECTION CONTAINS THE
3015      ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
3016      ;--
3017
3018 017570      BGNRPT
      017570      L$RPT::
3019 017570 010537 003452      MOV     R5,R5SAVE      ;SAVE CURRENT DEVICE POINTER.
3020 017574 004737 017102      JSR    PC,FIRSTU      ;FIND THE FIRST UNIT.
3021 017600      50326$: ;WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
3022 017600 026527 002604 177777      CMP    DEVTBL(R5),#END
3023 017606 001562      BEQ    50327$
3024 017610      PRINTS    #RPT1A,DEVTBL(R5),PASCNT(R5),RECCNT(R5)
      017610 016546 003376      MOV    RECCNT(R5),-(SP)
      017614 016546 003326      MOV    PASCNT(R5),-(SP)
      017620 016546 002604      MOV    DEVTBL(R5),-(SP)
      017624 012746 020432      MOV    #RPT1A, -(SP)
      017630 012746 000004      MOV    #4, -(SP)
      017634 010600      MOV    SP,R0
      017636 104416      TRAP   C$PNTS
      017640 062706 000012      ADD    #12,SP
3025 017644      PRINTS    #RPT1B,WRBC+30(R5),WRBC+20(R5),WRBC+10(R5),WRBC(R5)
      017644 016546 002626      MOV    WRBC(R5),-(SP)
      017650 016546 002636      MOV    WRBC+10(R5),-(SP)
      017654 016546 002646      MOV    WRBC+20(R5),-(SP)
      017660 016546 002656      MOV    WRBC+30(R5),-(SP)
      017664 012746 020507      MOV    #RPT1B, -(SP)
      017670 012746 000005      MOV    #5, -(SP)
      017674 010600      MOV    SP,R0

```

REPORT CODING SECTION

017676	104416				TRAP	C\$PNTS
017700	062706	000014			ADD	#14,SP
3026	017704		PRINTS	#RPT1C,RRBC+30(R5),RRBC+20(R5),RRBC+10(R5),RRBC(R5)	MOV	RRBC(R5),-(SP)
	017704	016546			MOV	RRBC+10(R5),-(SP)
	017710	016546			MOV	RRBC+20(R5),-(SP)
	017714	016546			MOV	RRBC+30(R5),-(SP)
	017720	016546			MOV	#RPT1C,-(SP)
	017724	012746			MOV	#5,-(SP)
	017730	012746			MOV	SP,R0
	017734	010600			TRAP	C\$PNTS
	017736	104416			ADD	#14,SP
3027	017740	062706	000014		MOV	RFBC(R5),-(SP)
	017744		PRINTS	#RPT1D,RFBC+30(R5),RFBC+20(R5),RFBC+10(R5),RFBC(R5)	MOV	RFBC+10(R5),-(SP)
	017744	016546			MOV	RFBC+20(R5),-(SP)
	017750	016546			MOV	RFBC+30(R5),-(SP)
	017754	016546			MOV	#RPT1D,-(SP)
	017760	016546			MOV	#5,-(SP)
	017764	012745			MOV	SP,R0
	017770	012746			TRAP	C\$PNTS
	017774	010600			ADD	#14,SP
	017776	104416			MOV	RFREC(R5),-(SP)
3028	020000	062706	000014		MOV	RRREC(R5),-(SP)
	020004		PRINTS	#RPT1F,WRREC(R5),RRREC(R5),RFREC(R5)	MOV	WRREC(R5),-(SP)
	020004	016546			MOV	#RPT1F,-(SP)
	020010	016546			MOV	#4,-(SP)
	020014	016546			MOV	SP,R0
	020020	012746			TRAP	C\$PNTS
	020024	012746			ADD	#12,SP
	020024	012746			MOV	RFUNR(R5),-(SP)
	020030	010600			MOV	RRUNR(R5),-(SP)
	020032	104416			MOV	WRUNR(R5),-(SP)
	020034	062706	000012		MOV	#RPT1G,-(SP)
3029	020040		PRINTS	#RPT1G,WRUNR(R5),RRUNR(R5),RFUNR(R5)	MOV	#4,-(SP)
	020040	016546			MOV	SP,R0
	020044	016546			TRAP	C\$PNTS
	020050	016546			ADD	#12,SP
	020054	012746			MOV	VFYCNT(R5),-(SP)
	020060	012746			MOV	FTLCNT(R5),-(SP)
	020064	010600			MOV	HRDCNT(R5),-(SP)
	020066	104416			MOV	SCCNT(R5),-(SP)
	020070	062706	000012		MOV	#RPT1I,-(SP)
3030	020074	105737	002210	TSTB BADTSW ;IFB BADTSW NE #0 THEN	MOV	#5,-(SP)
3031	020100	001402		BEQ 50330\$	MOV	SP,R0
3032	020102	004737	020164	JSR PC,BTRPT ;GO PRINT BAD TAPE SPOTS WHEN ENABLED	TRAP	C\$PNTS
3033					ADD	#12,SP
3034	020106		50330\$:			
3035	020106		PRINTS	#RPT1I,SCCNT(R5),HRDCNT(R5),FTLCNT(R5),VFYCNT(R5)	MOV	VFYCNT(R5),-(SP)
	020106	016546			MOV	FTLCNT(R5),-(SP)
	020112	016546			MOV	HRDCNT(R5),-(SP)
	020116	016546			MOV	SCCNT(R5),-(SP)
	020122	016546			MOV	#RPT1I,-(SP)
	020126	012746			MOV	#5,-(SP)
	020132	012746			MOV	SP,R0
	020136	010600			TRAP	C\$PNTS
	020140	104416			ADD	#14,SP
	020142	062706	000014			
3036	020146	004737	01150	JSR PC,NEXTU ;FIND THE NEXT UNIT.		
3037						

REPORT CODING SECTION

```

3038 020152 000612          BR      50326$
3039 020154          50327$:
3040 020154 013705 003452  MOV     R5SAVE,R5          ;RESTORE CURRENT DEVICE POINTER.
3041 020160          EXIT    RPT
      020160 000167          .WORD   J$JMP
      020162 001130          .WORD   L10010-2-.
3042
3043          ;      SUBR TO PRINT BAD TAPES SPOTS DURING THE REPORT PRINTS
3044          ;      WRITE RETRIES: CUMULATIVE COUNT
3045          ;      BAD TAPE SPOTS: COUNT PER TAPE PASS ONLY, NOT CUMULATIVE.
3046          ;      COUNT OF RECOVERABLE WRITE ERRORS EXCLUDES BAD TAPE SPOTS.
3047
3048 020164          BTRPT:: PRINTS #RPT1E,WRTYCT(R5)          ;PRINT GLOBAL WRITE RETRY COUNT
      020164 016546 003316          MOV     WRTYCT(R5),-(SP)
      020170 012746 021057          MOV     #RPT1E, -(SP)
      020174 012746 000002          MOV     #2, -(SP)
      020200 010600          MOV     SP,R0
      020202 104416          TRAP   C$PNTS
      020204 062706 000006          ADD     #6,SP
3049 020210 016537 002616 003512  MOV     BTADDR(R5),BTPT          ;BTPT IS BOTH THE BAD TAPE SPOT COUNTER
3050 020216 017703 163270          MOV     #BTPT,R3          ;AND THE LOGGING INDEX
3051 020222 006203          ASR     R3
3052 020224          PRINTS #RPT1J,R3          ;PRINT # OF BAD TAPE SPOTS
      020224 010346          MOV     R3, -(SP)
      020226 012746 021107          MOV     #RPT1J, -(SP)
      020232 012746 000002          MOV     #2, -(SP)
      020236 010600          MOV     SP,R0
      020240 104416          TRAP   C$PNTS
      020242 062706 000006          ADD     #6,SP
3053 020246 005703          TST     R3          ;PRINT RECORD # IF BAD SPOTS DETECTED
3054 020250 001457          BEQ     50331$
3055 020252 020327 000024          CMP     R3,#20.          ;IF R3 HI #20. THEN
3056 020256 101402          BLUS   50332$
3057 020258 012703 000024          MOV     #20.,R3          ;20 BAD SPOTS IS THE LIMIT
3058
3059 020264          50332$:
3060 020264          PRINTS #CRLFSP
      020264 012746 005744          MOV     #CRLFSP, -(SP)
      020270 012746 000001          MOV     #1, -(SP)
      020274 010600          MOV     SP,R0
      020276 104416          TRAP   C$PNTS
      020300 062706 000004          ADD     #4,SP
3061 020304 013704 003512          MOV     BTPT,R4          ;LET R4 := BTPT + #2 ;FETCH A BAD SPOT ID
3062 020310 062704 000002          ADD     #2,R4
3063 020314 005002          CLR     R2          ;R2 = PRINT COUNT PER LINE: 10 MAX
3064 020316          50333$: ;REPEAT
3065 020316          PRINTS #RPT1K,(R4)          ;PRINT A BAD SPOT ID
      020316 011446          MOV     (R4), -(SP)
      020320 012746 021174          MOV     #RPT1K, -(SP)
      020324 012746 000002          MOV     #2, -(SP)
      020330 010600          MOV     SP,R0
      020332 104416          TRAP   C$PNTS
      020334 062706 000006          ADD     #6,SP
      020340 005202          INC     R2          ;LET R2 := R2 + #1 ;COUNT PRINTS
      020342 062704 000002          ADD     #2,R4          ;LET R4 := R4 + #2 ;NEXT
3068 020346 020227 000012          CMP     R2,#10.          ;IF R2 EQ #10. THEN
3069 020352 001014          BNE    50334$

```

REPORT CODING SECTION

```

3070 020354          PRINTS #CRLFSP          ;GO TO NEXT PRINT LINE PAST 10 PRINTS
      020354 012746 005744          MOV      #CRLFSP, -(SP)
      020360 012746 000001          MOV      #1, -(SP)
      020364 010600          MOV      SP, R0
      020366 104416          TRAP     C#PNTS
      020370 062706 000004          ADD      #4, SP
3071 020374 162703 000012          SUB      #10., R3          ;LET R3 := R3 - #10.          ;ADJUST BAD SPOT COUNT
3072 020400 162702 000012          SUB      #10., R2          ;LET R2 := R2 - #10.          ;ADJUST PRINT COUNT
3073
3074 020404          50334$:
3075 020404 020203          CNP     R2, R3          ;UNTIL R2 EQ R3          ;LIMIT: # OF BAD SPOTS
3076 020406 001343          BNE     50333$
3077
3078 020410          50331$:
3079 020410          PRINTS #CRLF          ;
      020410 012746 005741          MOV      #CRLF, -(SP)
      020414 012746 000001          MOV      #1, -(SP)
      020420 010600          MOV      SP, R0
      020422 104416          TRAP     C#PNTS
      020424 062706 000004          ADD      #4, SP
3080 020430 000207          RTS PC
3081
3082          .NLIST BEX
3083 020432          045 116 045 RPT1A: .ASCIZ /#N#N#AUNIT #D1#S3#APASS:#D5#S3#ARECORD:#D5#N/
3084 020507          045 101 102 RPT1B: .ASCIZ /#ABYTES WRITTEN #D3#A,#Z3#A,#Z3#A,#Z3#N/
3085 020560          045 101 102 RPT1C: .ASCIZ /#ABYTES READ REV #D3#A,#Z3#A,#Z3#A,#Z3#N/
3086 020631          045 101 102 RPT1D: .ASCIZ /#ABYTES READ FWD #D3#A,#Z3#A,#Z3#A,#Z3#N/
3087 020701          045 123 062 .ASCIZ /#S23#AWRT#S4#ARDR#S4#ARDF#N/
3088 020735          045 101 122 RPT1F: .ASCIZ /#ARECOVERABLE ERRORS #D5#S2#D5#S2#D5#N/
3089 021006          045 101 125 RPT1G: .ASCIZ /#AUNRECOVERABLE ERRORS #D5#S2#D5#S2#D5#N/
3090 021057          045 101 127 RPT1E: .ASCIZ /#AWRITE RETRIES#S8#D5#N/
3091 021107          045 116 045 RPT1J: .ASCIZ /#N#D2#A BAD SPOTS THIS TAPE PASS PRECEDING RECORD #:/
3092 021174          045 104 065 RPT1K: .ASCIZ /#D5#S1/
3093 021203          045 101 123 RPT1I: .ASCIZ /#ASPEC COND#S3#AHARD#S3#AF ATAL#S3#ACOMPARE#N"
3094 021257          045 123 063 .ASCIZ /#S3#D5#S3#D5#S3#D5#S3#D5#N#N/
3095          .LIST BEX
3096          .EVEN
3097
3098 021314          ENDRPT
      021314          L10010:
      021314 104425          TRAP     C#RPT
3099
3100          .SBTTL LOAD DEVICE PROTECTION TABLE
3101
3102          ;++
3103          ;TABLE FOR SUPERVISOR TO IDENTIFY THE P-TBL FOR THE LOAD DEV
3104          ;THE SUPERVISOR USES THE TBL TO WARN THE OPERATOR WHEN HE TRIES TO TEST THE LOAD DEV
3105          ;--
3106
3107 021316          BGNPROT
      021316          L$PROT::
3108
3109 021316 000000          .WORD 0          ;P-TBL OFFSET OF TSOB
3110 021320 177777          .WORD -1          ;P-TBL OFFSET OF MASS BUS UNIT #: -1 = NOT A MASS BUS DE
3111 021322 177777          .WORD -1          ;P-TBL OFFSET OF DRIVE #: -1 = NONE, THREE DRIVES PER CONTRO
LLER
3112 021324          ENDRPT
3113

```

INITIALIZE SECTION

```

3114          .SBTTL  INITIALIZE SECTION
3115
3116          ;--
3117          ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
3118          ; AT THE BEGINNING OF EACH PASS.
3119          ;--
3120
3121          BGNINIT
3122          L$INIT::
3123          SETVEC  #140,#170000,#340          ;ODT ROM ADDRESS          ;JB REV A-0
          021324 012746 000340          MOV          #340,-(SP)
          021330 012746 170000          MOV          #170000,-(SP)
          021334 012746 000140          MOV          #140,-(SP)
          021340 012746 000003          MOV          #3,-(SP)
          021344 104437          TRAP        C$SVEC
          021346 062706 000010          ADD          #10,SP
3124
3125 021352 032727 000003 002330 INIT10: BIT      #BIT0!BIT1,#CMDPKT  ;IF CMD PACKET IS NOT ON MODULO 4 BOUNDRY:
3126 021360 001421          BEQ        50335$
3127 021362          ERRSF 1,CMDPKM          ;PRINT ERROR MSG,
          021362 104454          TRAP        C$ERSF
          021364 000001          .WORD      1
          021366 004346          .WORD      CMDPKM
          021370 000000          .WORD      0
3128 021372          DELAY 200.          ;GO TO SUPERVISOR, WAIT 2 SECONDS.
          021372 012727 000310          MOV          #200.,(PC)+
          021376 000000          .WORD      0
          021400 013727 002116          MOV          L$DLY,(PC)+
          021404 000000          .WORD      0
          021406 005367 177772          DEC          -6(PC)
          021412 001375          BNE          .-4
          021414 005367 177756          DEC          -22(PC)
          021420 001367          BNE          .-20
3129 021422 000753          BR         INIT10          ;
3130
3131          50335$:
3132
3133 021424 105737 002204          TSTB       CLRFLG          ;IF CLR COUNTERS FLAG SET:
3134 021430 001413          BEQ        50336$
3135 021432 105037 002204          CLRB      CLRFLG          ;INIT CLR FLAG.
3136 021436 005002          CLR       R2          ;LET R2 := #0
3137 021440          50337$: ;WHILE R2 NE #CNTLEN DO
          021440 020227 000550          CMP        R2,#CNTLEN
          021444 001405          BEQ        50340$
          021446 005062 002626          CLR       WRBC(R2)          ;CLR ALL STATISTICAL COUNTERS.
          021452 062702 000002          ADD        #2,R2          ;LET R2 := R2 + #2
3142
3143 021456 000770          BR         50337$
3144 021460          50340$:
3145
3146          50336$:
3147
3148 021460 105737 002205          TSTB       RRANV          ;IF RESET RANDOM VARIABLE FLAG IS SET THEN:
3149 021464 001406          BEQ        50341$
3150 021466 012737 153624 003432          MOV        #RANBC,RANB          ;RESET RANDOM BASE #.
3151

```

INITIALIZE SECTION

```

3152 021474 012737 032561 003434      MOV      #RANSC,RANS          ;RESET RANDOM SAVE LOCATION.
3153
3154 021502      50341$: READEF #EF.START          ;READ START COMMAND EVENT FLAG.
3155 021502      012700 000040      MOV      #EF.START,R0
021506 104447      TRAP    C$REFG
3156 021510      BNCOMPLETE INIT15          ;BRANCH IF NOT STARTING.
021510 103057      BCC     INIT15
3157 021512 105237 003526      INCB    STAFLG          ;SET START COMMAND FLAG.
3158 021516 012705 000006      MOV     #6,R5 ;LET R5 := #6
3159 021522      50342$: ;REPEAT          ;INITIATE UNIT NUMBER TABLE
3160 021522 012765 177774 002604      MOV     #NINUSE,DEVTBL(R5) ;BY STORING NOT IN USE IN EACH LOCATION.
3161 021530 162705 000002      SUB     #2,R5 ;LET R5 := R5 - #2
3162 021534 005705      TST    R5 ;UNTIL R5 EQ #0
3163 021536 001371      BNE    50342$
3164 021540 022737 000001 002012      CMP     #1,L$UNIT          ;ONLY ONE UNIT ALLOWED
3165 021546 001425      BEQ    5034$              ;OK
3166 021550      PRINTF #AUDRUN          ;TELL THE MAN
021550 012746 005146      MOV     #AUDRUN,-(SP)
021554 012746 000001      MOV     #1,-(SP)
021560 010600      MOV     SP,R0
021562 104417      TRAP    C$PNTF
021564 052706 000004      ADD     #4,SP
3167 021570      DELAY  25                ;WAIT
021570 012727 000025      MOV     #25,(PC)+
021574 000000      .WORD  0
021576 013727 002116      MOV     L$DLY,(PC)+
021602 000000      .WORD  0
021604 005367 177772      DEC     -6(PC)
021610 001375      BNE     -4
021612 005367 177756      DEC     -22(PC)
021616 001367      BNE     -20
3168 021620      DOCLN                    ;ABORT
021620 104444      TRAP    C$DCLN
3169 021622 013705 002012      5034$: MOV     L$UNIT,R5 ;LET R5 := L$UNIT SHIFT 1
3170 021626 006305      ASL    R5
3171 021630      50343$: ;REPEAT          ;STORE ALL UNIT
3172 021630 162705 000002      SUB     #2,R5 ;LET R5 := R5 - #2 ;NUMBERS IN DEVTBL.
3173 021634 010565 002604      MOV     R5,DEVTBL(R5) ;LET DEVTBL(R5) := R5 SHIFT -1
3174 021640 006265 002604      ASR    DEVTBL(R5)
3175 021644 005705      TST    R5 ;UNTIL R5 EQ #0
3176 021646 001370      BNE    50343$
3177
3178 021650      INIT15: READEF #EF.PWR          ;HAS THERE BE A POWER FAILURE?
021650 012700 000034      MOV     #EF.PWR,R0
021654 104447      TRAP    C$REFG
3179 021656      BNCOMPLETE INIT16          ;BRANCH IF NOT.
021656 103004      BCC     INIT16
3180 021660 105237 003526      INCB    STAFLG          ;IF SO - SET THE START FLAG.
3181 021664 105237 003527      INCB    PWRFLG          ;IF SO - SET THE POWER FAIL FLAG.
3182
3183 021670      INIT16: RFLAGS OPFLAG          ;READ AND STORE FLAGS SET BY OPERATOR
021670 104421      TRAP    C$RFLA
021672 010037 003536      MOV     R0,OPFLAG
3184 021676 005003      CLR    R3 ;LET R3 := #0 ;CLEAR EVENT FLAG
3185 021700 105737 003527      TSTB   PWRFLG          ;IF POWER FAIL HAS NOT OCCURRED THEN:
3186 021704 001020      BNE    50344$

```

INITIALIZE SECTION

```

3187 021706          READEFF OFF,NEW          ;UPDATE PASS COUNT WHEN
      021706 012700 000035                    MOV      OFF,NEW,RO
      021712 104447                    TRAP    C$REFF
3188 021714 103014          BCC      50345$          ;SUPERVISOR IS IN NEW PASS
3189 021716 105737 003526          TSTB   STAF LG          ;AND DIAG WAS NEITHER STARTED
3190 021722 001010          BNE    50346$
3191 021724          READEFF OFF,RES          ;NOR
      021724 012700 000037                    MOV      OFF,RES,RO
      021730 104447                    TRAP    C$REFF
3192 021732 103402          BCC    50347$          ;IFCOND CC THEN ;RESTARTED
3193 021734 005103          COM    R3              ;LET R3 := COMP R3 ;DO IT
3194
3195 021736 000401          BR     50350$
3196 021740          50347$:
3197 021740 005203          INC    R3              ;SET 1ST PASS IF NEW PASS AND
3198                                ;RESTARTING
3199 021742          50350$:
3200
3201 021742 000401          PC    50351$
3202 021744          50346$:
3203 021744 005203          INC    R3              ;SET 1ST PASS IF NEW PASS AND
3204                                ;STARTING
3205 021746          50351$:
3206                                ;DO NOT UPDATE IT ON CONTINUE
3207 021746          50345$:
3208                                ;OR ON POWER FAIL
3209 021746          50344$:
3210 021746 004737 017102          JSR    PC,FIRSTU          ;INIT DEVICE POINTER,
3211 021752 005002          CLR    R2              ;LET R2 := #0          ;INIT DEVICE COUNTER,
3212 021754          50352$: ;WHILE DEVTBL(R5) NE #END DO
3213 021754 026527 002604 177777          CMP    DEVTBL(R5),#END
3214 021762 001456          BEQ    50353$
3215 021764 005202          INC    R2              ;LET R2 := R2 + #1
3216 021766 010500          MOV    R5,RO          ;LET RO := R5 SHIFT -1
3217 021770 006200          ASR    RO
3218 021772          GPHARD RO,RO          ;GET HARDWARE P TABLE FROM SUPER.
3219 021774 104442          TRAP    C$GPHARD
3220 021776 011065 002514          BCC    50354$          ;IFCOND CS THEN
3221 022002 012065 002524          MOV    (RO),TSDB(R5)   ;SAVE TSDB ADDRESS.
3222 022006 062765 000002 002524          MOV    (RO),TSSR(R5)   ;SAVE TSSR ADDRESS.
3223 022014 012065 002534          ADD    #2,TSSR(R5)
3224 022020 011065 003532          MOV    (RO),TSVCT(R5)  ;SAVE INTERRUPT VECTOR ADDRESS.
3225 022024 011037 003534          MOV    (RO),TSUNT(R5)  ;SAVE NUMBER OF DRIVE
3226 022030          SETVEC TSVCT(R5),TSSINT(R5),#INTPRI ;SAVE FOR PRINT OUT'S
      022030 012746 000300          MOV    #INTPRI,(SP)
      022034 016546 002554          MOV    TSSINT(R5),(SP)
      022040 016546 002534          MOV    TSVCT(R5),-(SP)
      022044 012746 000003          MOV    #3,-(SP)
      022050 104437          TRAP    C$SVEC
      022052 062706 000010          ADD    #10,SP
3227
3228 022056 005065 003472          CLR    INTFLG(R5)      ;SET UP INTERRUPT PROCESSING CONDITIONS.
3229 022062 005703          TST    R3              ;CLEAR INTERRUPT FLAGS.
3230 022064 001410          BEQ    50355$          ;ACTUAL PASSCOUNT UPDATE PER R3
3231 022066 005703          TST    R3
3232 022070 002003          BGE    50356$          ;IF R3 LT #0 THEN

```

INITIALIZE SECTION

3233	022072	005265	003326	INC	PASCNT(R5)	;(LET PASCNT(R5) := PASCNT(R5) + 01
3234						
3235	022076	000403		BR	503571	
3236	022100					
3237	022100	012765	000001 003326	MOV	01,PASCNT(R5)	;(LET PASCNT(R5) := 01
3238						
3239	022106				503571:	
3240						
3241	022106				503551:	
3242						
3243	022106				503541:	
3244	022106	005065	003376	CLR	RECCNT(R5)	;(CLEAR RECORD COUNT
3245	022112	004737	017150	JSR	PC,NEXTU	;(DO IT FOR ALL DEVICES.
3246						
3247	022116	000716		BR	503521	
3248	022120				503531:	
3249						
3250	022120	005702		TST	R2	;(IF THERE ARE NO UNITS:
3251	022122	001026		BNE	503601	
3252	022124			PRINTF	0AUDRPM	;(PRINT ALL UNITS DROPPED,
	022124	012746	005114			MOV 0AUDRPM,-(SP)
	022130	012746	000001			MOV 01,-(SP)
	022134	010600				MOV SP,R0
	022136	104417				TRAP C:PNTF
	022140	062706	000004			ADD 04,SP
3253	022144			DELAY	200.	;(GO TO SUPERVISOR, WAIT 2 SECONDS.
	022144	012727	000310			MOV 0200.,(PC).
	022150	000000				.WORD 0
	022152	013727	002116			MOV L:DLY,(PC).
	022156	000000				.WORD 0
	022160	005367	177772			DEC -6(PC)
	022164	001375				BNE -4
	022166	005767	177756			DEC -22(PC)
	022172	001367				BNE -20
3254	022174			BREAK		;(GO TO SUPERVISOR, CHECK TTY.
	022174	104422				TRAP C:BRK
3255	022176			DOCLN		;(DO CLEAN CODE + ABORT PASS.
	022176	104444				TRAP C:DCLN
3256						
3257	022200				503601:	
3258						
3259						
3260	022200			SETPRI	0PRI00	;(LOWER CPU PRIORITY TO 0
	022200	012700	000000			MOV 0PRI00,R0
	022204	104441				TRAP C:SPRI
3261	022206	105737	002213	TSTB	IREC	;(IF ERROR RECOVERY IS ENABLED
3262	022212	001033		BNE	11	
3263	022214	032737	000020 003536	BIT	0ADR.OPFLAG	
3264	022222	001027		BNE	11	
3265	022224	004737	017102	JSR	PC,FIRSTU	;(AND AUTO-DROP NOT CALLED, THEN SET UP FOR FIRST
3266	022230					;(WHILE THERE ARE MORE DEVICES:
3267	022230	026527	002604 177777	CMP	DEVTBL(R5),0END	
3268	022236	001421		BEQ	11	
3269	022240	105037	003530	CLRB	TRAP04	;(CLEAR TRAP FLAG
3270	022244			SETVEC	04,0TRAP4,0INTPRI	;(SET VECTOR 4,PRIORITY 06
	022244	012746	000300			MOV 0INTPRI,-(SP)
	022250	012746	023756			MOV 0TRAP4,-(SP)

INITIALIZE SECTION

```

022254 012746 000004
022260 012746 000003
022264 104437
022266 062706 000010
3271
3272 022272 012737 000001 003436      MOV    01,TIME1
3273 022300 000404                      BR     50365$
3274 022302 000137 023130      1$:   JMP    50363$
3275
3276          50366$:
3277 022306 005237 003436      INC    TIME1
3278 022312          50365$:
3279 022312 023727 003436 000025      CMP    TIME1,025
3280 022320 003134                      BGT    4$
3281 022322 012775 002340 002514      MOV    0GSCP,0TSD8(R5)
3282 022330          DELAY 25
                                ;AND GET UNITS STATUS
                                ;WAIT
022330 012727 000025
022334 000000
022336 013727 002116
022342 000000
022344 005367 177772
022350 001375
022352 005367 177756
022356 001367
3283 022360          CLRVEC 04
                                ;CLEAR VECTOR AT 4
022360 012700 000004
022364 104436
3284 022366 105737 003530      TSTB   TRAPD4
3285 022372 001423                      BEQ    2$
3286 022374 005265 003366      INC    FTLCNT(R5)
3287 022400          PRINTF 0NODEV,TSSR(R5)
                                ;PRINT ERROR
022400 016546 002524
022404 012746 005543
022410 012746 000002
022414 010600
022416 104417
022420 062706 000006
3288 022424 016537 002604 0174:0      MOV    DEVTBL(R5),DROPN
3289 022432 010500                      MOV    R5,R0
3290 022434 006200                      ASR    R0
3291 022436          DODU   R0
                                ;DROP THE UNIT
022436 104451
3292
3293 022440          DOCLN
                                ; EXEC BGNDU-ENODU CODE IF IDU = 0
                                ;DO CLEAN CABORT
022440 104444
3294
3295 022442 105037 003530      2$:   CLRB   TRAPD4
3295 022446          SETVEC 04,0TRAP4,0INTPRI
                                ;CLEAR TRAP FLAG
                                ;SET VECTOR 4,PRIORITY 06
022446 012746 000300
022452 012746 023756
022456 012746 000004
022462 012746 000003
022466 104437
022470 062706 000010
3297 022474 005775 002524      TST    0TSSR(R5)
3298 022500          DELAY 25
                                ;CHECK FOR ADDRESS
                                ;WAIT
022500 012727 000025
                                MOV    025,(PC)
                                ;START 3.5 MINUTE COUNTER
                                ;INCR TIME1 FROM 01 TO 025 BY 01
                                MOV    04,-(SP)
                                MOV    03,-(SP)
                                TRAP   C$SVEC
                                ADD    010,SP
                                MOV    025,(PC)
                                .WORD 0
                                MOV    L$DLY,(PC)
                                .WORD 0
                                DEC    -6(PC)
                                BNE    -4
                                DEC    -22(PC)
                                BNE    -20
                                MOV    04,R0
                                TRAP   C$CVEC
                                MOV    TSSR(R5),-(SP)
                                MOV    0NODEV,-(SP)
                                MOV    02,-(SP)
                                MOV    SP,R0
                                TRAP   C$PNTF
                                ADD    06,SP
                                TRAP   C$DODU
                                ; EXEC BGNDU-ENODU CODE IF IDU = 0
                                TRAP   C$DCLN
                                MOV    0INTPRI, -(SP)
                                MOV    0TRAP4, -(SP)
                                MOV    04, -(SP)
                                MOV    03, -(SP)
                                TRAP   C$SVEC
                                ADD    010,SP
                                MOV    025,(PC)

```


INITIALIZE SECTION

```

3357
3358 023124 000137 022230          JMP      50362$
3359
3360 023130          50363$:
3361 023130          50361$:
3362 023130 105737 003527          TSTB    PWRFLG ;IFB PWRFLG EQ #0 THEN
3363 023134 001026          BNE     50400$
3364 023136          MEMORY DATAWT          ;REQUEST MEMORY FROM SUPER FOR RD/WR BUFFERS.
                                TRAP    C$MEM
                                MOV     RO,DATAWT
                                023136 104431
                                023140 010037 003406          MOV     DATAWT,DATARD ;SET RD BFR ADDRESS
3365 023144 013737 003406 003410          ADD     #DATCNT,DATARD
3366 023152 062737 004000 003410          CMP     @DATAWT,@DATCNT ;WHEN NOT ENOUGH FREE MEMO AVAILABLE
3367 023160 027727 160222 004000          BGE     50401$
3368 023166 002011          PRINTF @MEMOM          ;WARN OPERATOR
3369 023170          MOV     @MEMOM,-(SP)
                                023170 012746 023236          MOV     #1,-(SP)
                                023174 012746 000001          MOV     SP,RO
                                023200 010600          TRAP   C$PNTF
                                023202 104417          ADD     #4,SP
                                023204 062706 000004          DOCLN          ;AND ABORT PASS
3370 023210          TRAP   C$DCLN
                                023210 104444          ;DIAG MUST BE RE-LOADED IN A CPU WITH LARGER MEMO
3371
3372 023212          50401$:
3373
3374 023212          50400$:
3375
3376 023212 105037 002214          CLRB   CHGFLG          ;CLR CHANGE CMD SEQ TBL FLAG.
3377 023216 012703 003526          MOV     #ENDFLG,R3    ;LET R3 := #ENDFLG
3378 023222 004737 012664          JSR     PC,CLRERR     ;CLEAR ALL FLAGS.
3379 023226 105037 003527          CLRB   PWRFLG        ;CLEAR THE POWER FAIL FLAG.
3380
3381 023232          EXIT    INIT
                                023232 104432          TRAP   C$EXIT
                                023234 000104          .WORD  L10012-.
3382 023236 045 101 106 MEMOM: .ASCII /*AFREE MEMO TOO SMALL FOR RD-WR BFRS*/
                                023241 122 105 105
                                023244 040 115 105
                                023247 115 117 040
                                023252 124 117 117
                                023255 040 123 115
                                023260 101 114 114
                                023263 040 106 117
                                023266 122 040 122
                                023271 104 055 127
                                023274 122 040 102
                                023277 106 122 123
                                023302 045 116
3383 023304 045 101 122 .ASCIZ /*ARE-LOAD IN LARGER MEMO*/
                                023307 105 055 114
                                023312 117 101 104
                                023315 040 111 116
                                023320 040 114 101
                                023323 122 107 105
                                023326 122 040 115
                                023331 105 115 117
                                023334 045 116 000

```

INITIALIZE SECTION

```

3384 .EVEN
3385
3386 023340 ENDINIT
023340 L10012:
023340 104411 TRAP C$INIT
3387
3388 .SBTTL AUTO DROP SECTION
3389
3390 ;**
3391 ;SECTION EXECUTED AFTER THE INIT CODE WHEN "ADR" FLAG IS SET BY OPERATOR
3392 ;SECTION CHECKS FOR A VALID INTERFACE LOCATION. DROPS UNIT IF NO RESPONSE
3393 ;FROM INTERFACE
3394 ;--
3395
3396 023342 BGNAUTO
023342 L$AUTO::
3397
3398 023342 004737 017102 JSR PC,FIRSTU ;FIND FIRST UNIT
3399 023346 50402$: ;WHILE DEVTBL(R5) NE #END DO ;
3400 023346 026527 002604 177777 CMP DEVTBL(R5),#END
3401 023354 001525 BEQ 50403$
3402 023356 105037 003530 CLRB TRAPD4 ;LET TRAPD4 :B= #0 ;
3403 023362 SETVEC #4,#TRAP4,#INTPRI ;SET VECTOR 4 ;
023362 012746 000300 MOV #INTPRI,-(SP)
023366 012746 023756 MOV #TRAP4,-(SP)
023372 012746 000004 MOV #4,-(SP)
023376 012746 000003 MOV #3,-(SP)
023402 104437 TRAP C$SVEC
023404 062706 000010 ADD #10,SP
3404 023410 017502 002514 MOV @TSDB(R5),R2 ;ADDRESS TS05 INTERFACE
3405 023414 CLRVEC #4 ;CLEAR VECTOR AT 4
023414 012700 000004 MOV #4,R0
023420 104436 TRAP C$CVEC
3406 023422 105737 003530 TSTB TRAPD4 ;IFB TRAPD4 NE #0 THEN
3407 023426 001423 REQ 50404$
3408 023430 005265 003366 INC FTLCNT(R5) ;LET FTLCNT(R5) := FTLCNT(R5) + #1
3409 023434 PRINTF #AUTODM,TSDB(R5) ;PRINT ERROR
023434 016546 002514 MOV TSDB(R5),-(SP)
023440 012746 023632 MOV #AUTODM,-(SP)
023444 012746 000002 MOV #2,-(SP)
023450 010600 MOV SP,R0
023452 104417 TRAP C$PNTF
023454 062706 000006 ADD #6,SP
3410 023460 016537 002604 017410 MOV DEVTBL(R5),DROPN ;SAVE # OF UNIT TO BE DROPPED.
3411 023466 010500 MOV R5,R0 ;R0=LOGICAL DEVICE NUMBER
3412 023470 006200 ASR R0
3413 023472 DODU R0 ;DROP THE UNIT: EXEC BGNDU-ENDDU CODE IF IDU = 0
023472 104451 TRAP C$DODU
3414
3415 023474 000452 BR 50405$
3416 023476 50404$:
3417 023476 012775 002340 002514 MOV #GSCP,#TSDB(R5) ;SEND GET STATUS COMMAND
3418 023504 004737 012700 JSR PC,WSSR ;WAIT
3419 023510 032775 000200 002524 BIT #TS.SSR,@TSSR(R5) ;IF #TS.SSR SETIN @TSSR(R5) THEN
3420 023516 001423 BEQ 50406$
3421 023520 032775 000100 002524 BIT #TS.OFL,@TSSR(R5) ;IF #TS.OFL SETIN @TSSR(R5) THEN
3422 023526 001416 BEQ 50407$

```

AUTO DROP SECTION

```

3423 023530 005265 003366          INC      FTLCNT(R5)          ;LET FTLCNT(R5) := FTLCNT(R5) + #1
3424 023534          PRINTF #OFLINM,TSNP
      023534 013746 003534          MOV      TSNP, -(SP)
      023540 012746 005456          MOV      #OFLINM, -(SP)
      023544 012746 000002          MOV      #2, -(SP)
      023550 010600          MOV      SP,R0
      023552 104417          TRAP    C$PNTF
      023554 062706 000006          ADD      #6,SP
3425 023560 004737 017324          JSR PC,DROPUA
3426
3427 023564          50407$:
3428
3429 023564 000416          BR      50410$
3430 023566          50406$:
3431 023566 005265 003366          INC      FTLCNT(R5)          ;LET FTLCNT(R5) := FTLCNT(R5) + #1
3432 023572          PRINTF #NRDYM,DEVTBL(R5)
      023572 016546 002604          MOV      DEVTBL(R5), -(SP)
      023576 012746 023726          MOV      #NRDYM, -(SP)
      023602 012746 000002          MOV      #2, -(SP)
      023606 010600          MOV      SP,R0
      023610 104417          TRAP    C$PNTF
      023612 062706 000006          ADD      #6,SP
3433 023616 004737 017324          JSR PC,DROPUA
3434
3435 023622          50410$:
3436
3437 023622          50405$:
3438 023622 004737 017150          JSR PC,NEXTU
3439
3440 023626 000647          BR      50402$
3441 023630          50403$:
3442
3443 023630          ENDAUTO
      023630          L10013:
      023630 104461          TRAP    C$AUTO
3444
3445 023632 045 101 102 AUTODM: .ASCII /#ABUS TRAP AT #06#N/
      023635 125 123 040
      023640 124 122 101
      023643 120 040 101
      023646 124 040 045
      023651 117 066 045
      023654 116
3446 023655 045 101 111 .ASCIZ /#AINTERFACE BAD OR NOT SET TO ABOVE AD#N/
      023660 116 124 105
      023663 122 106 101
      023666 103 105 040
      023671 102 101 104
      023674 040 117 122
      023677 040 116 117
      023702 124 040 123
      023705 105 124 040
      023710 124 117 040
      023713 101 102 117
      023716 126 105 040
      023721 101 104 045
      023724 116 000

```

AUTO DROP SECTION

```

3447 023726      045      101      125 NRDYM: .ASCIZ /*AUNIT *D1*A NOT RDY*N/
      023731      116      111      124
      023734      040      045      104
      023737      061      045      101
      023742      040      116      117
      023745      124      040      122
      023750      104      131      045
      023753      116      000
3448
3449
3450
3451
3452
3453
3454 023756 105237 003530 TRAP4:: INCB TRAPD4;LET TRAPD4 ;B= TRAPD4 + #1
3455 023762 000002 RTI
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465 023764 BGNCLN
      023764 L$CLEAN::
3466
3467 023764 004737 017102 JSR PC,FIRSTU ;FIND FIRST UNIT.
3468 023770 50411$: ;WHILE DEVTBL(R5) NE #END DO
3469 023770 026527 002604 177777 CMP DEVTBL(R5),#END
3470 023776 001410 BEQ 50412$
3471 024000 004737 012700 JSR PC,WSSR ;WAIT FOR UNIT READY OR TIMEOUT,
3472 024004 CLRVEC TSVCT(R5) ;RELEASE INTERRUPT VECTORS FOR ALL DEV.
      024004 016500 002534 MOV TSVCT(R5),R0
      024010 104436 TRAP C$CVEC
3473 024012 004737 017150 JSR PC,NEXTU ;FIND NEXT UNIT.
3474
3475 024016 000764 BR 50411$
3476 024020 50412$:
3477
3478 024020 EXIT CLN
      024020 104432 TRAP C$EXIT
      024022 000002 .WORD L10014-.
3479
3480
3481 024024 ENDCLN
      024024 L10014:
      024024 104412 TRAP C$CLEAN
3482
3483
3484
3485
3486
3487
3488
3489

```

.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
; --

```

DROP UNIT SECTION

```

3490
3491 024026          BGNDU
      024026          L$DU::
3492
3493 024026 010005          MOV      R0,R5          ;R5 = LOGICAL DEVICE NUMBER X 2.
3494 024030 006305          ASL      R5
3495 024032 012765 177774 002604  MOV      #NINUSE,DEVTBL(R5) ;SET NOT IN USE FLAG FOR THE DEVICE.
3496 024040          CLRVEC TSVCT(R5)          ;RELEASE THE INTERRUPT VECTOR.
      024040 016500 002534          MOV      TSVCT(R5),R0
      024044 104436          TRAP     C$CVEC
3497 024046          PRINTF #DROPNM,DROPN          ;PRINT DROP DEVICE MESSAGE
      024046 013746 017410          MOV      DROPN,-(SP)
      024052 012746 005065          MOV      #DROPNM,-(SP)
      024056 012746 000002          MOV      #2,-(SP)
      024062 010600          MOV      SP,R0
      024064 104417          TRAP     C$PNTF
      024066 062706 000006          ADD      #6,SP
3498 024072          EXIT      DU
      024072 000167          .WORD   J$JMP
      024074 000000          .WORD   L10015-2-.
3499
3500          .EVEN
3501 024076          ENDDU
      024076          L10015:
      024076 104453          TRAP     C$DU
3502
3503          .SETTL ADD UNIT SECTION
3504
3505          ;**
3506          ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
3507          ; TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING. IF
3508          ; "EF.AUNIT" IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
3509          ;--
3510
3511 024100          BGNAU
      024100          L$AU::
3512 024100 010005          MOV      R0,R5          ;R5 = LOGICAL DEVICE NUMBER X 2.
3513 024102 006305          ASL      R5
3514 024104 010065 002604  MOV      R0,DEVTBL(R5) ;STORE UNIT # IN DEVICE TABLE.
3515 024110          GPHARD R0,R0          ;GET HARDWARE P TABLE FROM SUPER.
      024110 104442          TRAP     C$GPHRD
3516 024112 011065 002514          MOV      (R0),TSDB(R5) ;SAVE TSDB ADDRESS.
3517 024116 012065 002524          MOV      (R0)+,TSSR(R5) ;SAVE TSSR ADDRESS.
3518 024122 062765 000002 002524  ADD      #2,TSSR(R5)
3519 024130 011065 002534          MOV      (R0),TSVCT(R5) ;SAVE INTERRUPT VECTOR ADDRESS.
3520 024134 011065 003532          MOV      (R0),TSUNT(R5) ;SAVE NUMBER OF DRIVE
3521 024140 011037 003534          MOV      (R0),TSNP ;SAVE FOR PRINT OUT'S
3522 024144          SETVEC TSVCT(R5),TSSINT(R5),#INTPRI
      024144 012746 000300          MOV      #INTPRI,-(SP)
      024150 016546 002554          MOV      TSSINT(R5),-(SP)
      024154 016546 002534          MOV      TSVCT(R5),-(SP)
      024160 012746 000003          MOV      #3,-(SP)
      024164 104437          TRAP     C$SVEC
      024166 062706 000010          ADD      #10,SP
3523
3524 024172 005065 003472  CLR      INTFLG(R5) ;SET UP INTERRUPT PROCESSING CONDITIONS.
3525          ;CLEAR INTERRUPT FLAGS.

```

ADD UNIT SECTION

```

3526 024176          EXIT    AU
      024176 000167
      024200 000000          .WORD  J$JMP
                                .WORD  L10016-2-.
3527
3528          .EVEN
3529
3530 024202          ENDAU
      024202          L10016:
      024202 104452          TRAP    C$AU
3531
3532
3533
3534          .TITLE  HARDWARE TESTS
3535          .SBTTL  TEST 1:  BASIC FUNCTIONS.
3536
3537          ;++
3538          ; TEST TO EXECUTE ALL TS05 FUNCTIONS.
3539          ;--
3540
3541
3542 024204          BGNMOD
3543
3544 024204          BGNTST
      024204          T1::
3545
3546 024204 105037 003515          CLR   RANDOM          ;CLR THE RANDOM OPERATIONS FLAG.
3547 024210 105037 003514          CLR   EXPBOT          ;CLR EXPECT BOT FLAG.
3548
3549 024214          BGNSUB          ;SUBTEST 1 - SET CHAR, DRIVE INIT, GET STATUS.
      024214          T1.1:
      024214 104402          TRAP    C$BSUB
3550
3551 024216 004737 017102          JSR   PC,FIRSTU          ;FIND THE FIRST UNIT.
3552 024222 004737 007072          JSR   PC,SOFINIT        ;INIT DEVICE
3553 024226 103404
3554 024230          BCS   11$
      024230 104455          ERRDF  2,NSSRM,STAERM          ;REPORT TS05 NOT READY
      024232 000002          TRAP    C$ERDF
      024234 004536          .WORD  2
      024236 006120          .WORD  NSSRM
                                .WORD  STAERM
3555
3556 024240 004737 007466          11$: JSR   PC,MOSET          ;GO DO SETUP'S
3557 024244 012702 025122          MOV   #BFSEQ0,R2          ;ADR OF CMD SEQ.
3558 024250 004737 025076          JSR   PC,BFSEQ          ;SET UP CMD SEQ.
3559 024254 004737 010226          JSR   PC,EXALL          ;EXECUTE CMD SEQ ON ALL DEVICES.
3560 024260 004737 017102          JSR   PC,FIRSTU          ;FIND THE FIRST UNIT.
3561 024264          50413$: ;WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
3562 024264 026527 002604 177777          CMP   DEVTBL(R5),#END
3563 024272 001451          BEQ   50414$
3564 024274 016502 002544          MOV   MSGPKA(R5),R2          ;GET MSG PACKET ADR,
3565 024300 062702 000012          ADD   #12,R2          ;LET R2 := R2 + #12          ;GET XSTAT2 ADR,
3566 024304 011265 002564          MOV   (R2),TS5CL(R5)          ;STORE CODE LEVEL FROM DTR BYTE,
3567 024310 042765 177700 002564          BIC   #177700,TS5CL(R5)
3568 024316 011265 002574          MOV   (R2),TS5SW(R5)          ;STORE SWITCH SETTINGS
3569 024322 042765 177477 002574          BIC   #177477,TS5SW(R5)
3570 024330          PRINTF #CODELM,DEVTBL(R5),TS5CL(R5)
      024330 016546 002564          MOV   TS5CL(R5),-(SP)

```

TEST 1: BASIC FUNCTIONS.

024334	016546	002604				MOV	DEVTBL(R5), -(SP)
024340	012746	004162				MOV	#CODELM, -(SP)
024344	012746	000003				MOV	#3, -(SP)
024350	010600					MOV	SP, R0
024352	104417					TRAP	C#PNTF
024354	062706	000010				ADD	#10, SP
3571							
3572	024360			PRINTF	#SWSET, DEVTBL(R5), TS5SW(R5)		;PRINT THE TS05 MICROCODE LEVEL.
	024360	016546	002574			MOV	TS5SW(R5), -(SP)
	024364	016546	002604			MOV	DEVTBL(R5), -(SP)
	024370	012746	004231			MOV	#SWSET, -(SP)
	024374	012746	000003			MOV	#3, -(SP)
	024400	010600				MOV	SP, R0
	024402	104417				TRAP	C#PNTF
	024404	062706	000010			ADD	#10, SP
3573							;PRINT THE TS05 SWITCH SETTINGS.
3574	024410			50415\$:			
3575	024410	004737	017150		JSR	PC, NEXTU	;FIND NEXT UNIT.
3576							
3577	024414	000723			BR	50415\$	
3578	024416			50414\$:			
3579							
3580	024416				ENDSUB		
	024416			L10020:			
	024416	104403					TRAP C#ESUB
3581							
3582	024420				BGNSUB		;SUBTEST 2 - REWIND.
	024420			T1.2:			
	024420	104402					TRAP C#BSUB
3583							
3584	024422	012702	025174		MOV	#BFSEQ1, R2	;ADR OF CMD SEQ.
3585	024426	004737	025076		JSR	PC, BFSEQ	;SET UP CMD SEQ.
3586	024432	004737	010226		JSR	PC, EXALL	;EXECUTE CMD SEQ ON ALL DEVICES.
3587	024435	105037	003526		CLRB	STAF LG	;CLEAR START FLAG
3588	024442				ENDSUB		
	024442			L10021:			
	024442	104403					TRAP C#ESUB
3589							
3590	024444				BGNSUB		;SUBTEST 3 - WRITE/VERIFY.
	024444			T1.3:			
	024444	104402					TRAP C#BSUB
3591							
3592	024446	012702	025206		MOV	#BFSEQ2, R2	;ADR OF CMD SEQ.
3593	024452	004737	025076		JSR	PC, BFSEQ	;SET UP CMD SEQ.
3594	024456	004737	010226		JSR	PC, EXALL	;EXECUTE CMD SEQ ON ALL DEVICES.
3595	024462				ENDSUB		
	024462			L10022:			
	024462	104403					TRAP C#ESUB
3596							
3597	024464				BGNSUB		;SUBTEST 4 - WRITE TAPE MARK, ERASE.
	024464			T1.4:			
	024464	104402					TRAP C#BSUB
3598							
3599	024466	012702	025300		MOV	#BFSEQ3, R2	;ADR OF CMD SEQ.
3600	024472	004737	025076		JSR	PC, BFSEQ	;SET UP CMD SEQ.
3601	024476	004737	010226		JSR	PC, EXALL	;EXECUTE CMD SEQ ON ALL DEVICES.
3602	024502				ENDSUB		

TEST 1: BASIC FUNCTIONS.

	024502			L10023:					
	024502	104403						TRAP	C\$ESUB
3603									
3604	024504				BGNSUB				;SUBTEST 5 - SPACE FILES.
	024504			T1.5:					
	024504	104402						TRAP	C\$BSUB
3605									
3606	024506	012702	025352		MOV	#BFSEQ4,R2			;ADR OF CMD SEQ.
3607	024512	004737	025076		JSR	PC,BFSEQ			;SET UP CMD SEQ.
3608	024516	004737	010226		JSR	PC,EXALL			;EXECUTE CMD SEQ ON ALL DEVICES.
3609	024522				ENDSUB				
	024522			L10024:					
	024522	104403						TRAP	C\$ESUB
3610									
3611	024524				BGNSUB				;SUBTEST 6 - SPACE RECORDS.
	024524			T1.6:					
	024524	104402						TRAP	C\$BSUB
3612									
3613	024526	012702	025414		MOV	#BFSEQ5,R2			;ADR OF CMD SEQ.
3614	024532	004737	025076		JSR	PC,BFSEQ			;SET UP CMD SEQ.
3615	024536	004737	010226		JSR	PC,EXALL			;EXECUTE CMD SEQ ON ALL DEVICES.
3616	024542				ENDSUB				
	024542			L10025:					
	024542	104403						TRAP	C\$ESUB
3617									
3618	024544				BGNSUB				;SUBTEST 7 - WRITE RETRY.
	024544			T1.7:					
	024544	104402						TRAP	C\$BSUB
3619									
3620	024546	012702	025466		MOV	#BFSEQ6,R2			;ADR OF CMD SEQ.
3621	024552	004737	025076		JSR	PC,BFSEQ			;SET UP CMD SEQ.
3622	024556	004737	010226		JSR	PC,EXALL			;EXECUTE CMD SEQ ON ALL DEVICES.
3623	024562				ENDSUB				
	024562			L10026:					
	024562	104403						TRAP	C\$ESUB
3624									
3625	024564				BGNSUB				;SUBTEST 8 - READ REV RETRY.
	024564			T1.8:					
	024564	104402						TRAP	C\$BSUB
3626									
3627	024566	012702	025540		MOV	#BFSEQ7,R2			;ADR OF CMD SEQ.
3628	024572	004737	025076		JSR	PC,BFSEQ			;SET UP CMD SEQ.
3629	024576	004737	010226		JSR	PC,EXALL			;EXECUTE CMD SEQ ON ALL DEVICES.
3630	024602				ENDSUB				
	024602			L10027:					
	024602	104403						TRAP	C\$ESUB
3631									
3632	024604				BGNSUB				;SUBTEST 9 - READ FWD RETRY.
	024604			T1.9:					
	024604	104402						TRAP	C\$BSUB
3633									
3634	024606	012702	025572		MOV	#BFSEQ8,R2			;ADR OF CMD SEQ.
3635	024612	004737	025076		JSR	PC,BFSEQ			;SET UP CMD SEQ.
3636	024616	004737	010226		JSR	PC,EXALL			;EXECUTE CMD SEQ ON ALL DEVICES.
3637	024622				ENDSUB				
	024622			L10030:					
	024622	104403						TRAP	C\$ESUB

11.0

TEST 1: BASIC FUNCTIONS.

```

3638
3639 024624          BGNSUB          ;SUBTEST 10 - CLEAN.
      024624          T1.10:
      024624 104402          TRAP      C#BSUB
3640
3641 024626 012702 025624      MOV      #BFSEQ9,R2      ;ADR OF CMD SEQ.
3642 024632 004737 025076      JSR      PC,BFSEQ      ;SET UP CMD SEQ.
3643 024636 004737 010226      JSR      PC,EXALL      ;EXECUTE CMD SEQ ON ALL DEVICES.
3644 024642          ENDSUB
      024642          L10031:
      024642 104403          TRAP      C#BSUB
3645
3646 024644          BGNSUB          ;SUBTEST 11 - WTV SWAPPED DATA BYTES.
      024644          T1.11:
      024644 104402          TRAP      C#BSUB
3647 024646 012702 025646      MOV      #BFSEQ10,R2     ;ADR OF CMD SEQ.
3648 024652 004737 025076      JSR      PC,BFSEQ      ;SET UP CMD SEQ.
3649 024656 004737 010226      JSR      PC,EXALL      ;WRITE/VERIFY RECORDS 1 AND 2.
3650 024662 112737 000001 003520  MOVB     #1,SWBFLG      ;ENABLE BYTE SWAPPING.
3651 024670 004737 010226      JSR      PC,EXALL      ;WRITE/VERIFY RECORDS 3 AND 4.
3652 024674 105037 003520      CLRB     SWBFLG      ;DISABLE BYTE SWAPPING.
3653 024700          ENDSUB
      024700          L10032:
      024700 104403          TRAP      C#BSUB
3654 024702 013702 003406      MOV      DATAW,R2      ;INIT WRITE BUFFER POINTER.
3655 024706 062702 000012      ADD     #10.,R2
3656 024712          50416: ;WHILE R2 NE DATAW DO ;UNTIL 10 BYTES HAVE BEEN SWAPPED.
3657 024712 020237 003406      CMP     R2,DATAW
3658 024716 001402          BEQ     50417:
3659 024720 000342          SWAB   -(R2)          ;SWAP DATA BYTES IN WRITE BUFFER.
3660
3661 024722 000773          BR     50416:
3662 024724          50417:
3663 024724 105237 003523      INCR   T1SWB          ;SET T1 SWAP BYTES FLAG FOR "CKDATA" SUBR
3664
3665 024730          BGNSUB          ;SUBTEST 12 - READ SWAPPED DATA BYTES.
      024730          T1.12:
      024730 104402          TRAP      C#BSUB
3666 024732 012737 104401 003420  MOV     #ORDR,CMDWRD      ;CMD IS READ REV.
3667 024740 004737 016152          JSR     PC,VFEXC      ;VERIFY ODD LENGTH SWAP (RECORD 4).
3668 024744 012737 000012 002336  MOV     #12,CMDPKT+CP.CNT ;CHANGE BYTE COUNT TO 10.
3669 024752 004737 016152          JSR     PC,VFEXC      ;VERIFY EVEN LENGTH SWAP (RECORD 3).
3670 024756 112737 000001 003520  MOVB     #1,SWBFLG      ;ENABLE BYTE SWAPPING.
3671 024764 012737 000011 002336  MOV     #11,CMDPKT+CP.CNT ;CHANGE BYTE COUNT TO 9.
3672 024772 004737 016152          JSR     PC,VFEXC      ;VERIFY ODD LENGTH SWAP (RECORD 2).
3673 024776 012737 000012 002336  MOV     #12,CMDPKT+CP.CNT ;CHANGE BYTE COUNT TO 10.
3674 025004 004737 016152          JSR     PC,VFEXC      ;VERIFY EVEN LENGTH SWAP (RECORD 1).
3675 025010 012737 104001 003420  MOV     #ORDF,CMDWRD      ;CMD IS READ FWD.
3676 025016 004737 016152          JSR     PC,VFEXC      ;VERIFY EVEN LENGTH SWAP (RECORD 1).
3677 025022 012737 000011 002336  MOV     #11,CMDPKT+CP.CNT ;CHANGE BYTE COUNT TO 9.
3678 025030 004737 016152          JSP     PC,VFEXC      ;VERIFY ODD LENGTH SWAP (RECORD 2).
3679 025034 105037 003520      CLRB     SWBFLG      ;DISABLE BYTE SWAPPING.
3680 025040 012737 000012 002336  MOV     #12,CMDPKT+CP.CNT ;CHANGE BYTE COUNT TO 10.
3681 025046 004737 016152          JSR     PC,VFEXC      ;VERIFY EVEN LENGTH SWAP (RECORD 3).
3682 025052 012737 000011 002336  MOV     #11,CMDPKT+CP.CNT ;CHANGE BYTE COUNT TO 9.
3683 025060 004737 016152          JSR     PC,VFEXC      ;VERIFY ODD LENGTH SWAP (RECORD 4).
3684

```

TEST 1: BASIC FUNCTIONS.

```

3685 025064          ENDSUB
      025064          L10033:
      025064 104403          TRAP      CSF00B
3686
3687 025066 105037 003523  CLR B      Y15WB          ;CLEAR '1 SWAP BYTES FLAG
3688
3689 025072          EXIT      TST
      025072 104432          TRAP      CSEXIT
      025074 000574          .WORD    L10017.
3690
3691          ;      SUBROUTINE TO MOVE A COMMAND SEQUENCE TO THE SEQUENCE TABLE.
3692          ;      INPUTS:          R2 = FWA OF COMMAND SEQUENCE.
3693          ;      OUTPUTS:
3694          ;      REGISTERS:
3695          ;      CALLS:
3696
3697 025076 012701 003540  BFSEQ:: MOV      #CMDSEQ,R1          ;INIT SEQ TABLE ADDRESS.
3698 025102 50420$: ;WHILE (R2) NE #END DO          ;WHILE THERE ARE MORE COMMANDS:
3699 025102 021227 177777  CMP      (R2),#END
3700 025106 001402          BEQ      50421$
3701 025110 012221          MOV      (R2)+,(R1)+          ;MOVE COMMANDS TO SEQ TABLE.
3702
3703 025112 000773          BR      50420$
3704 025114 50421$:
3705 025114 012711 177777  MOV      #END,(R1)          ;STORE END OF SEQUENCE CODE.
3706 025120 000207          RTS      PC          ;RETURN.
3707
3708
3709          ;      BASIC FUNCTION COMMAND SEQUENCE
3710
3711 025122 140004          BFSEQ0: .WORD    SCH          ;SET CHAR. 200.          (1)
3712 025124 000200          200
3713 025126 000001          1
3714 025130 000000          0
3715 025132 100013          DRI          ;DRIVE INIT.          (2)
3716 025134 000001          1
3717 025136 000001          1
3718 025140 000000          0
3719 025142 140004          SCH          ;SET CHAR. 20          (3)
3720 025144 000020          20
3721 025146 000001          1
3722 025150 000000          0
3723 025152 100017          GES          ;GET STATUS.          (4)
3724 025154 000001          1
3725 025156 000001          1
3726 025160 000000          0
3727 025162 140004          SCH          ;SET CHAR. 40.          (5)
3728 025164 000040          40
3729 025166 000001          1
3730 025170 000000          0
3731 025172 177777          .WORD    END
3732
3733 025174 102010          BFSEQ1:          RWD          ;REWIND TWICE.          (6)
3734 025176 000001          1
3735 025200 000000          2
3736 025202 000000          0
3737 025204 177777          .WORD    END

```

TEST 1: BASIC FUNCTIONS.

```

3738
3739 025206 104105          BFSEQ2:          WTV          ;WRITE/VERIFY PAT 1.  (7)
3740 025210 004000          DATCNT
3741 025212 000001          1
3742 025214 000001          1
3743 025216 104105          WTV          ;WTV PAT 2.          (8)
3744 025220 004000          DATCNT
3745 025222 000001          1
3746 025224 000002          2
3747 025226 104105          WTV          ;WTV PAT 3.          (9)
3748 025230 004000          DATCNT
3749 025232 000001          1
3750 025234 000003          3
3751 025236 104105          WTV          ;WTV PAT 4.          (10)
3752 025240 004000          DATCNT
3753 025242 000001          1
3754 025244 000004          4
3755 025246 104105          WTV          ;WTV PAT 5.          (11)
3756 025250 004000          DATCNT
3757 025252 000001          1
3758 025254 000005          5
3759 025256 104105          WTV          ;WTV PAT 6.          (12)
3760 025260 004000          DATCNT
3761 025262 000001          1
3762 025264 000006          6
3763 025266 104105          WTV          ;WTV PAT 0.          (13)
3764 025270 004000          DATCNT
3765 025272 000001          1
3766 025274 000000          0
3767 025276 177777          .WORD        END
3768
3769 025300 100011          BFSEQ3:          WTM          ;WRITE TAPE MARK.    (14)
3770 025302 000001          1
3771 025304 000001          1
3772 025306 000000          0
3773 025310 104005          WRT          ;WRITE 10 RECORDS.  (15)
3774 025312 004000          DATCNT
3775 025314 000010          10
3776 025316 000001          1
3777 025320 100411          ERS          ;ERASE 10 TIMES.    (16)
3778 025322 000001          1
3779 025324 000010          10
3780 025326 000000          0
3781 025330 100011          WTM          ;WRITE TAPE MARK.    (17)
3782 025332 000001          1
3783 025334 000001          1
3784 025336 000000          0
3785 025340 101011          WTR          ;WTM RETRY           (18)
3786 025342 000001          1
3787 025344 000001          1
3788 025346 000000          0
3789 025350 177777          .WORD        END
3790
3791 025352 105410          BFSEQ4:          SFR          ;SPACE 2 FILES REV.  (19)
3792 025354 000002          2
3793 025356 000001          1
3794 025360 000000          0
    
```

TEST 1: BASIC FUNCTIONS.

3795	025362	105010		SFF		;SPACE 2 FILES FWD.	(20)
3796	025364	000002		2			
3797	025366	000001		1			
3798	025370	000000		0			
3799	025372	105410		SFR		;SPACE 2 FILES REV.	(21)
3800	025374	000001		1			
3801	025376	000002		2			
3802	025400	000000		0			
3803	025402	105010		SFF		;SPACE 2 FILES FWD.	(22)
3804	025404	000001		1			
3805	025406	000002		2			
3806	025410	000000		0			
3807	025412	177777	.WORD	END			
3808							
3809	025414	102010	BFSEQ5:	RWD		;REWIND.	(23)
3810	025416	000001		1			
3811	025420	000001		1			
3812	025422	000000		0			
3813	025424	104010		SRF		;SPACE 7 RECORDS FWD.	(24)
3814	025426	000007		7			
3815	025430	000001		1			
3816	025432	000000		0			
3817	025434	104410		SRR		;SPACE 7 RECORDS REV.	(25)
3818	025436	000007		7			
3819	025440	000001		1			
3820	025442	000000		0			
3821	025444	104010		SRF		;SPACE 7 RECORDS FWD.	(26)
3822	025446	000001		1			
3823	025450	000007		7			
3824	025452	000000		0			
3825	025454	104410		SRR		;SPACE 7 RECORDS REV.	(27)
3826	025456	000001		1			
3827	025460	000007		7			
3828	025462	000000		0			
3829	025464	177777	.WORD	END			
3830							
3831	025466	102010	BFSEQ6:	RWD		;REWIND.	(28)
3832	025470	000001		1			
3833	025472	000001		1			
3834	025474	000000		0			
3835	025476	104005		WRT		;WRITE.	(29)
3836	025500	004000		DATCNT			
3837	025502	000001		1			
3838	025504	000001		1			
3839	025506	105005		WRR		;WRITE RETRY.	(30)
3840	025510	004000		DATCNT			
3841	025512	000001		1			
3842	025514	000001		1			
3843	025516	100011		WTM		;WRITE TAPE MARK.	
3844	025520	000001		1			
3845	025522	000001		1			
3846	025524	000000		0			
3847	025526	105410		SFR		;SPACE 1 FILE REV.	
3848	025530	000001		1			
3849	025532	000001		1			
3850	025534	000000		0			
3851	025536	177777	.WORD	END			

TEST 1: BASIC FUNCTIONS.

```

3852
3853 025540 104401      BFSEQ7:      RDR      ;READ REV.      (31)
3854 025542 004000      DATCNT
3855 025544 000001      1
3856 025546 000001      1
3857 025550 105401      RNR      ;READ NEXT REV.  (32)
3858 025552 004000      DATCNT
3859 025554 000001      1
3860 025556 000001      1
3861 025560 125401      RNF      ;READ NEXT FWD.  (33)
3862 025562 004000      DATCNT
3863 025564 000001      1
3864 025566 000001      1
3865 025570 177777      .WORD    END
3866
3867 025572 104001      BFSEQ8:      RDF      ;READ FWD.      (34)
3868 025574 004000      DATCNT
3869 025576 000001      1
3870 025600 000001      1
3871 025602 105001      RPF      ;READ PREVIOUS FWD. (35)
3872 025604 004000      DATCNT
3873 025606 000001      1
3874 025610 000001      1
3875 025612 125001      RPR      ;READ PREVIOUS REV. (36)
3876 025614 004000      DATCNT
3877 025616 000001      1
3878 025620 000001      1
3879 025622 177777      .WORD    END
3880
3881 025624 101012      BFSEQ9: .WORD CLN      ;CLEAN.      (37)
3882 025626 000001      1
3883 025630 000001      1
3884 025632 000000      0
3885 025634 102010      RWD      ;REWIND      (38)
3886 025636 000001      1
3887 025640 000001      1
3888 025642 000000      0
3889 025644 177777      .WORD    END      ;END OF SEQUENCE.
3890
3891 025646 104105      BFSEQ10:     WTV      ;WRITE/VERIFY EVEN LENGTH. (39)
3892 025650 000012      12
3893 025652 000001      1
3894 025654 000000      0
3895 025656 104105      WTV      ;WRITE/VERIFY ODD LENGTH. (40)
3896 025660 000011      11
3897 025662 000001      1
3898 025664 000000      0
3899 025666 177777      .WORD    END
3900
3901
3902 025670      ENDTST
      025670      L10017:
      025670 104401      TRAP      C$ETST
3903
3904 .SBTTL TEST 2: DATA RELIABILITY.
3905
3906 ;++

```

TEST 2: DATA RELIABILITY.

```

3907 ; TEST TO CHECK THE DATA RELIABILITY OF THE TS05.
3908
3909 025672          BGNTST
      025672      T2::
3910
3911 025672 112737 000001 003515      MOVB   #1,RANDOM      ;SET THE RANDOM OPERATIONS FLAG.
3912 025700 105037 003514              CLRB   EXPBOT        ;CLEAR EXPECT BOT FLAG.
3913 025704 005037 003456              CLR    WTMFLG        ;CLEAR WRITE TAPE MARK FLAG
3914 025710 004737 017102              JSR    PC,FIRSTU     ;FIND THE FIRST UNIT.
3915 025714 004737 007072              JSR    PC,SOFINIT   ;INIT DEVICE
3916 025720 103404                      BCS    11$
3917 025722          ERRDF 2,NSSRM,STAERM ;REPORT TS05 NOT READY
      025722 104455                      TRAP   C$ERRDF
      025724 000002                      .WORD 2
      025726 004536                      .WORD NSSRM
      025730 006120                      .WORD STAERM
3918
3919 025732 004737 007466      11$:   JSR    PC,M0SET      ;GO DO SETUP'S
3920 025736 012702 004000          MOV    #DATCNT,R2    ;SET UP THE RECORD LENGTH MASK,
3921 025742 005302                      DEC    R2
3922 025744 010237 003430          MOV    R2,LENMSK    ;ALLOW MAXIMUM BUFFER.
3923 025750 005137 003430          COM   LENMSK
3924 025754 004737 010162          JSR   PC,SETCH      ;CMD 1 = SET CHARACTERISTIC.
3925 025760 105737 003526          TSTB  STAFLG ;IFB STAFLG NE #0 THEN ;IF STARTING THEN:
3926 025764 001417          BEQ   50424$
3927 025766 004737 010206          JSR   PC,SETRW     ;CMD2=REWIND
3928 025772 105037 003526          CLRB  STAFLG ;LET STAFLG :B= #0 ;CLR START FLAG.
3929
3930 025776          50422$:
3931 025776 012721 104105          MOV   #WTV,(R1)+
3932 026002 012721 004000          MOV   #DATCNT,(R1)+
3933 026006 012702 177740          MOV   #RNOPSC,R2
3934 026012 005102          COM   R2
3935 026014 010221          MOV   R2,(R1)+
3936 026016 012721 000007          MOV   #RANP,(R1)+
3937
3938 026022          50423$: BREAK ; DO A SUPER BREAK FIRST.
      026022 104422                      TRAP   C$BRK
3939
3940 026024          50424$: ;FILL SEQ TBL WITH RANDOM CMDS.
3941 026024 020127 003740          CMP   R1,#SEQEND
3942 026030 002012          BGE   50425$
3943 026032 063737 003432 003434          ADD  RANB,RANS ;LET RANS := RANS + RANB
3944 026040 013702 003434          MOV  RANS,R2
3945 026044 042702 177741          BIC  #177741,R2
3946 026050 004772 026206          JSR  PC,#RANCMD(R2) ;SET UP A RANDOM CMD + BRP.
3947
3948 026054 000763          BR   50424$
3949 026056          50425$:
3950 026056 012711 177777          MOV  #END,(R1) ;STORE END OF SEQUENCE CODE IN TABLF.
3951 026062 004737 010226          JSR  PC,EXALL ;GO EXECUTE ALL CMDS IN SEQUENCE TABLE.
3952
3953 026066 012701 003540          MOV  #CMDSEQ,R1 ;INIT CMD SEQ TBL POINTER,
3954 026072 005702          TST  R2 ;REPEAT UNTIL EOT IS REACHED
3955 026074 001752          BEQ  50423$
3956 026076 105237 003524          INCB ALLEOT ;FLAG ALL UNITS @ EOT
3957 026102 000240          NOP

```

TEST 2: DATA RELIABILITY.

```

3958 026104 000240      NOP
3959 026106 000240      NOP
3960 026110 004737 027662 JSR    PC,T5WEOT      ;WRITE ONE RECORD BEYOND EOT ON ALL UNITS
3961                                     ;SO THAT SHORTER READ STOP DISTANCE
3962                                     ;SHALL POSITION HEAD IN CLEAN IRG GAP
3963                                     ;READ REV THAT EXTRA REC TO RE-POSITION THE TAPE
3964 026114 004737 026246 JSR    PC,RANRD      ;SET UP READ REV/FWD CMDS,
3965 026120 012737 177740 003544 MOV    #RNOPSC,CMDSEQ+4 ;# OF RECORDS FOR READ REV.
3966 026126 005137 003544 COM    CMDSEQ+4
3967 026132 013737 003544 003554 MOV    CMDSEQ+4,CMDSEQ+14 ;# OF RECORDS FOR READ FORWARD.
3968 026140 012711 177777 MOV    #END,(R1)      ;STORE END OF SEQUENCE CODE IN SEQ TABLE.
3969 026144 004737 010226 JSR    PC,EXALL      ;GO EXECUTE READ REV/FWD OF LAST N RECORDS.
3970 026150 105037 003524 CLR    ALLEOT        ;CLEAR ALL UNITS @ EOT FLAG
3971 026154 112737 000001 003517 MOVB  #1,RPTFLG      ;REQUEST PERFORMANCE REPORT DURING REWIND.
3972 026162 012701 003540 MOV    #CMDSEQ,R1    ;INIT SEQ TBL POINTER,
3973 026166 004737 010206 JSR    PC,SETRW      ;STORE REWIND IN SEQ TBL,
3974 026172 012711 177777 MOV    #END,(R1)      ;STORE END IN SEQ TBL.
3975 026176 004737 010226 JSR    PC,EXALL      ;EXECUTE REWIND CMD ON ALL UNITS
3976
3977 026202      EXIT    TST
      026202 104432      TRAP    C$EXIT
      026204 000320      .WORD   L10034-.

3978
3979      ;      ADDRESSES OF SUBROUTINES USED TO SET UP RANDOM OPERATIONS IN
3980      ;      THE DATA RELIABILITY TEST.
3981
3982 026206 026360 RANCMD: RANWR      ;WRITE
3983 026210 026404 RANWR      ;WRITE.
3984 026212 026360 RANWR      ;WRITE.
3985 026214 026360 RANWR      ;WRITE.
3986 026216 026360 RANWR      ;WRITE.
3987 026220 026360 RANWR      ;WRITE.
3988 026222 026360 RANWR      ;WRITE.
3989 026224 026360 RANWR      ;WRITE.
3990 026226 026246 RANRD      ;READ.
3991 026230 026246 RANRD      ;READ.
3992 026232 026246 RANRD      ;READ.
3993 026234 026246 RANRD      ;READ.
3994 026236 026246 RANRD      ;READ.
3995 026240 026246 RANRD      ;READ.
3996 026242 026246 RANRD      ;READ.
3997 026244 026246 RANRD      ;READ.
3998
3999
4000      ;      SUBROUTINE TO SET UP READ COMMANDS IN SEQUENCE TABLE.
4001      ;      INPUTS:
4002      ;      OUTPUTS:
4003      ;      REGISTERS:      R2
4004      ;      CALLS:
4005
4006 026246 005737 003456 RANRD:: TST    WTMFLG      ;WAS LAST CMD A WRITE?
4007 026252 001406 BEQ    1$              ;NO,GO AHEAD
4008 026254 004737 026416 JSR    PC,RAWTM      ;YES PUT DOWN TAPE MARK
4009 026260 004737 026444 JSR    PC,RASFR      ;AND SPACE FILE REV
4010 026264 005037 003456 CLR    WTMFLG        ;THEN CLEAR THE FLAG
4011 026270 020127 003740 1$:  CMP    R1,#SEQEND
4012 026274 002030 BGE    2$

```

TEST 2: DATA RELIABILITY.

```

4070      ;      INPUTS:
4071      ;      OUTPUTS:
4072      ;      REGISTERS:
4073      ;      CALLS:
4074
4075 026444 020127 003740      RASFR:: CMP      R1,#SEQEND
4076 026450 002007              BGE      1$
4077 026452 012721 105410      MOV      #SFR,(R1)+      ;STORE SPACE FILE REVERSE
4078 026456 012721 000001      MOV      #1,(R1)+      ;BRF
4079 026462 012721 000001      MOV      #1,(R1)+      ;# OF OPERATIONS
4080 026466 005721              TST      (R1)+      ;SKIP PATTERNS
4081 026470 000207              1$:      RTS PC
4082
4083
4084      ;      SUBROUTINE TO STORE BR# # OF OPERATIONS, PATTERN IN COMMAND
4085      ;      SEQUENCE TABLE FOR WRITE AND WRITE/VERIFY COMMANDS.
4086      ;      INPUTS:
4087      ;      OUTPUTS:
4088      ;      REGISTERS:      R2
4089      ;      CALLS:
4090
4091 026472 012721 004000      RANW:: MOV      #DATCNT,(R1)+      ;SET BR# TO MAX FOR PATTERN GENERATION.
4092                                ;RANDOM BR# WILL BE GENERATED FOR EACH RECORD.
4093 026476 063737 003434 003432      ADD      RANS,RANB      ;LET RANB := RANB + RANS
4094 026504 013702 003432              MOV      RANB,R2      ;LET R2 := RANB CLR.BY #RNOPSC
4095 026510 042702 177740              BIC      #RNOPSC,R2
4096 026514 010221              MOV      R2,(R1)+      ;SET RANDOM # OF OPERATIONS.
4097 026516 012721 000007      MOV      #RANP,(R1)+      ;RANDOM PATTERN.
4098 026522 000207              RTS PC      ;RETURN.
4099
4100      .EVEN
4101
4102 026524              ENDTST
4103 026524              L10034:
4104 026524 104401              TRAP      C$ETST
4105
4106      .SBTTL TEST 3: WRITE COMPATABILITY/WRITE UTILITY.
4107      ;++
4108      ; TEST TO WRITE RECORDS FROM BOT TO EOT.
4109      ;--
4110 026526              BGNTST
4111 026526              T3::
4112 026526 112737 000001 003515      MOVB     #1,RANDOM      ;SET THE RANDOM OPERATIONS FLAG.
4113 026534 105037 003514              CLRB     EXPBOT ;LET EXPBOT := #0      ;CLEAR EXPECT BOT FLAG.
4114
4115 026540 004737 017102              JSR      PC,FIRSTU      ;FIND THE FIRST UNIT.
4116 026544 004737 007072              JSR      PC,SOFINIT     ;INIT DEVICE
4117 026550 103404              BCS     11$
4118 026552              ERRDF 2,NSSRM,STAERM      ;REPORT TS05 NOT READY
4119      026552 104455              TRAP     C$ERDF
4120      026554 000002              .WORD   2
4121      026556 004536              .WORD   NSSRM
4122      026560 006120              .WORD   STAERM

```

TEST 3: WRITE COMPATABILITY/WRITE UTILITY.

```

4120 026562 004737 007466      11$: JSR    PC,MDSET          ;GO DO SETUP'S
4121 026566 012702 004000      MOV    #DATCNT,R2        ;SET UP THE RECORD LENGTH MASK.
4122 026572 005302             DEC    R2
4123 026574 010237 003430      MOV    R2,LENMSK        ;ALLOW MAXIMUM BUFFER.
4124 026600 005137 003430      COM    LENMSK
4125 026604 004737 010162      JSR PC,SETCH            ;CMD 1 = SET CHARACTERISTIC.
4126 026610 004737 010206      JSR PC,SETRW           ;CMD2=REWIND
4127 026614 105037 003526      CLRB  STAFLG ;LEI STAFLG ;B= #0 ;CLEAR START FLAG
4128 026620 50426$: BREAK          ; DO A SUPVSR BREAK FIRST.
                                TRAP    C$BRK
                                026620 104422
4129
4130 026622 50427$:             ;WHILE THERE IS MORE ROOM IN SEQ TABLE:
4131 026622 020127 003740      CMP    R1,#SEQEND
4132 026626 002003             BGE    50430$
4133 026630 004737 026360      JSR    PC,RANWR         ;STORE A WRITE CMD IN SEQUENCE TABLE.
4134 026634 000772             BR     50427$
4135 026636 50430$:
4136 026636 012711 177777      MOV    #END,(R1)        ;STORE END OF SEQUENCE CODE IN TABLE.
4137 026642 004737 010226      JSR    PC,EXALL         ;EXECUTE ALL CMDs IN SEQ TBL ON UNITS.
4138 026646 012701 003540      MOV    #CMDSEQ,R1      ;INIT SEQ TBL POINTER.
4139 026652 005702             TST    R2               ;REPEAT UNTIL EOT IS REACHED
4140 026654 001761             BEQ    50426$
4141 026656 105237 003524      INCB  ALLEOT           ;SET ALL UNITS @ EOT FLAG
4142 026662 000240             NOP
4143 026664 000240             NOP
4144 026666 000240             NOP
4145 026670 004737 027662      JSR    PC,TSWEOT       ;WRITE ONE RECORD BEYOND EOT ON ALL UNITS
4146                                     ;SO THAT SHORTER READ STOP DISTANCE
4147                                     ;SHALL POSITION HEAD IN CLEAN IRG GAP
4148                                     ;READ REV THAT EXTRA REC TO RE-POSITION TAPE
4149 026674 105037 003524      CLRB  ALLEOT           ;CLEAR ALL UNITS @ EOT FLAG
4150 026700 004737 010206      JSR PC,SETRW           ;STORE REWIND IN SEQ TBL.
4151 026704 012711 177777      MOV    #END,(R1)      ;STORE END IN SEQ TBL.
4152 026710 004737 010226      JSR PC,EXALL          ;EXECUTE REWIND CMD ON ALL UNITS
4153
4154
4155 026714             EXIT    TST
                                TRAP    C$EXIT
                                026714 104432             .WORD  L10035-.
                                026716 000002
4156
4157             .EVEN
4158
4159 026720             ENDTST
                                L10035:
                                TRAP    C$ETST
                                026720 104401
4160
4161             .SBTTL TEST 4: READ COMPATABILITY/READ UTILITY.
4162
4163             ;++
4164             ; TEST TO READ ENTIRE TAPE FORWARD AND REVERSE.
4165             ;--
4166
4167             BGNTST
4168 026722             T4::
                                026722
4169
4170 026722 112737 000001 003515      MOVB  #1,RANDOM        ;SET THE RANDOM OPERATIONS FLAG.

```

TEST 4: READ COMPATABILITY/READ UTILITY.

```

4171 026730 112737 000001 003514      MOVB    #1,EXPBOT      ;SET EXPECT BOT FLAG.
4172
4173 026736 004737 017102      JSR     PC,FIRSTU     ;FIND THE FIRST UNIT.
4174 026742 004737 007072      JSR     PC,SOFINIT   ;INIT DEVICE
4175 026746 103404
4176 026750      BCS     11$          ;REPORT TS05 NOT READY
      026750 104455
      026752 000002      TRAP    C$ERDF
      026754 004536      .WORD  2
      026756 006120      .WORD  NSSRM
      .WORD  STAERM
4177
4178 026760 004737 007466      11$:   JSR     PC,MDSET   ;GO DO SETUP'S
4179 026764 004737 010162      JSR     PC,SETCH     ;CMD 1 = SET CHARACTERISTIC.
4180 026770 004737 010206      JSR     PC,SETRW    ;CMD2=REWIND.
4181 026774 105037 003526      CLRB   STAFLG ;LET STAFLG :B= #0 ;CLEAR START FLAG
4182 027000 012721 104001      MOV    #RDF,(R1)+    ;CMD3 = READ FORWARD.
4183 027004 012721 004000      MOV    #DATCNT,(R1)+ ;SET LENGTH TO MAX FOR UNKNOWN LENGTHS.
4184 027010 012721 077777      MOV    #77777,(R1)+ ;SET RECORD COUNT TO MAX FOR WHOLE TAPE.
4185 027014 012721 000007      MOV    #RANP,(R1)+  ;PATTERN = RANDOM.
4186 027020 012711 177777      MOV    #END,(R1)    ;STORE END OF SEQUENCE CODE IN TABLE.
4187 027024 004737 010226      JSR     PC,EXALL    ;EXECUTE ALL CMD'S IN SEQ TBL ON ALL UNITS.
4188 027030 105237 003524      INCB   ALLEOT       ;FLAG TO ALLOW ALL UNITS AT EOT TO READ REV
4189 027034 012701 003540      MOV    #CMDSEQ,R1   ;INIT CMD SEQ TBL POINTER.
4190 027040 012721 104401      MOV    #RDR,(R1)+  ;CMD1 = READ REVERSE.
4191 027044 012721 004000      MOV    #DATCNT,(R1)+ ;SET LENGTH TO MAX FOR UNKNOWN LENGTHS.
4192 027050 012721 077777      MOV    #77777,(R1)+ ;RECORD COUNT = MAX FOR WHOLE TAPE.
4193 027054 012721 000007      MOV    #RANP,(R1)+  ;PATTERN = RANDOM.
4194 027060 012711 177777      MOV    #END,(R1)    ;STORE END OF SEQUENCE CODE IN TABLE.
4195 027064 004737 010226      JSR     PC,EXALL    ;GO EXECUTE READ REV. OF ENTIRE TAPE.
4196 027070 105037 003524      CLRB   ALLEOT       ;CLEAR ALL UNITS @ EOT FLAG
4197
4198 027074      EXIT    TST
      027074 104432      TRAP    C$EXIT
      027076 000002      .WORD  L10036-.
4199
4200      .EVEN
4201
4202 027100      ENDTST
      027100
      027100 104401      L10036: TRAP    C$ETST
4203
4204      .SBTTL TEST 5: EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.
4205
4206      ;++
4207      ; TEST TO EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.
4208      ;--
4209
4210 027102      BGNTST
      027102
4211      T5::
4212 027102 105037 003515      CLRB   RANDOM       ;CLEAR RAMDOM MODE FLAG.
4213 027106 112737 000001 003514      MOVB   #1,EXPBOT    ;SET EXPECT BOT FLAG.
4214
4215 027114 004737 017102      JSR     PC,FIRSTU   ;FIND THE FIRST UNIT.
4216 027120 004737 007072      JSR     PC,SOFINIT  ;INIT DEVICE
4217 027124 103404
4218 027126      BCS     11$          ;REPORT TS05 NOT READY
      ERRDF  2,NSSRM,STAERM

```

TEST 5: EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.

```

4264 027410          50433$: ;WHILE NCNT LT NCNT1 DO ;WHILE THERE ARE RECORDS REMAINING:
4265 027410 023737 003412 003414  CMP NCNT,NCNT1
4266 027416 002103          BGE 50434$
4267 027420 004737 011064          JSR PC,CMDAC ;STORE CMD ASCII IN ERROR MSG.
4268 027424 004737 010524          JSR PC,EXSUB ;ISSUE CMD TO ALL,AWAIT INTS,CHECK STATUS.
4269 027430 023727 003420 100017  CMP CMDWRD,#GES ;IF CMD IS GET STATUS THEN:
4270 027436 001002          BNE 50435$
4271 027440 004737 017412          JSR PC,PRXST ;PRINT EXTENDED STATUS REGISTERS.
4272
4273 027444          50435$:
4274 027444 004737 017500          JSR PC,CKHAE ;CHECK HALT AFTER EACH CMD FLAG.
4275 027450 012702 000001          MOV #1,R2 ;SET ALL UNITS AT BOT/EOT.
4276 027454 004737 017102          JSR PC,FIRSTU ;FIND FIRST UNIT.
4277 027460          50436$: ;WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE UNITS:
4278 027460 026527 002604 177777  CMP DEVTBL(R5),#END
4279 027466 001426          BEQ 50437$
4280 027470 032737 000400 003420  BIT #MOD.CO,CMDWRD ;IF CMD IS REVERSE THEN:
4281 027476 001406          BEQ 50440$
4282 027500 032765 000002 003502  BIT #XO.BOT,EOTFLG(R5) ;IF NOT AT BOT THEN:
4283 027506 001001          BNE 50441$
4284 027510 005002          CLR R2 ;CLEAR EOT/BOT FLAG.
4285
4286 027512          50441$:
4287 027512 000411          BR 50442$ ;ELSE IF CMD IS NOT REVERSE:
4288 027514          50440$:
4289 027514 032765 000001 003502  BIT #XO.EOT,EOTFLG(R5)
4290 027522 001404          BEQ 50443$
4291 027524 032737 000001 003420  BIT #CMD.CO,CMDWRD
4292 027532 001001          BNE 50444$
4293 027534          50443$:
4294
4295 027534 005002          CLR R2 ;IF NOT AT EOT OR NOT A MOTION CMD THEN:
4296 ;LET R2 := #0 ;CLEAR EOT/BOT FLAG.
4297 027536          50444$:
4298
4299 027536          50442$:
4300 027536 004737 017150          JSR PC,NEXTU ;FIND NEXT UNIT
4301
4302 027542 000746          BR 50436$
4303 027544          50437$:
4304 027544 020227 000001          CMP R2,#1 ;IF ALL UNIT ARE AT EOT/BOT THEN:
4305 027550 001016          BNE 50445$ ;FORCE TERMINATION OF COMMAND.
4306 027552 013737 003412 003414  MOV NCNT,NCNT1
4307 027560 005237 003414          INC NCNT1
4308 027564 105237 003524          INCB ALLEOT ;FLAG ALL UNITS AT EOT/BOT TO ALLOW VERIFY OF D
4309 027570 023727 003426 000002  CMP CMDLG,#2 ;WHEN WRITING IS CURRENT COMMAND
4310 027576 001002          BNE 50446$
4311 027600 004737 027662          JSR PC,T5WEOT ;GO WRITE/READ REV ONE RECORD BEYOND EOT
4312
4313 027604          50446$:
4314
4315 027604 000402          BR 50447$
4316 027606          50445$:
4317 027606 105037 003524          CLRB ALLEOT ;WHEN NOT ALL #EOT, CLEAR FLAG
4318
4319 027612          50447$:
4320 027612 005237 003412          INC NCNT ;UPDATE RECORD COUNT.

```

TEST 5: EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.

```

4321 027616 013737 003420 003424      MOV    CMDWRD,PCMDWD      ;SAVE PREVIOUS COMMAND WORD.
4322
4323 027624 000671                    BR     504331
4324 027626                    504341:
4325 027626 004737 016066              JSR    PC,VFYDAT        ;IF LAST CMD WAS A WRITE VERIFY, THEN GO
4326                                     ;VERIFY THE LAST N RECORDS OF DATA.
4327
4328 027632 000601                    BR     504311
4329 027634                    504321:
4330
4331 027654                                EXIT    1ST
      027635 104432                                TRAP    384707
      027636 000140                                .WORD  L100370.

4332
4333      ;      SUBROUTINE TO MOVE A COMMAND FROM THE SOFTWARE P TABLE TO
4334      ;      THE COMMAND SEQUENCE TABLE.
4335      ;      INPUTS:          R2 = POINTER TO SOFTWARE P TABLE
4336      ;      OUTPUTS:
4337      ;      REGISTERS:      R3.
4338      ;      CALLS:
4339
4340 027640 012203      PTCMDS:;MOV    (R2)+,R3          ;R3 = COMMAND TABLE INDEX.
4341 027642 005303      DEC    R3
4342 027644 006303      ASL    R3
4343 027646 016321 003752      MOV    CMDTBL(R3),(R1)+      ;MOVE COMMAND WORD.
4344 027652 012221      MOV    (R2)+,(R1)+          ;MOVE # OF BYTES.
4345 027654 012221      MOV    (R2)+,(R1)+          ;MOVE # OF OPERATIONS.
4346 027656 012221      MOV    (R2)+,(R1)+          ;MOVE PATTERN CODE.
4347 027660 000207      RTS    PC
4348
4349      ;      SUBROUTINE TO WRITE THEN READ REVERSE ONE RECORD BEYOND EOT
4350      ;      INPUTS:
4351      ;      OUTPUTS:
4352      ;      REGISTERS:
4353      ;      CALLS:          CMDAC,EXSUB,CKHAE
4354
4355 027662 000240      TSWEOT:;NOP
4356 027664 000240      NOP
4357 027666 004737 010524      JSR    PC,EXSUB          ;WRITE ONE RECORD BEYOND EOT
4358 027672 004737 017500      JSR    PC,CKHAE          ;SO THAT READ SHORTER STOP DISTANCE
4359                                     ;SHALL POSITION HEAD IN CLEAN IRG GAP
4360 027676 012700 000002      MOV    #2,R0              ;SET UP COUNTER FOR EOT
4361 027702 013737 003420 003424 11:  MOV    CMDWRD,PCMDWD      ;LET PCMDWD := CMDWRD ;REPOSITION TAPE
4362 027710 012737 104401 003420      MOV    #RDR,CMDWRD        ;LET CMDWRD := #RDR ;BEFORE EXTRA RECORD
4363 027716 012737 000004 003426      MOV    #4,CMDLG           ;BY READING REVERSE
4364 027724 013737 003420 002330      MOV    CMDWRD,CMDPKT      ;LET CMDPKT := CMDWRD CLR.BY #BRF.C
4365 027732 042737 004000 002330      BIC    #BRF.C,CMDPKT
4366 027740 013737 002330 003422      MOV    CMDPKT,CMDSAV      ;LET CMDSAV := CMDPKT ;THAT RECORD TO ALLOW
4367 027746 013737 003410 002332      MOV    DATARD,CMDPKT+CP.ADL ;NEXT COMMAND IN THE
4368 027754 004737 011064      JSR    PC,CMDAC           ;TABLE TO BE EXECUTED
4369 027760 004737 010524      JSR    PC,EXSUB
4370 027764 004737 017500      JSR    PC,CKHAE
4371 027770 005300      DEC    R0                ;FOUND EOT ?ET?
4372 027772 001343      BNE    11                ;NO,KEEP GOING
4373 027774 000207      RTS    PC                ;YES,RETURN
4374
4375      .EVEN

```

SYMBOL TABLE

SETRW	010206	G	TRAP4	023756	G	T\$LAST=	000001	T1	024204	G	WRTY	014636	G
SETUP	011172	G	TSAM	004705	G	T\$LOLI=	000000	T1SWB	003523	G	WRTYCT	003316	G
SFF	= 105010	G	TSBA	= 002514	G	T\$LSYM=	010000	T1.1	024214		WRTYER	003464	G
SFPTBL	002204	G	TSC.FC=	177717	G	T\$LTNO=	000005	T1.10	024524		WRTYFG	003463	G
SFR	= 105410	G	TSC.TC=	177761	G	T\$NEST=	177777	T1.11	024644		WRUNR	002776	G
SOFINI	007072	G	TSDB	002514	G	T\$NSO =	000000	T1.12	024730		WSM	= 140006	G
SRF	= 104010	G	TSHD	031305		T\$NS1 =	000005	T1.2	024420		WSMBK	002506	G
SRR	= 104410	G	TSNP	003534	G	T\$NS2 =	000002	T1.3	024444		WSSR	012700	G
STAERM	006120	G	TSSR	002524	G	T\$PCNT=	000000	T1.4	024464		WTBF	031355	
STAER1	006436		TSSREG	003454	G	T\$PTAB=	010043	T1.5	024504		WTBUF	002316	G
STAER2	006616		TSUNT	003532	G	T\$PTHV=	000001	T1.6	024524		WTM	= 100011	G
STAER3	006675		TSVCT	002534	G	T\$PTNU=	000001	T1.7	024544		WTMFLG	003456	G
STAER4	006733		TS.A16=	000400	G	T\$SAVL=	177777	T1.8	024564		WTR	= 101011	G
STAER5	006755		TS.A17=	001000	G	T\$SEGL=	177777	T1.9	024604		WTV	= 104105	G
STAER6	006562		TS.NBA=	002000	G	T\$SIZE=	000005	T2	025672	G	WTVRM	004430	G
STAER7	006530		TS.NXM=	004000	G	T\$SUBN=	000000	T3	026526	G	WTYBRF	015110	
STAF LG	003526	G	TS.OFL=	000100	G	T\$TAGL=	177777	T4	027722	G	WTYCMD	015104	
SVCGBL=	000000		TS.RMR=	010000	G	T\$TAGN=	010045	T5	027102	G	WTYWRD	015106	
SVCINS=	000001		TS.SC =	100000	G	T\$TEMP=	000000	TSWEOT	027662	G	X\$ALWA=	000000	
SVC SUB=	000000		TS.SPE=	020000	G	T\$TEST=	000005	UAM	= 000200	G	X\$FALS=	000040	
SVCTAG=	000000		TS.SSR=	000200	G	T\$TSTM=	177777	UNIWLK	005653		X\$OFFS=	000400	
SVCTST=	000000		TS.UPE=	040000	G	T\$TSTS=	000001	UNL	= 100412	G	X\$TRUE=	000020	
SWBFLG	003520	G	TS1MD	002312	G	T\$AU =	010016	UNREC	003470	G	X0.BOT=	000002	G
SWB.C	= 010000	G	TS5ADR	030040		T\$AUT=	010013	URERM	005041	G	X0.EOT=	000001	G
SWSET	004231	G	TS5CL	002564	G	T\$CLE=	010014	VFEXC	016152	G	X0.LET=	020000	G
S\$LSYM=	010000		TS5INT	002554	G	T\$DAT=	010044	VFISU	016400	G	X0.ONL=	000100	G
TCCRA	013414		TS5IN0	010034	G	T\$DU =	010015	VFYCNT	003346	G	X0.RLL=	010000	G
TCC0	013434	G	TS5IN1	010042	G	T\$HAR=	010040	VFYDAT	016066	G	X0.RLS=	040000	G
TCC1	013452	G	TS5IN2	010050	G	T\$HW =	010000	VFYFLG	003516	G	X0.TMK=	100000	G
TCC2	013470	G	TS5IN3	010056	G	T\$INI=	010012	VFY.C =	000100	G	X0.WLK=	000004	G
TCC3	013600	G	TS5SW	002574	G	T\$MSG=	010003	WAITF	007204	G	X2.BFE=	000100	G
TCC4	013616	G	TS5UNT	030064		T\$PC =	000001	WLKCHK	007356	G	X2.EFE=	000200	G
TCC5	014232	G	TS5VCT	030055		T\$PRO=	010011	WLKZRO	011734		X2.OPM=	100000	G
TCC6	014330	G	T\$ARGC=	000003		T\$PPTA=	010043	WRBC	002626	G	X3.DCK=	000010	G
TCC7	014472	G	T\$CODE=	001004		T\$RPT=	010010	WRECL	= 000020	G	X3.RNY=	157400	G
TC2RIN	013576		T\$ERRN=	000002		T\$SOF=	010041	WRR	= 105005	G	X4.HSS=	100000	G
TIME1	003436	G	T\$EXCP=	000000		T\$SRV=	010007	WRREC	002756	G	X4.RCE=	040000	G
TIME2	003440	G	T\$FLAG=	000041		T\$SUB=	010033	WRT	= 104005	G	ZROPAT	011704	
TOERM	004453	G	T\$FREE=	032016		T\$SW =	010001	WRTCHK	007270	G	\$LSTIN=	000001	
TOJMM	004727	G	T\$GMAN=	000000		T\$TES=	010037	WRTCHR	007436	G	\$LSTTA=	000001	
TRAPD4	003530	G	T\$HILI=	000010									

. ABS. 032016 000
000000 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 28665 WORDS (112 PAGES)
DYNAMIC MEMORY: 20060 WORDS (77 PAGES)
ELAPSED TIME: 00:05:29
CNTSEAO.BIC.CNTSEAO.SEQ/-SP=SVC34/ML,TSV1E,CNTSEA.MAC/EN:AMA:ABS/DS:GBL

CNTSEAO 1SV05 DATA R.B1
.....C1
.....D1
.....E1
.....F1
.....G1
.....H1
.....I1
.....J1
.....K1
.....L1
.....M1
.....N1

.....B2
.....C2
.....D2
.....E2
.....F2
.....G2
.....H2
.....I2
.....J2
.....K2
.....L2
.....M2
.....N2

.....B3
.....C3
.....D3
.....E3
.....F3
.....G3
.....H3
.....I3
.....J3
.....K3
.....L3
.....M3
.....N3

.....B4
.....C4
.....D4
.....E4
.....F4
.....G4
.....H4
.....I4
.....J4
.....K4
.....L4
.....M4
.....N4

.....B5
.....C5
.....D5
.....E5
.....F5
.....G5
.....H5
.....I5
.....J5
.....K5
.....L5
.....M5
.....N5

.....B6
.....C6
.....D6
.....E6
.....F6
.....G6
.....H6
.....I6
.....J6
.....K6
.....L6
.....M6
.....N6

.....B7
.....C7
.....D7
.....E7
.....F7
.....G7
.....H7
.....I7
.....J7
.....K7
.....L7
.....M7
.....N7

.....B8
.....C8
.....D8
.....E8
.....F8
.....G8
.....H8
.....I8
.....J8
.....K8
.....L8
.....M8
.....N8

.....B9
.....C9
.....D9
.....E9
.....F9
.....G9
.....H9
.....I9
.....J9
.....K9
.....L9
.....M9
.....N9

.....B10
.....C10
.....D10
.....E10
.....F10
.....G10
.....H10
.....I10
.....J10
.....K10
.....L10
.....M10
.....N10

.....B11
.....C11
.....D11
.....E11
.....F11
.....G11
.....H11
.....I11
.....J11
.....K11
.....L11