

TK25

TK25 DATA RELIAB TST
CZTKIAR0

COPYRIGHT (c) 1984
AH-T784A-MC
FICHE 01 OF 01

JUL 1984
digital
Made In USA

Table with multiple columns and rows of data, including headers like 'TEST NO.', 'TEST DATE', 'TEST TIME', 'TEST RESULT', and 'TEST STATUS'. The data is organized in a grid format.

1001 0001 1001
1001 0001 1001
1001 0001 1001

.REM &

IDENTIFICATION

PRODUCT CODE: AC-T783A-MC

PRODUCT NAME: CZTKIA0 TK25 DATA RELIABILITY TEST

PRODUCT DATE: 15 - MARCH - 1984

MAINTAINER: MAGTAPE DIAGNOSTIC ENGINEERING

AUTHOR: ROBERT F. WERY/JACK RICHARDSON/TERRENCE REILLY

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	

4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42

44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97

USER'S GUIDE

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 ALGORITHM
 1.1.4.3.2 OPERATIONAL WRITE-ERROR-RECOVERY ALGORITHM
 1.1.4.4 EARLY WARNING WRITE ERRORS
 1.1.4.5 DIAGNOSTIC TIMING ADJUSTMENT
 1.2 SYSTEM REQUIREMENTS
 1.2.1 HARDWARE REQUIREMENTS
 1.2.2 SOFTWARE REQUIREMENTS
 1.3 RELATED DOCUMENTS AND STANDARDS
 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
 1.5 ASSUMPTIONS
 1.6 DIAGNOSTIC HISTORY
 2.0 OPERATING INSTRUCTIONS
 2.1 HARDWARE PARAMETERS
 2.2 SOFTWARE PARAMETERS
 2.2.1 CLEAR COUNTERS
 2.2.2 RESET RANDOM VARIABLES
 2.2.3 PRINT SOFT ERRORS
 2.2.4 INHIBIT RECOVERY
 2.2.5 BAD TAPE SPOT DETECTION
 2.2.6 DISABLE INTERRUPTS
 2.2.7 INHIBIT RFC ERROR REPORTS
 2.2.8 CONTROLLER RAM DUMP
 2.2.9 ENABLE EARLY WARNING MESSAGES
 2.2.10 CHANGE CMD SEQUENCE
 2.2.11 COMMAND LIST FOR USE IN SOFTWARE DIALOGUE.
 2.2.12 DATA PATTERN LIST FOR USE IN SOFTWARE DIALOGUE
 2.3 EXAMPLES OF SOFTWARE DIALOGUE
 2.3.1 BASIC FUNCTION AND DATA RELIABILITY WITH ALL ERROR REPORTING ENABLED
 2.3.2 TO SET UP A SCOPE LOOP FOR A FAILURE IN BASIC FUNCTIONS.
 2.3.3 TO SET UP A SCOPE LOOP FOR A FAILURE IN DATA RELIABILITY
 2.4 EXECUTION TIMES
 2.4.1 SYSTEM CONFIGURATION
 2.4.2 TEST EXECUTION TIMES
 3.0 ERROR INFORMATION
 3.1 ERROR REPORTING

99		
100		
101		
102	3.1.1	ERROR 1 - COMMAND PACKET ADDRESS NOT ON A
103		MODULO 4 BOUNDARY:
104	3.1.2	ERROR 2 - TK25 NOT READY:
105	3.1.3	ERROR 3 - NO RESPONSE ERROR:
106	3.1.4	ERROR 4 - NO INTERRUPT ERROR:
107	3.1.5	SPECIAL CONDITION ERRORS:
108	3.1.5.1	ERROR 5 - TERMINATION CLASS CODE 0, UNDEFINED
109		SPECIAL CONDITION
110	3.1.5.2	ERROR 6 - TERMINATION CLASS CODE 1, ATTENTION
111		CONDITION
112	3.1.5.3	ERROR 7 - TERMINATION CLASS CODE 2, TAPE
113		STATUS ALERT
114	3.1.5.4	ERROR 8 - TERMINATION CLASS CODE 3, FUNCTION
115		REJECT
116	3.1.5.5	ERROR 9 - TERMINATION CLASS CODE 4,
117		RECOVERABLE ERROR
118	3.1.5.6	ERROR 10 - TERMINATION CLASS CODE 5,
119		RECOVERABLE ERROR
120	3.1.5.7	ERROR 11 - TERMINATION CLASS CODE 6,
121		UNRECOVERABLE ERROR
122	3.1.5.8	ERROR 12 - TERMINATION CLASS CODE 7, FATAL
123		SUBSYSTEM ERROR
124	3.1.6	ERROR 13 - RFC NON-ZERO ERROR:
125	3.1.7	ERROR 14 - RETRY LIMIT EXCEEDED:
126	3.1.8	ERROR 15 - TOO MANY INTERRUPTS:
127	3.1.9	ERROR 16 - CAPSTAN RUNAWAY:
128	3.1.10	ERROR 17 - DATA COMPARE ERROR:
129	3.2	ERROR HALTS
130	4.0	PERFORMANCE REPORT
131	5.0	TEST SUMMARIES
132	5.1	TEST 1 - BASIC FUNCTIONS.
133	5.2	TEST 2 - DATA RELIABILITY.
134	5.3	TEST 3 - WRITE AND READ STREAMING TEST.
135	5.4	TEST 4 - WRITE COMPATABILITY/WRITE UTILITY.
136	5.5	TEST 5 - READ COMPATABILITY/READ UTILITY.
137	5.6	TEST 6 - EXECUTE OPERATOR SELECTED COMMAND
138		SEQUENCE.
139	6.0	DEVICE INFORMATION TABLES
140	6.1	GENERAL
141	6.2	BUS INTERFACE SPECIFICATIONS
142	6.3	BIT DEFINITIONS FOR TK25 REGISTERS
143	6.3.1	TK25 REGISTER SUMMARY
144	6.3.2	TK25 STATUS REGISTER (TSSR)
145	6.3.2.1	TSSR READ ONLY
146	6.3.2.2	TSSR WRITE ONLY
147	6.3.3	EXTENDED STATUS REGISTER 0 (XSTAT0)
148	6.3.4	EXTENDED STATUS REGISTER 1 (XSTAT1)
149	6.3.5	EXTENDED STATUS REGISTER 2 (XSTAT2)
150	6.3.6	EXTENDED STATUS REGISTER 3 (XSTAT3)

152		
153		
154		
155	GLOSSARY	
156	-----	
157		
158	ACT	AUTOMATED COMPUTER TEST SYSTEM
159		
160	APT	AUTOMATED PRODUCT TEST SYSTEM
161		
162	BYTE/RECORD/FILE COUNT	IS STORED IN THE 4TH WORD OF THE COMMAND PACKET
163	BRF	AND IT'S USE BY THE TK25 DEPENDS ON THE TYPE OF
164		COMMAND.
165		
166	CMD	TK25 COMMAND
167		
168	COMMAND PACKET	FOUR WORD PACKET IN THE CPU MEMORY WHICH
169	CMDPKT	CONTAINS ALL INFORMATION NEEDED BY THE TK25 TO
170		EXECUTE A COMMAND.
171		
172	EXTENDED STATUS	FOUR WORDS OF TK25 STATUS WHICH ARE TRANSFERRED
173		AS PART OF THE MESSAGE PACKET AT THE COMPLETION
174		OF A COMMAND.
175		
176	MESSAGE PACKET	SEVEN WORD PACKET IN THE CPU MEMORY INTO WHICH
177		THE TK25 STORES STATUS AT THE COMPLETION OF A
178		COMMAND.
179		
180	PC	PROGRAM COUNTER
181		
182	PSW	PROCESSOR STATUS WORD
183		
184	RESIDUAL FRAME COUNT	THIS COUNT IS PART OF THE MESSAGE PACKET AND
185	RFC	CONTAINS THE NUMBER OF BYTES/RECORDS/FILES
186		REMAINING TO BE PROCESSED AT THE COMPLETION OF A
187		COMMAND.
188		
189	SPECIAL CONDITION	TSS4 BIT15. WHEN SET, INDICATES THAT THE LAST
190	SPEC COND	COMMAND DID NOT COMPLETE WITHOUT INCIDENT.
191		
192	TERMINATION CLASS CODE	THREE BIT CODE IN THE TSSR WHICH INDICATES THE
193	TCC	THE TYPE OF COMMAND TERMINATION.
194		
195	TSBA	TAPE SYSTEM BUS ADDRESS REGISTER.
196		
197	TSDB	TAPE SYSTEM DATA BUFFER REGISTER.
198		
199	TSSR	TAPE SYSTEM STATUS REGISTER.
200		
201	XST0	EXTENDED STATUS REGISTER 0
202		
203	XST1	EXTENDED STATUS REGISTER 1
204		
205	XST2	EXTENDED STATUS REGISTER 2
206		

208
209
210
211
212
213
214
215
216

XST3

EXTENDED STATUS REGISTER 3

XXDP.

XXDP. IS A "CATCH-ALL" NAME FOR A GROUP OF
PDP-11 DIAGNOSTIC PACKAGES AVAILABLE ON
MULTIMEDIA.

218
219
220
221
222
223
224
225
226
227
228
229
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

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

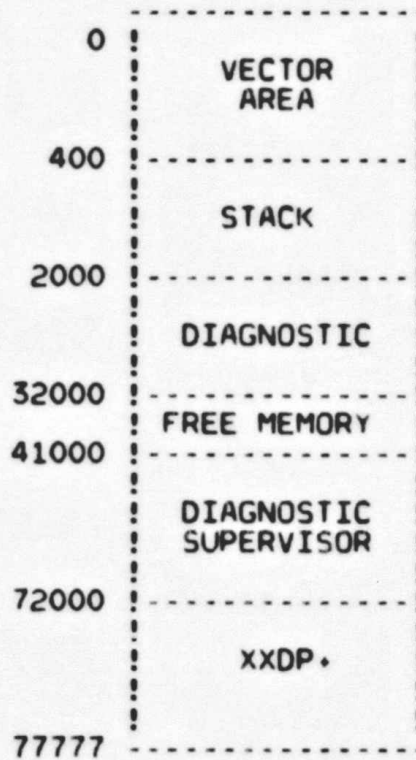
1.1.1 FUNCTIONAL DESCRIPTION -

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

1.1.2 STRUCTURE OF PROGRAM -

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

1.1.3 MEMORY MAP -



FREE MEMORY SPACE FOR THE WRITE/READ BUFFERS IS ALLOCATED BY THE SUPERVISOR ON REQUEST OR CHOSEN BY THE PROGRAMMER TO RESIDE BETWEEN THE DIAGNOSTIC AND THE SUPERVISOR.

274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325

1.1.4 DIAGNOSTIC INFORMATION -

1.1.4.1 SCOPE -

THIS DIAGNOSTIC CAN TEST UP TO FOUR (4) UNITS IN A ROUND ROBIN FASHION. THE FOUR UNITS ARE ASSIGNED LOGICAL UNIT NUMBERS 0 - 3 BY THE DIAGNOSTIC.

THERE ARE 6 TESTS IN THIS PROGRAM:

- TEST 1 - BASIC FUNCTIONS.
- TEST 2 - DATA RELIABILITY.
- TEST 3 - WRITE AND READ STREAMING TEST.
- TEST 4 - WRITE COMPATABILITY/WRITE UTILITY.
- TEST 5 - READ COMPATABILITY/READ UTILITY.
- TEST 6 - OPERATOR SELECTED SEQUENCE UTILITY.

1.1.4.2 ERROR RECOVERY -

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

1.1.4.3 WRITE ERROR RECOVERY -

THERE ARE 2 DISTINCT, SELECTABLE WRITE-ERROR-RECOVERY ALGORITHMS:

- 1. MEDIA/OPERATIONAL SELECTIVE ALGORITHM
- 2. OPERATIONAL ALGORITHM

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

TO SELECT THE SECOND ALGORITHM:

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

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

327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378

1.1.4.3.1 MEDIA/OPERATIONAL SELECTIVE WRITE ERROR RECOVERY
ALGORITHM -

SCOPE

THE ALGORITHM DISCERNS MEDIA RELATED WRITE ERRORS FROM
OPERATIONAL ONES.

ALGORITHM

A WRITE RETRY SUBROUTINE IS CALLED BY THE RECOVERABLE ERROR
SUBROUTINE UPON DETECTION OF A RECOVERABLE WRITE ERROR. THE
WRITE RETRY SUBROUTINE REWRITES THE RECORD IN THE SAME SPOT ON
TAPE FOUR TIMES. IF ALL 4 REPEATS ARE GOOD, THE RECORD IS
CONSIDERED RECOVERED AND A RECOVERABLE WRITE ERROR IS LOGGED AT
THAT RECORD NUMBER.

IF ANY OF THE 4 REPEATS FAIL, THE BAD RECORD IS ERASED, SUSPECTED
BAD SPOT AT THAT RECORD NUMBER IS LOGGED, AND THE RECORD IS
RETRIED AGAIN 3 INCHES FURTHER DOWN TAPE. THIS IS DONE UP TO 4
TIMES, UP TO 4 REPEATS EACH. IF THE RECORD CANNOT BE WRITTEN
WITHOUT RECOVERABLE ERRORS AFTER 4 RETRIES, THE RECORD IS ERASED
AND A BAD SPOT DETECTED ON RETRY FAILURE IS REPORTED. THE
RECOVERABLE ERROR SUBROUTINE THEN CONTINUES TO CALL THE WRITE
RETRY SUBROUTINE, WHICH REISSUES THE GROUP OF 4 RETRIES, UNTIL
THE RECORD IS RECOVERED OR 20 BAD SPOTS HAVE BEEN LOGGED.

TWO HUNDRED FIFTY (250) BAD SPOTS MAXIMUM ARE ALLOWED PER TAPE
PER PASS. WHEN 250 BAD SPOTS HAVE BEEN LOGGED, THE TAPE IS
CONSIDERED DEFECTIVE. A BAD TAPE OVERFLOW MESSAGE WILL BE
PRINTED AND THE UNIT IS REWOUND, THEN DROPPED.

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

RECORDS WHICH WERE NOT WRITTEN WITHOUT ERROR WILL BE ERASED.
THIS IS SO THAT ALL RECORDS LEFT ON TAPE ARE GOOD WRITTEN
RECORDS. BAD SPOTS ARE ERASED, WITH ERASE GAPS FROM 3 TO 12
INCHES PER RETRY GROUP.

IF NO BAD SPOTS WERE PREVIOUSLY DETECTED, UP TO 20 FEET OF ERASE
GAP COULD RESULT WHEN RETRYING TO RECOVER A SINGLE RECORD. THAT
LONG STRETCH OF BAD TAPE WOULD THEN BE FLAGGED WITH 20 BAD SPOTS
AT THE SAME RECORD NUMBER.

BAD SPOTS REPORTS

380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422

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

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

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

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

TO CLEAR CUMULATIVE COUNTS, ANSWER 'Y' TO: CLEAR COUNTERS (L) Y ? . THE BAD TAPE SPOTS COUNT IS CLEARED WHEN WRITING 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 ULTIMATELY ARE KNOWN TO BE 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.

424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475

EXAMPLE OF A TAPE PASS PRINT:

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

CZTKIA SFT ERR 00009 ON UNIT 00 TST 002 SUB 000 PC: 012100
 RECOVERABLE ERROR
 WRT CMD FAILED - UNIT 0 PASS: 1 RECORD:10210
 PREVIOUS CMD WAS WRT
 CMDPKT TSBA RFC TSSR TCC
 100205 002406 000000 100210 4
 026600
 000000
 004000
 XST0 XST1 XST2 XST3
 000350 000002 100010 000000
 RECOVERED ON RETRY # 1
 †C
 DR>PRI
 UNIT 0 PASS: 1 RECORD:10210
 BYTES WRITTEN 0,272,279,691
 BYTES READ REV 0,301,123,654
 BYTES READ REV 0,301,120,381

	WRT	RDR	RDF
RECOVERABLE ERRORS	1	0	0
UNRECOVERABLE ERRORS	0	0	0
WRITE RETRIES	3		

2 BAD SPOTS THIS TAPE PASS PRECEDING RECORD #:
 6 6
 SPEC COND HARD FATAL COMPARE
 2 0 0 0
 DR>

477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530

THIS EXAMPLE SHOWS THAT RECORD 6 RECOVERED ON 2ND RETRY GROUP. THE 2 BAD SPOTS RESIDE IN A 18 INCH ERASE GAP BETWEEN RECORDS 5 AND 6. RECORD 10210 RECOVERED ON 1ST RETRY OF 4 GOOD REPEATS. THREE WRITE GROUP RETRIES WERE ATTEMPTED, RESULTING IN ONE RECOVERABLE WRITE ERROR FROM RECORD 10210 AND TWO BAD SPOTS BETWEEN RECORDS 5 AND 6.

1.1.4.3.2 OPERATIONAL WRITE-ERROR-RECOVERY ALGORITHM -

WHEN THIS ALGORITHM IS SELECTED, THE TK25 WRITE RETRY COMMAND IS ISSUED UP TO 16 TIMES OR UNTIL THE RECORD IS RECOVERED. THE WRITE RETRY COMMAND CONSISTS OF A SPACE REVERSE OVER THE BAD RECORD, AN ERASE OF 3 INCHES OF TAPE AND A REWRITE OF THE RECORD. THAT COMPOSITE COMMAND DOES NOT DETECT BAD SPOTS ON TAPE. THEREFORE NO BAD TAPE SPOTS STATUS IS PRINTED.

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

1.1.4.4 EARLY WARNING WRITE ERRORS -

SINCE THE TK25 DRIVE RECORDS IN A SERPENTINE MANNER, THE TAPE CARTRIDGE HAS PHYSICAL MARKERS AT EACH END OF THE TAPE. THE TK25 CONTROLLER WILL NOT ALLOW THE WRITING OF DATA OVER THE AREA OF THESE "EARLY WARNING" MARKERS. THEREFORE, WHEN AN ATTEMPT IS MADE TO WRITE DATA AT THE END OF TRACK, A WRITE ERROR STATUS (TCC4) IS RETURNED WITH THE EARLY WARNING (EW) BIT SET.

WHEN A WRITE ERROR OCCURS AND THE EARLY WARNING BIT IS SET IN XSTAT1, THE ERROR IS RETRIED IF ERROR RECOVERY IS ENABLED. THE ONLY METHOD OF RETRY USED FOR EARLY WARNING WRITE ERRORS IS TO SET THE RETRY MODIFIER ON THE ORIGINAL COMMAND AND TO REISSUE IT. IN ADDITION, EARLY WARNING WRITE ERRORS ARE NOT COUNTED IN ANY ERROR TALLIES.

1.1.4.5 DIAGNOSTIC TIMING ADJUSTMENT -

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

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
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586

IF TIME-OUT OCCURS WHEN NO APPARENT MALFUNCTIONS IN THE TAPE UNIT ARE EVIDENT, THE TIMINGS OF THE DIAGNOSTIC MAY BE ADJUSTED. THIS IS ACCOMPLISHED BY PATCHING THE FIXED DELAY ELEMENT "L\$DLY".

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS -

- 0 PDP-11 PROCESSOR WITH 16K OR MORE OF MEMORY
- 0 CONSOLE DEVICE (LA36,LA120,VT100,ETC.)
- 0 PROGRAM LOAD DEVICE
- 0 TK25 DRIVE AND CONTROLLER

1.2.2 SOFTWARE REQUIREMENTS -

- 0 DIAGNOSTIC SUPERVISOR

1.3 RELATED DOCUMENTS AND STANDARDS

- 0 XXDP+ USERS MANUAL MD-11-CHQUS
- 0 PDP-11 DIAGNOSTIC SUPERVISOR INTERFACE SPECIFICATION
- 0 PDP-11 DIAGNOSTIC SUPERVISOR PROGRAMMER'S GUIDE
- 0 TK25 PROGRAMMING SPECIFICATION
- 0 TK25 COMMAND PACKET SPECIFICATION

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

ORDER OF HOST CPU DIAGNOSTIC USAGE:

- 1) FRONT END FUNCTIONAL PROGRAM - ALL TESTS.
- 2) DATA RELIABILITY PROGRAM:

588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642

- A) BASIC FUNCTION TEST.
- B) DATA RELIABILITY TEST.

1.5 ASSUMPTIONS

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

1.6 DIAGNOSTIC HISTORY

REVISION A - 15-MAR-84 - ORIGINAL RELEASE

2.0 OPERATING INSTRUCTIONS

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

2.1 HARDWARE PARAMETERS

ON A "N" RESPONSE TO "CHANGE HW?". THE DIAGNOSTIC SHALL RUN ASSUMING THAT THERE IS ONE UNIT AT TSSR = 172522 WITH A VECTOR = 224.

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

TSSR ADDRESS (172522) ?

VECTOR (224) ?

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

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

644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693

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 THE RESPONDING UNITS.

IF THE "ADR" FLAG IS NOT SET, THE READY AND OFF-LINE STATUS OF THE UNITS ARE CHECKED. A MESSAGE SHALL BE PRINTED EVERY SO OFTEN TO WARN THE OPERATOR OF UNITS THAT ARE NOT READY OR OFF-LINE. THESE UNITS SHALL BE DROPPED AFTER A REASONABLE AMOUNT OF TIME.

2.2 SOFTWARE PARAMETERS

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

CLEAR COUNTERS (L) Y ?
RESET RANDOM VARIABLES (L) N ?
HALT AFTER EACH CMD (L) N ?
PRINT SOFT ERRORS (L) N ?
INHIBIT RECOVERY (L) N ?
BAD TAPE SPOT DETECT (L) Y ?
DISABLE INTERRUPTS (L) N ?
INHIBIT RFC ERROR REPORTS (L) N ?
CONTROL FR RAM DUMP (L) N ?
ENABLE EARLY WARNING MESSAGES (L) N ?
CHANGE CMD SEQUENCE (L) N ?

2.2.1 CLEAR COUNTERS

IF YOU ANSWER YES TO THIS QUESTION, ALL COUNTERS (ERROR COUNTS, SPECIAL CONDITION COUNT, BYTES TRANSFERRED, ETC.) WILL BE SET TO ZERO.

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
742

2.2.2 RESET RANDOM VARIABLES -

IF YOU REQUEST THAT THE RANDOM VARIABLES BE RESET, THESE VARIABLES WILL TAKE ON THEIR DEFAULT VALUES.

2.2.3 PRINT SOFT ERRORS -

THIS QUESTION WILL ALLOW YOU TO ENABLE OR DISABLE THE PRINTING OF RECOVERABLE ERROR REPORTS. IF YOU ARE ONLY INTERESTED IN THE SUMMARY OF ERRORS AND NOT THE INDIVIDUAL ERRORS, YOU SHOULD ANSWER (N) TO THIS QUESTION.

2.2.4 INHIBIT RECOVERY -

IF YOU SO CHOOSE, YOU CAN INHIBIT ALL ERROR RECOVERY WITH THIS QUESTION. IF YOU ANSWER (Y) TO THIS QUESTION, THE FOLLOWING QUESTION WILL NOT BE ASKED.

2.2.5 BAD TAPE SPOT DETECTION -

IF YOU ANSWER (Y) TO THIS QUESTION, THE BAD TAPE SPOT DETECTION RETRY ALGORITHM WILL BE USED. OTHERWISE, THE OPERATIONAL WRITE ERROR RECOVERY ALGORITHM WILL BE USED FOR ERROR RECOVERY. THESE ALGORITHMS ARE DESCRIBED IN SECTIONS ABOVE. THIS QUESTION WILL NOT BE ASKED IF ERROR RECOVERY IS INHIBITED.

2.2.6 DISABLE INTERRUPTS -

YOU CAN DISABLE OR ENABLE CONTROLLER INTERRUPTS WITH THIS QUESTION.

2.2.7 INHIBIT RFC ERROR REPORTS -

THIS QUESTION ALLOWS YOU TO INHIBIT THE RFC (RESIDUAL FRAME COUNT) ERROR REPORTS. THESE ERROR REPORTS INDICATE THAT N BYTES/RECORDS/FILES WERE NOT PROCESSED AT THE END OF THE COMMAND.

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

2.2.8 CONTROLLER RAM DUMP -

THIS QUESTION ALLOWS YOU TO CHOOSE WHETHER A DUMP OF THE CONTROLLER'S RAM WILL OCCUR WITH EACH ERROR.

2.2.9 ENABLE EARLY WARNING MESSAGES -

THIS QUESTION ALLOWS YOU TO ENABLE OR DISABLE THE PRINTING OF A MESSAGE EACH TIME A WRITE ERROR OCCURS AND THE EARLY WARNING (EW) BIT IS SET IN EXTENDED STATUS REGISTER 1.

2.2.10 CHANGE CMD SEQUENCE -

THIS QUESTION WILL ALLOW YOU TO CHANGE THE COMMAND SEQUENCE USED IN TEST 6.

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

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
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816

CHARACTERISTICS CODE (0) 40 ?	(0,20,40,200)	(OCTAL)
CMD/2 (D) 13 ?	(1-27)	(DECIMAL)
BRF COUNT (D) 1 ?	(1-4K)	(DECIMAL)
# OF OPERATIONS (D) 1 ?	(1-32K)	(DECIMAL)
PATTERN (D) 7 ?	(0-8)	(DECIMAL)
CMD/3 (D) 4 ?	(1-27)	(DECIMAL)
BRF COUNT (D) 4096 ?	(1-4K)	(DECIMAL)
# OF OPERATIONS (D) 11719 ?	(1-32K)	(DECIMAL)
PATTERN (D) 7 ?	(0-8)	(DECIMAL)
CMD/4 (D) 7 ?	(1-27)	(DECIMAL)
BRF COUNT (D) 4096 ?	(1-4K)	(DECIMAL)
# OF OPERATIONS (D) 11719 ?	(1-32K)	(DECIMAL)
PATTERN (D) 7 ?	(0-8)	(DECIMAL)
CMD/5 (D) 2 ?	(1-27)	(DECIMAL)
BRF COUNT (D) 4096 ?	(1-4K)	(DECIMAL)
# OF OPERATIONS (D) 11719 ?	(1-32K)	(DECIMAL)
PATTERN (D) 7 ?	(0-8)	(DECIMAL)
CMD/6 (D) 13. ?	(1-27)	(DECIMAL)
BRF COUNT (D) 1 ?	(1-4K)	(DECIMAL)
# OF OPERATIONS (D) 1 ?	(1-32K)	(DECIMAL)
PATTERN (D) 7 ?	(0-8)	(DECIMAL)
CMD/7 (D) 27. ?	(1-27)	(DECIMAL)
BRF COUNT (D) 4096 ?	(1-4K)	(DECIMAL)
# OF OPERATIONS (D) 11719 ?	(1-32K)	(DECIMAL)
PATTERN (D) 7 ?	(0-8)	(DECIMAL)
CMD/8 (D) 27. ?	(1-27)	(DECIMAL)
BRF COUNT (D) 4096 ?	(1-4K)	(DECIMAL)
# OF OPERATIONS (D) 11719 ?	(1-32K)	(DECIMAL)
PATTERN (D) 7 ?	(0-8)	(DECIMAL)

NOTE: THE PROGRAM AUTOMATICALLY INSERTS A CHARACTERISTICS CODE 40 (SEE BELOW) AS THE FIRST COMMAND IN THE SEQUENCE TABLE. IF A DIFFERENT CHARACTERISTIC IS DESIRED, THE OPERATOR SHOULD ENTER THAT CHARACTERISTIC CODE.

A TOTAL OF 7 COMMANDS MAY BE ENTERED IN ADDITION TO THE SET CHARACTERISTICS COMMAND. IF THE OPERATOR WISHES TO USE LESS THAN 7 COMMANDS, AN END COMMAND (27) MUST BE ENTERED AND THEN A CONTROL Z (Z) CAN BE ENTERED TO TERMINATE SOFTWARE DIALOGUE.

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
856
857
858
859
860

2.2.11 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. (WRITE N RECORDS; SPACE REVERSE N RECORDS; 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 TK25 PROGRAMMING SPECIFICATION FOR DESCRIPTION.
24	NOT USED	
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 NUMBER OF OPERATIONS FIELD.
26	DLY	DELAY "N" MILLISECONDS WHERE N IS DEFINED IN THE NUMBER OF OPERATIONS. THIS DELAY IS USED BETWEEN EACH EXECUTABLE COMMAND.
27	END	END OF COMMAND SEQUENCE.

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

2.2.12 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. (125/25)
6	ALTERNATING BYTES OF 000 AND 377.
7	RANDOM DATA PATTERN.
8	NO PATTERN GENERATION.

2.3. EXAMPLES OF SOFTWARE DIALOGUE

2.3.1 BASIC FUNCTION AND DATA RELIABILITY WITH ALL ERROR REPORTING ENABLED -

- A) RECEIVE PROMPT (DR>)
- B) ENTER STA/TES:1-2<CR>
- C) ANSWER HARDWARE QUESTIONS.
- D) PROCEED WITH THE FOLLOWING DIALOGUE:
 - CHANGE SW (L) ? Y<CR>
 - CLEAR COUNTERS (L) N ? Y<CR>
 - RESET RANDOM VARIABLES (L) N ? N<CR>
 - HALT AFTER EACH CMD (L) N ? N<CR>
 - PRINT SOFT ERRORS (L) N ? Y<CR>
 - INHIBIT RECOVERY (L) N ? N<CR>
 - BAD TAPE SPOT DETECT (L) Y ? Y<CR>
 - DISABLE INTERRUPTS (L) N ? N<CR>
 - INHIBIT RFC ERROR REPORT (L) N ? N<CR>
 - CONTROLLER RAM DUMP (L) N ? N<CR>
 - ENABLE EARLY WARNING MESSAGES (L) N ? N<CR>
 - CHANGE CMD SEQUENCE (L) N ? N<CR>

904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958

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>
HALT AFTER EACH CMD (L) N ? N<CR>
PRINT SOFT ERRORS (L) N ? N<CR>
INHIBIT RECOVERY (L) N ? N<CR>
BAD TAPE SPOT DETECT (L) Y ? N<CR>
DISABLE INTERRUPTS (L) N ? N<CR>
INHIBIT RFC ERROR REPORT (L) N ? Y<CR>
CONTROLLER RAM DUMP (L) N ? N<CR>
ENABLE EARLY WARNING MESSAGES (L) N ? N<CR>
CHANGE CMD SEQUENCE (L) N ? N<CR>

```

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

- A) RECEIVE PROMPT (DR>)
- B) ENTER STA/TES:5/FLA:IER:ISR:IDU/EOP:1000<CR>
- C) ANSWER HARDWARE QUESTIONS.
- D) PROCEED WITH THE FOLLOWING DIALOGUE:

```

CHANGE SW (L) ? Y<CR>
CLEAR COUNTERS (L) N ? Y<CR>
RESET RANDOM VARIABLES (L) N ? N<CR>
HALT AFTER EACH CMD (L) N ? N<CR>
PRINT SOFT ERRORS (L) N ? N<CR>
INHIBIT RECOVERY (L) N ? N<CR>
BAD TAPE SPOT DETECT (L) Y ? N<CR>
DISABLE INTERRUPTS (L) N ? Y<CR>
INHIBIT RFC ERROR REPORT (L) N ? Y<CR>
CONTROLLER RAM DUMP (L) N ? N<CR>
ENABLE EARLY WARNING MESSAGES (L) N ? N<CR>
CHANGE CMD SEQUENCE (L) N ? Y<CR>
CHARACTERISTICS CODE (0) 40 ? 40<CR>
CMD/2 (D) 5 ? 13<CR> (REWIND) (COULD BE ANY COMMA
BRF COUNT (D) 4096 ? 1<CR>
# OF OPERATIONS (D) 10 ? 1<CR>
PATTERN (D) 7 ? 1<CR>
CMD/3 (D) 5 ? 4<CR> (WRITE) (COULD BE ANY COMMAN
BRF (D) 4096 ? 1000<CR>
# OF OPERATIONS (D) 10 ? 10000<CR>
PATTERN (D) 7 ? 1<CR>
CMD/4 (D) 5 ? 27<CR> (END) (COULD BE ANY COMMAN
BRF (D) 4096 ? <+Z>

```

960
961
962
963
964
965
966
967
968
969
970
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

2.4 EXECUTION TIMES

2.4.1 SYSTEM CONFIGURATION -

- 0 PDP-11/23+ PROCESSOR
- 0 128KW MEMORY
- 0 LA34 TERMINAL
- 0 1 - TK25 DRIVE
- 0 1 - TK25 Q-BUS CONTROLLER

2.4.2 TEST EXECUTION TIMES -

- TEST 1 - BASIC FUNCTIONS - 0.5 MINUTES PER PASS.
- TEST 2 - DATA RELIABILITY - 83 MINUTES PER PASS.
- TEST 3 - WRITE/READ STREAMING TEST - 53 MINUTES PER PASS.
- TEST 4 - WRITE COMPATABILITY - 34 MINUTES PER PASS.
- TEST 5 - READ COMPATABILITY - 23 MINUTES PER PASS.
- TEST 6 - OPERATOR SELECTED SEQUENCE - 48 MINUTES (FOR DEFAULT SEQUENCE).

NOTE

ALL EXECUTION TIMES ARE SHOWN FOR ONE UNIT OPERATION USING A 600 FOOT CARTRIDGE AND THE SYSTEM CONFIGURATION ABOVE. FOR SYSTEM CONFIGURATIONS OTHER THAN AN 11/23+ WITH 128KW OF MEMORY AND A SINGLE TK25 SUBSYSTEM, TEST DURATIONS WILL VARY.

3.0 ERROR INFORMATION

1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050

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, NUMBER OF RECORDS FROM BOT, RECORD READ COUNT, THE COMMAND PACKET, TSSR, TCC, TSBA, RFC, AND THE EXTENDED STATUS REGISTERS.

STANDARD ERROR REPORT FORMAT:

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

* CAUTION *

INTERPRET THE "RECORD READ" COUNT WITH CAUTION. IF IT IS VERY DIFFERENT FROM RECORD COUNT TRACKED BY THE DIAGNOSTIC, POSITION IS NOT NECESSARELY LOST. ERRORS IN READING THAT RECORD MIGHT HAVE CAUSED THE RECORD COUNT TO BE ERRONEOUSLY READ FROM TAPE.

FOR EXAMPLE, IF IN TEST 2 THE DIAGNOSTIC IS RESTARTED OR CONTINUED, THE RECORD COUNT IS RESET TO ZERO ALTHOUGH TAPE WAS NOT REWOUND. THIS IS NECESSARY BECAUSE THERE IS NO ACCURATE WAY TO DETERMINE ON WHAT RECORD COUNT OF WHAT UNIT THE DIAGNOSTIC WAS HALTED BEFORE RESTARTING OR CONTINUING. IT IS SUGGESTED THAT A "PRINT" BE REQUESTED WHEN HALTING THE DIAGNOSTIC TO GET A PRINT OF THE RECORD COUNT WHEN HALTED.

1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105

EXAMPLE OF AN ERROR REPORT:

```

CZTKIA SFT ERR 00009 TST 002 SUB 000 PC: 010606
RECOVERABLE ERROR
WRT CMD FAILED - UNIT 2 PASS: 2 RECORD: 254
PREVIOUS CMD WAS WRT
CNDPKT TSBA RFC TSSR TCC
100005 002324 000000 100210 4
051766
000000
000371
XST0 XST1 XST2 XST3
000350 000002 100004 000000

```

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 - TK25 NOT READY: -

BEFORE ANY COMMAND IS ISSUED TO THE TK25, THE SUBSYSTEM READY BIT IN THE TSSR4 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 TK25 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 WAS 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.

1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154

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.

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.

3.1.5.4 ERROR 8 - TERMINATION CLASS CODE 3, FUNCTION REJECT -

THE SPECIFIED FUNCTION WAS NOT INITIATED. BITS OF INTEREST INCLUDE 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.

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
1200
1201
1202
1203
1204
1205

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 THE RETRY LIMIT IS REACHED BEFORE THE ERROR IS RECOVERED, A "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 THE RETRY LIMIT IS REACHED BEFORE THE ERROR IS RECOVERED, A "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. 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 PROCEEDS NORMALLY. THE REPORTING AND LOGGING OF THESE ERRORS IS OPTIONAL.

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

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. 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 NUMBER 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.

1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308

4.0 PERFORMANCE REPORT

```

UNIT X   PASS:XXXXX  RECORD:XXXXX
BYTES WRITTEN  XXX,XXX,XXX,XXX
BYTES READ REV XXX,XXX,XXX,XXX
BYTES READ FWD XXX,XXX,XXX,XXX

RECOVERABLE ERRORS      WRT      RDR      RDF
UNRECOVERABLE ERRORS   XXXXX    XXXXX    XXXXX

SPEC COND  HARD  FATAL  COMPARE
          XXXXX XXXXX XXXXX XXXXX

```

5.0 TEST SUMMARIES

5.1 TEST 1 - BASIC FUNCTIONS.

EXECUTES AND VERIFIES CORRECT COMPLETION OF ALL TK25 FUNCTIONS.

- SUBTEST 1 - SET CHAR, DRIVE INIT, GET STATUS.
 - SET CHARACTERISTIC 200. (ENABLES SKIP TAPE MARKS STOP)
 - DRIVE INITIATE.
 - SET CHARACTERISTIC 20. (ENABLES MESSAGE BUFF RELEASE INTERRUPTS)
 - GET STATUS
 - SET CHARACTERISTIC 40. (ENABLES ATTENTION INTERRUPTS)
- 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.

1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362

- 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.
- 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.

1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414

5.2 TEST 2 - DATA RELIABILITY.

1. THE TAPE IS INITIATED WITH THE FOLLOWING COMMANDS:
 SET CHARACTERISTIC 40
 REWIND
 WRITE 31 RECORDS OF RANDOM LENGTH AND DATA
2. WRITE, REPOSITION RECORD REVERSE, 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 AND READ STREAMING TEST.

*** NOTE: THE TAPE LENGTH MUST BE 600 FEET FOR THIS TEST ***

1. REWINDS ALL UNITS, THEN ON EACH UNIT:

2. WRITE PATTERN 5 FOR 11719 - 4096 BYTE RECORDS.
3. SPACE REVERSE FOR 11719 RECORDS.
4. READ FORWARD FOR 11719 RECORDS.

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
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462

NOTE

WRITE AND READ ITERATIONS ARE INTERRUPTED SO THAT THE TAPE DOES NOT CONTINUOUSLY STREAM. THREE RECORDS ARE ALLOWED TO STREAM BEFORE THE UNIT IS INTERRUPTED FOR BOTH WRITE AND READ OPERATIONS. THE INTERRUPTION SCHEME IS SET UP TO PERMIT STREAMING OPERATIONS 75% OF THE TIME THIS TEST IS IN EXECUTION.

THIS TEST EXECUTES IN A SINGLE SERVER MANNER - ONLY ONE UNIT AT A TIME IS EVER IN OPERATION. FOR A FOUR (4) UNIT CONFIGURATION, THIS MEANS THAT THE DUTY CYCLE OF THE SYSTEM WILL BE 25%, BUT THE STREAMING OPERATION OF THE SPECIFIC UNIT UNDER TEST WILL BE 75%. FOR A THREE (3) UNIT CONFIGURATION THE SYSTEM DUTY CYCLE WILL BE 33.3%, AND A TWO UNIT CONFIGURATION WILL BE 50%. THE STREAMING DUTY CYCLE FOR ANY PARTICULAR UNIT UNDER TEST WILL ALWAYS BE 75% REGARDLESS OF THE SYSTEM DUTY CYCLE.

5.4 TEST 4 - WRITE COMPATABILITY/WRITE UTILITY.

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

5.5 TEST 5 - READ COMPATABILITY/READ UTILITY.

REWINDS AND READS ENTIRE TAPE IN THE FORWARD DIRECTION. REWIND.

1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517

5.6 TEST 6 - EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.

THE SEQUENCE OF COMMANDS ENTERED BY THE OPERATOR IS EXECUTED. IF NO COMMANDS WERE ENTERED, A DEFAULT SEQUENCE OF REWIND/WRITE/REWIND/READ FWD/REWIND OF THE ENTIRE TAPE IS EXECUTED WITH RANDOM PATTERN AND RECORD LENGTH OF 4096 BYTES.

6.0 DEVICE INFORMATION TABLES

6.1 GENERAL

THE TK25 SUBSYSTEM IS A 1/4 INCH CARTRIDGE TAPE DRIVE WITH EITHER A UNIBUS OR Q-BUS CONTROLLER. THE CONTROLLER MODULES USE THE TS11 PROTOCOL AS DEFINED IN THE TK25 SUBSYSTEM PROGRAMMERS GUIDE.

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 TK25 SUBSYSTEM TO FETCH THE WORD(S) WITHIN THE COMMAND PACKET. THE WORDS WITHIN THE COMMAND PACKET ARE:

- 1. COMMAND WORD
- 2. LOW ORDER BUFFER ADDRESS
- 3. HIGH ORDER BUFFER ADDRESS
- 4. BYTE COUNT

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.

EXTENDED STATUS REGISTERS ARE NOT READ DIRECTLY FROM DRIVE REGISTERS; RATHER, A "GET STATUS" COMMAND IS ISSUED WHICH WILL CAUSE THE TK25 TO TRANSFER EXTENDED STATUS INFORMATION TO THE MEMORY AREA POINTED TO BY THE BUFFER ADDRESS OF THE "GET STATUS" COMMAND. THERE ARE FOUR EXTENDED STATUS REGISTERS.

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.

COMMAND PACKETS MUST RESIDE ON DIVIDE BY FOUR MEMORY BOUNDARIES (AS OPPOSED TO DIVIDE BY 2 OR WORD BOUNDARIES).

1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546

6.2 BUS INTERFACE SPECIFICATIONS

THE STANDARD ADDRESSES AND VECTORS ARE ASSIGNED AS FOLLOWS:

TK25	REGISTER	Q-BUS		UNIBUS ADDRESS	VECTOR
		ADDRESS (I/O PAGE)	BBS7		
0	TSBA/TSDB	1	2 520	772 520	224
	TSSR	1	2 522	772 522	
1	TSBA/TSDB	1	2 524	772 524	*
	TSSR	1	2 526	772 526	
2	TSBA/TSDB	1	2 530	772 530	*
	TSSR	1	2 532	772 532	
3	TSBA/TSDB	1	2 534	772 534	*
	TSSR	1	2 536	772 536	

THE "*" INDICATES THAT THE VECTOR IS A FLOATING VECTOR WITH RANK OF 37. FLOATING VECTORS ARE ASSIGNED STARTING AT 300 WITH THE LOWER RANK NUMBERS BEING ASSIGNED FIRST. NOTE THAT THE FLOATING VECTORS MAY CHANGE FROM SYSTEM TO SYSTEM.

1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
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

6.3 BIT DEFINITIONS FOR TK25 REGISTERS

6.3.1 TK25 REGISTER SUMMARY -

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
TSBA)A15))A14))A13))A12))A11))A10))A09))A08))A07))A06))A05))A04))A03))A02))A01))A00)
TSDB)P15))P14))P13))P12))P11))P10))P09))P08))P07))P06))P05))P04))P03))P02))P17))P16)
TSSR)SC))))RMR))NXM))NBA))A17))A16))SSR))OFL))FC1))FC0))TC2))TC1))TC0))
XST0)TMK))RLS))LET))RLL))WLE))NEF))ILC))ILA))MOT))ONL))IE))VCK)))WLK))BOT))EOT)
XST1)DLT))))CRS))NER)))))TN3))TN2))TN1))TN0))EW)))UNC))
XST2)OPM))DCF))DHF))SPD))0))1)))) ERROR ADDRESS LST SIGNIFICANT BYTE)							
XST3) ERROR ADDRESS MOST SIGNIFICANT BYTE)))))))OPI))REV))TCH))STP))))RIB)

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

1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637

6.3.2 TK25 STATUS REGISTER (TSSR) -

6.3.2.1 TSSR READ ONLY -

```

      15  14  13  12  11  10  09  08  07  06  05  04  03  02  01  00
TSSR }SC } } } }RMR} }NXM}NBA}A17} }A16}SSR}OFL} }FC1}FC0}TC2} }TC1}TC0} }
      +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+ +---+

```

BIT	NAME	TERMINATION CLASS	DEFINITION
15	SC	S	SPECIAL CONDITION: USED TO INDICATE THAT EITHER AN ERROR OR AN EXCEPTION OCCURRED WHILE EXECUTING A COMMAND.
14	-	-	NOT USED
13	-	-	NOT USED
12	RMR	S	REGISTER MODIFICATION REFUSED: SSR (SUB-SYSTEM READY) WAS NOT SET WHEN A COMMAND POINTER WAS LOADED INTO TSDB.
11	NXM	4/5	NONEXISTENT MEMORY: WHEN AN ATTEMPT TO TRANSFER TO OR FROM A NONEXISTENT MEMORY LOCATION, THIS BIT WILL SET.
10	NBA	S	NEED BUFFER ADDRESS: INDICATES THAT THE MESSAGE BUFFER ADDRESS IS REQUIRED.
09	A17	S	BUS ADDRESS BIT 17: DISPLAYS VALUE OF BIT 17 OF THE TSBA REGISTER.
08	A16	S	BUS ADDRESS BIT 16: DISPLAYS VALUE OF BIT 16 OF THE TSBA REGISTER.

1639
1640
1641
1642
1643
1644
1645
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

BIT	NAME	TERMINATION CLASS	DEFINITION
07	SSR	S	SUBSYSTEM READY: THE SUBSYSTEM IS NOT BUSY AND IS READY TO ACCEPT A NEW COMMAND POINTER WHENEVER THIS BIT IS SET.
06	OFL	S	OFF LINE: INDICATES THE TAPE TRANSPORT IS OFF LINE AND UNAVAILABLE FOR ANY MOTION COMMANDS.
05	FC1	-	FATAL TERMINATION CLASS 1 - NOT USED
04	FC0	-	FATAL TERMINATION CLASS 0 - NOT USED
03	TC2	S	TERMINATION CLASS BIT 2: THIS BIT, ALONG WITH TC1 AND TCO, ACTS AS AN OFFSET VALUE WHENEVER AN ERROR OR EXCEPTION CONDITION OCCURS ON A COMMAND. EACH OF THE EIGHT POSSIBLE VALUES OF THIS FIELD REPRESENTS A PARTICULAR CLASS OF ERRORS WHICH HAVE A SIMILAR SIGNIFICANCE AND, AS APPLICABLE, SIMILAR RECOVERY PROCEDURES. THE CODE PROVIDED IN THIS FIELD IS EXPECTED TO BE USED AS AN OFFSET INTO A DISPATCH TABLE FOR HANDLING OF THE CONDITION.
02	TC1	S	TERMINATION CLASS BIT 1: SEE TC2
01	TC0	S	TERMINATION CLASS BIT 0: SEE TC2
00	-		NOT USED

1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713

6.3.2.2 TSSR WRITE ONLY -

WRITE COMMANDS TO THE TSSR DO NOT WRITE, BUT INVOKE CERTAIN SPECIALIZED FUNCTIONS.

1. A WRITE WORD (DATO) TO THE TSSR WILL INITIALIZE THE TK25 SUBSYSTEM AND REWIND THE TAPE ON THE CARTRIDGE.
2. A WRITE BYTE (DATOB) TO THE LOW BYTE OF THE TSSR WILL CAUSE THE CONTROLLER TO EXECUTE ITS RESIDENT SELF TESTS. IF THE CONTROLLER PASSES THE SELF TESTS, IT WILL THEN INITIALIZE ITSELF AND THE DRIVE AS ABOVE. THE TSDB/BA AND THE TSSR WILL NOT RESPOND TO BUS TRANSACTIONS FOR THE DURATION OF THE SELF TEST (APPROXIMATELY 100 MICROSECONDS). ANY ATTEMPT BY THE HOST TO READ OR WRITE THESE REGISTERS DURING SELF TEST WILL RESULT IN A NON-EXISTENT DEVICE REGISTER TIMEOUT.
3. IF AN OPERATION IS NOT IN PROGRESS (SSR SET IN THE TSSR), A WRITE BYTE TO THE HIGH BYTE OF THE TSSR WITH A "1" IN THE HIGH ORDER DATA BIT POSITION (BIT 7) WILL CAUSE THE SUBSYSTEM TO BOOT THE CPU BY MEANS OF THE FOLLOWING SEQUENCE OF EVENTS (Q-BUS CONTROLLER ONLY):
 - 0 REWIND THE TAPE
 - 0 SPACE FORWARD ONE RECORD
 - 0 READ THE FIRST 256 BITS OF THE SECOND RECORD INTO CPU MEMORY STARTING AT ADDRESS 0.

1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
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
1764
1765
1766
1767

6.3.3 EXTENDED STATUS REGISTER 0 (XSTATO) -

```

      15  14  13  12  11  10  09  08  07  06  05  04  03  02  01  00
XSTO)---)---)---)---)---)---)---)---)---)---)---)---)---)---)---)
      )---)---)---)---)---)---)---)---)---)---)---)---)---)---)---)

```

BIT	NAME	TERMINATION CLASS	DEFINITION
15	TMK	S/2	TAPE MARK DETECTED: SET WHENEVER A TAPE MARK IS DETECTED ON TAPE.
14	RLS	2	RECORD LENGTH SHORT: USED TO INDICATE ONE OF THREE CONDITIONS: 1. THE RECORD READ WAS SHORTER THAN THE BYTE COUNT 2. A SPACE RECORD OPERATION TERMINATED BEFORE THE POSITION COUNT WAS COMPLETED (THIS IS NORMAL IF A TAPE MARK IS DETECTED OR BOT IS ENCOUNTERED IN A SPACE REVERSE). 3. A SKIP TAPE MARKS COMMAND TERMINATED BEFORE THE POSITION COUNT WAS COMPLETED. THIS IS NORMAL IF A DOUBLE TAPE MARK (SEE LET) IS DETECTED OR BOT IS ENCOUNTERED IN A REVERSE OPERATION.
13	LET	2	LOGICAL END OF TAPE: SETS IF A TAPE MARK IS DETECTED WHEN MOVING FROM BOT OR IF DOUBLE TAPE MARKS ARE DETECTED. NOTE: THIS BIT WILL ONLY SET IF THE COMMAND IS SKIP TAPE MARKS AND THE MODE OF TERMINATION WAS ENABLED BY THE SET CHARACTERISTICS COMMAND.
12	RLL	2	RECORD LENGTH LONG: THE RECORD READ WAS LONGER THAN THE BYTE COUNT SPECIFIED.
11	WLE	3.(6)	WRITE LOCK ERROR: THE WRITE OPERATION WAS ISSUED TO A WRITE PROTECTED TAPE. NOTE: THE TK25 SETS THE TERMINATION CODE 3 ONLY. TC 6 APPLIES ONLY TO DRIVES EQUIPPED WITH A WRITE LOCK SWITCH
10	NEF	3	NON-EXECUTABLE FUNCTION: CAUSED TO SET BY ONE OF THE FOLLOWING: . REVERSE COMMAND WHEN ALREADY AT BOT

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

			. ANY MOTION COMMAND WITHOUT A CLEAR VOLUME CHECK WHILE THE TRANSPORT WAS OFF-LINE.
			. A WRITE COMMAND TO A WRITE PROTECTED TAPE.
09	ILC	3	ILLEGAL COMMAND: ANY COMMAND THAT CONTAINS CODES IN EITHER THE COMMAND FIELD OR MODE FIELD THAT ARE NOT SUPPORTED BY THE TAPE TRANSPORT.
08	ILA	3	ILLEGAL ADDRESS: WHEN THIS BIT IS SET, AN ILLEGAL ADDRESS IS ISSUED.
07	MOT	S	CAPSTAN MOVED: INDICATES TAPE WAS MOVED DURING AN OPERATION.
06	ONL	S/1/3	ON-LINE: INDICATES THAT THE SELECTED TRANSPORT IS ON-LINE AND READY. TERMINATION CLASS 1 IS SET FOR ATTN INTERRUPTS AND TC 3 FOR NON-EXECUTABLE FUNCTIONS REJECTED WHILE OFF LINE.
05	IE	S	INTERRUPT ENABLE:
04	VCK	S/3	VOLUME CHECK: ALWAYS SET AFTER INITIALIZATION OR WHEN THE ON-LINE STATUS OF THE TRANSPORT CHANGES. (EITHER ON TO OFF OR OFF TO ON)
03	PED	S	PHASE ENCODED DRIVE: TK25 IS NOT P.E.. SO THIS BIT IS ALWAYS ZERO.
02	WLK	S/3	WRITE LOCKED: THE TAPE IS WRITE PROTECTED.
01	BOT	S/2/3	BEGINNING OF TAPE: THE TAPE IS AT LOAD POINT, TC2 IS SET IF TAPE IS REVERSED INTO BOT, AND TC3 IF A REVERSE COMMAND IS ISSUED WITH THE TAPE ALREADY AT BOT.
00	EOT	S/2	END OF TAPE: SETS AS THE HEAD PASSES TO TRACK 10 IN THE FORWARD DIRECTION. IT IS NOT RESET UNTIL THE HEAD PASSES BACK TO TRACK 9 IN THE REVERSE DIRECTION. (STATUS ON READ; TC2 ON WRITE. SUBSYSTEM INIT ALSO RESETS THIS BIT.

1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874

6.3.4 EXTENDED STATUS REGISTER 1 (XSTAT1) -

BIT	NAME	TERMINATION CLASS	DEFINITION
15	DLT	4	DATA LATE: THIS BIT IS SET WHEN THE 16 BYTE FIFO BUFFER IS FULL ON A READ, OR EMPTY ON A WRITE, AND THE TAPE TRANSPORT REQUIRES A DATA TRANSFER.
14	-	-	NOT USED.
13	-	-	NOT USED.
12	CRS	4	CREASE DETECTED. DATA DROPPED OUT FOR UP TO 1.8 MS (APPROXIMATELY 0.2 INCH).
11	NER	4	NOISE DETECTED DURING ERASE.
10	-	-	NOT USED.
09	-	-	NOT USED.
08	-	-	NOT USED.
07	TN3	S	TAPE TRACK NUMBER HIGH ORDER BIT.
06	TN2	S	TAPE TRACK NUMBER BIT 2.
05	TN1	S	TAPE TRACK NUMBER BIT 1.
04	TN0	S	TAPE TRACK NUMBER LOW ORDER BIT.
03	EW	S/4	EARLY WARNING HOLE AT END OF TRACK SEEN.
02	-	-	NOT USED.
01	UNC	4	UNCORRECTABLE DATA: IN THE TK25 ALL TAPE ERRORS ARE UNCORRECTABLE, SINCE THERE IS NO INTERNAL ERROR CORRECTION.
00	-	-	NOT USED.

1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925

6.3.5 EXTENDED STATUS REGISTER 2 (XSTAT2) -

```

15  14  13  12  11  10  09  08  07  06  05  04  03  02  01  00
.....
)OPM) )DCF)DHF)SPD) ) 0 ) ) 1 ) ) ) ) ERROR ADDRESS LST SIGNIFICANT BYTE)
.....
SET CHARACTERISTICS COMMAND: )XFS)  CONTROLLER FIRMWARE VERSION )
.....

```

BIT	NAME	TERMINATION CLASS	DEFINITION
15	OPM	S	OPERATION IN PROGRESS: TAPE MOVED
14	DCF	7	DRIVE COMMUNICATION FAULT. FAILURE OF READ SENSE COMMAND TO TCS INITIATED AFTER DETECTING INT OR DER RESULTING FROM TAPE MOTION COMMAND TO TCS; OR HEALTH CHECK FAULT RECEIVED IN TCS SENSE BYTES.
13	DHF	7	DRIVE HARDWARE FAULT. NO LAMP CURRENT STATUS OR HEAD POSITIONING FAULT RETURNED IN TCS SENSE BYTES.
12	SPD	7	CAPSTAN SPEED ERROR, FAST OR SLOW
11	0	-	TK25 IDENTIFIER. ALWAYS ZERO.
10	-	-	NOT USED.
09	1	-	TK25 IDENTIFIER. ALWAYS ONE.
08	-	-	NOT USED.
07-00	EAD 07-00	S	ERROR ADDRESS LEAST SIGNIFICANT BYTE. (PROGRAM COUNTER IN THE TCD MICROCODE.)

NOTE: XSTAT 2 BITS 07 THRU 00 HAVE A DIFFERENT MEANING DURING THE SET CHARACTERISTICS COMMAND: XSTAT 2 BITS 06 THRU 00 RETURN THE MAJOR REVISION LEVEL OF THE CONTROLLER MICROCODE (IN BINARY). IF BIT 07 IS A 1, THE CONTROLLER ASSUMES A 22 BIT Q-BUS, AND IF 0, AN 18 BIT Q-BUS OR A UNIBUS. IN THE LATTER CASE, COMMAND PACKETS CONTAINING ADDRESSES GREATER THAN 18 BITS WILL GENERATE ERRORS.

1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973

6.3.6 EXTENDED STATUS REGISTER 3 (XSTAT3) -

```

15  14  13  12  11  10  09  08  07  06  05  04  03  02  01  00
)-----)
) ERROR ADDRESS MOST SIGNIFICANT BYTE ) )OPI) )REV)TCH)STP) ) ) )RIB)
)-----)

```

BIT	NAME	TERMINATION CLASS	DEFINITION
15-08	EAD 15-08	5	ERROR ADDRESS MOST SIGNIFICANT BYTE. IF XSTAT 1 BIT 14 IS 1, THE ADDRESS IS THE PROGRAM COUNTER IN THE SIA, AND IF 0 IN THE TCS.
07	-	-	NOT USED.
06	OPI	6	OPERATION INCOMPLETE: SETS IF 16 FEET OF TAPE WITHOUT DATA PASSES THE READ HEAD ON READ, SPACE, OR SKIP OPERATIONS OR 4 FEET OF TAPE IN WRITE OPERATIONS.
05	REV	5	REVERSE: INDICATES REVERSE IS THE DIRECTION OF CURRENT TAPE OPERATION.
04	TCH	7	NO TACHS: INDICATES THAT CAPSTAN MOTOR DID NOT START OR THAT TACHS ARE NOT BEING GENERATED OR DETECTED.
03	STP	S/6	STRIPE: SERVO STRIPE IS FAULTY OR MISSING.
02	-	-	NOT USED.
01	-	-	NOT USED.
00	RIB	2	REVERSE INTO BOT: SETS WHEN REVERSE OPERATIONS ENCOUNTER THE BOT EARLY WARNING HOLE AFTER TAPE IS IN MOTION.

ε

```

1986          .TITLE PROGRAM HEADER AND TABLES
1987          .SBTTL PROGRAM HEADER
2018
2020 000000          .ENABL ABS,AMA
2021          002000          "      2000
2023 002000          BGNMOD
2024
2025          ;**
2026          ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
2027          ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
2028          ;--
2029
2030 002000          POINTER BGNRPT,BGNSW,BGNSFT,BGNAU,BGNDU,BGNSETUP
2031
2039
2040 002000          HEADER CZTKI,A,0,5000,1,INTPRI
          L$NAME::          ;DIAGNOSTIC NAME
          002000          103          .ASCII /C/
          002001          132          .ASCII /Z/
          002002          124          .ASCII /T/
          002003          113          .ASCII /K/
          002004          111          .ASCII /I/
          002005          000          .BYTE 0
          002006          000          .BYTE 0
          002007          000          .BYTE 0
          002010          L$REV::          ;REVISION LEVEL
          002010          101          .ASCII /A/
          002011          L$DEPO::          ;0
          002011          060          .ASCII /O/
          002012          L$UNIT::          ;NUMBER OF UNITS
          002012          000001          .WORD T$PTHV
          002014          L$TIML::          ;LONGEST TEST TIME
          002014          005000          .WORD 5000
          002016          L$HPCP::          ;POINTER TO H.W. QUES.
          002016          027750          .WORD L$HARD
          002020          L$SPCP::          ;POINTER TO S.W. QUES.
          002020          030022          .WORD L$SOFT
          002022          L$HPTP::          ;PTR. TO DEF. H.W. PTABLE
          002022          002176          .WORD L$HW
          002024          L$SPTP::          ;PTR. TO S.W. PTABLE
          002024          002204          .WORD L$SW
          002026          L$LADP::          ;DIAG. END ADDRESS
          002026          031510          .WORD L$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$PRIO::          ;DIAGNOSTIC RUN PRIORITY
          002042          000340          .WORD INTPRI
          002044          L$ENVI::          ;FLAGS DESCRIBE HOW IT WAS SETUP
          002044          000000          .WORD 0

```

002046		L\$EXP1::	;EXPANSION WORD	.WORD	0
002046	000000				
002050		L\$MREV::	;SVC REV AND EDIT #	.BYTE	C\$REVISION
002050	003			.BYTE	C\$EDIT
002051	003				
002052		L\$EF::	;DIAG. EVENT FLAGS	.WORD	0
002052	000000			.WORD	0
002054	000000				
002056		L\$SPC::		.WORD	0
002056	000000				
002060		L\$DEVP::	; POINTER TO DEVICE TYPE LIST	.WORD	L\$DVTYP
002060	002166				
002062		L\$REPP::	;PTR. TO REPORT CODE	.WORD	L\$RPT
002062	020052				
002064		L\$EXP4::		.WORD	0
002064	000000				
002066		L\$EXP5::		.WORD	0
002066	000000				
002070		L\$AUT::	;PTR. TO ADD UNIT CODE	.WORD	L\$AU
002070	023710				
002072		L\$DUT::	;PTR. TO DROP UNIT CODE	.WORD	L\$DU
002072	023644				
002074		L\$LUN::	;LUN FOR EXERCISERS TO FILL	.WORD	0
002074	000000				
002076		L\$DESP::	;POINTER TO DIAG. DESCRIPTION	.WORD	L\$DESC
002076	002140				
002100		L\$LOAD::	;GENERATE SPECIAL AUTOLOAD EMT	EMT	E\$LOAD
002100	104035				
002102		L\$ETP::	;POINTER TO ERR_TBL	.WORD	0
002102	000000				
002104		L\$ICP::	;PTR. TO INIT CODE	.WORD	L\$INIT
002104	021664				
002106		L\$CCP::	;PTR. TO CLEAN-UP CODE	.WORD	L\$CLEAN
002106	023602				
002110		L\$ACP::	;PTR. TO AUTO CODE	.WORD	L\$AUTO
002110	023160				
002112		L\$PRT::	;PTR. TO PROTECT TABLE	.WORD	L\$PROT
002112	021656				
002114		L\$TEST::	;TEST NUMBER	.WORD	0
002114	000000				
002116		L\$DLY::	;DELAY COUNT	.WORD	0
002116	000000				
002120		L\$HIME::	;PTR. TO HIGH MEM	.WORD	0
002120	000000				

```

2049      .SBTTL DISPATCH TABLE
2050
2051      ;**
2052      ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
2053      ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
2054      ;--
2055
2056      DISPATCH 6          ; SIX TESTS
                                .WORD 6
002122    000006
002122    000006          L$DISPATCH::
002124    024004          .WORD T1
002126    025372          .WORD T2
002130    026046          .WORD T3
002132    026524          .WORD T4
002134    026670          .WORD T5
002136    027002          .WORD T6
2057
2064
2065      .SBTTL DESCRIPTIVE TEXT
2066
2067      ;**
2068      ;2 LINES OF TEXT PRINTED TO THE OPERATOR TO IDENTIFY THE DIAGNOSTIC AND THE DEVICE UNDER TES
2069      ;--
2070
2071      DESCRIPT          <DATA RELIABILITY TEST>
                                .ASCIZ /DATA RELIABILITY TE
002140    104      101      124
002140    101      040      122
002143    101      040      122
002146    105      114      111
002151    101      102      111
002154    114      111      124
002157    131      040      124
002162    105      123      124
002165    000
2072      DEVTYP          <TK25>          .EVEN
002166    124      113      062          L$DVTYP::          .ASCIZ /TK25/
002166    124      113      062
002171    065      000
                                .EVEN

```

```

2075      .SBTTL  DEFAULT HARDWARE P-TABLE
2076
2077      ;**
2078      ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
2079      ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
2080      ; IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
2081      ;--
2082
2083      BGNHW  DFPTBL
                .WORD  L10000-L$HW/2
                L$HW::
                DFPTBL::
2084
2090
2091      002176  172522      ;TSSR ADDRESS.
2092      002200  000224      ;VECTOR ADDRESS.
2093
2094      002202
                ENDHW
                L10000:

```

```

2097          .SBTTL  SOFTWARE P-TABLE
2098
2099          ;**
2100          ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
2101          ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
2102          ;--
2103          BGNSW  SFPTBL
                .WORD  L10001-L$SW/2
                L$SW::
                SFPTBL::
2110 002202    001    CLRFLG::.BYTE 1      ;CLEAR COUNTERS FLAG.
2111 002205    000    RRANV::.BYTE 0      ;RESET RANDOM VARIABLES EACH PASS FLAG.
2112 002206    000    HAE::.BYTE 0       ;HALT AFTER EACH COMMAND FLAG.
2113 002207    000    ERCVER::.BYTE 0    ;ENABLE RECOVERABLE ERROR PRINTS FLAG.
2114 002210    000    IREC::.BYTE 0     ;INHIBIT ERROR RECOVERY FLAG.
2115 002211    001    BADTSW::.BYTE 1    ;BAD TAPE SWITCH TO REWRITE ON SAME SPOT & DETECT BAD TAPE
2116 002212    000    DINT::.BYTE 0     ;DISABLE INTERRUPTS FLAG.
2117 002213    000    PIRE::.BYTE 0     ;INHIBIT RESIDUAL FRAMECOUNT ERROR REPORT FLAG.
2118 002214    000    RAMWRT::.BYTE 0   ;ENABLE OPTIONAL RAM DUMP
2119 002215    000    .BYTE 0          ;SPARE
2120 002216    000    EWPRT::.BYTE 0    ;ENABLE EARLY WARNING END OF TRACK STATUS PRINTS
2121 002217    000    CHGFLG::.BYTE 0   ;CHANGE CMD SEQ TABLE FLAG.
2122          .EVEN
2123 002220    000040 CHAR:: CH.EAI      ;CHARACTERISTICS CODE (DEFAULT = 40).
2124 002222    000015 CMDD:: .WORD 13.   ;COMMAND 2 (DEFAULT = REWIND).
2125 002224    000001 .WORD 1         ;BYTE COUNT
2126 002226    000001 .WORD 1         ;NUMBER OF OPERATIONS
2127 002230    000007 .WORD RANP      ;RANDOM DATA
2128 002232    000004 .WORD 4         ;COMMAND 3 (DEFAULT = WRITE)
2129 002234    010000 .WORD DATCNT    ;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
2130 002236    035230 .WORD 15000.   ;NUMBER OF OPERATIONS (DEFAULT = 15000).
2131 002240    000007 .WORD RANP      ;RANDOM DATA
2132 002242    000015 .WORD 13.     ;COMMAND 4 (DEFAULT = REWIND)
2133 002244    000001 .WORD 1         ;BYTE COUNT
2134 002246    000001 .WORD 1         ;ONE OPERATION
2135 002250    000007 .WORD RANP      ;RANDOM DATA
2136 002252    000002 .WORD 2         ;COMMAND 5 (DEFAULT = READ FWD).
2137 002254    010000 .WORD DATCNT    ;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
2138 002256    035230 .WORD 15000.   ;NUMBER OF OPERATIONS (DEFAULT = 15000)
2139 002260    000007 .WORD RANP      ;RANDOM DATA
2140 002262    000033 .WORD 27.     ;TERMINATOR
2141 002264    000001 .WORD 1         ;BYTE COUNT
2142 002266    000001 .WORD 1         ;NUMBER OF OPERATIONS
2143 002270    000007 .WORD RANP      ;PATTERN
2144 002272    000033 .WORD 27.     ;END OF CMD SEQ TABLE CODE (DEF) OR CMD 7
2145 002274    010000 .WORD DATCNT    ;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
2146 002276    035230 .WORD 15000.   ;NUMBER OF OPERATIONS (DEFAULT = 15000).
2147 002300    000007 .WORD RANP      ;RANDOM DATA
2148 002302    000033 .WORD 27.     ;END OF CMD SEQ TABLE CODE (DEF) OR CMD 8
2149 002304    010000 .WORD DATCNT    ;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
2150 002306    035230 .WORD 15000.   ;NUMBER OF OPERATIONS (DEFAULT = 15000).
2151 002310    000007 .WORD RANP      ;RANDOM DATA
2152 002312    .ENDSW
                L10001:
2153 002312    .ENDMOD

```

2166
2167
2168
2177
2178 002312
2179
2180
2181
2182
2183
2184
2185 002312

.TITLE GLOBAL AREAS
.SBTTL GLOBAL EQUATES SECTION

BGNMOD

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

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

000040	EF.START==	32.	; START COMMAND WAS ISSUED
000037	EF.RESTART==	31.	; RESTART COMMAND WAS ISSUED
000036	EF.CONTINUE==	30.	; CONTINUE COMMAND WAS ISSUED
000035	EF.NEW==	29.	; A NEW PASS HAS BEEN STARTED
000034	EF.PWR==	28.	; A POWER-FAIL/POWER-UP OCCURRED

; PRIORITY LEVEL DEFINITIONS

000340	PRI07==	340
000300	PRI06==	300


```

000240      PRI05== 240
000200      PRI04== 200
000140      PRI03== 140
000100      PRI02== 100
000040      PRI01== 40
000000      PRI00== 0
    
```

```

;
;OPERATOR FLAG BITS
;
000004      EVL==      4
000010      LOT==     10
000020      ADR==     20
000040      IDU==     40
000100      ISR==    100
000200      UAM==    200
000400      BOE==    400
001000      PNT==   1000
002000      PRI==   2000
004000      IXE==   4000
010000      IBE==  10000
020000      IER==  20000
040000      LOE==  40000
100000      HOE== 100000
    
```

2186
 2194
 2195
 2196
 2197
 2198
 2199
 2200
 2201
 2202
 2203
 2204
 2205
 2206
 2207
 2208
 2209
 2210
 2211
 2212
 2213
 2214
 2215
 2216
 2217
 2218
 2219

```

; REGISTER USAGE.
;
;      R0 - PASSES PARAMETERS TO/FROM DIAGNOSTIC SUPERVISOR.
;      R1 - COMMAND SEQUENCE TABLE POINTER.
;      R2 - GENERAL PURPOSE REGISTER.
;      R3 - GENERAL PURPOSE REGISTER.
;      R4 - GENERAL PURPOSE REGISTER.
;      R5 - CURRENT LOGICAL DEVICE NUMBER x 2.
;      R6 - STACK POINTER.
;      R7 - PROGRAM COUNTER.
    
```

;THE FOLLOWING ARE BIT DEFINITIONS FOR THE TSSR REGISTERS.

```

100000      TS.SC==100000      ;SPECIAL CONDITION BIT.
040000      TS.UPE==40000      ;UNIBUS PARITY ERROR
020000      TS.SPE==20000      ;SERIAL BUS PARITY ERROR.
010000      TS.RMR==10000      ;REGISTER MODIFICATION REFUSED.
004000      TS.NXM==4000       ;NON-EXISTENT MEMORY.
002000      TS.NBA==2000       ;NEED BUFFER ADDRESS.
001000      TS.A17==1000       ;BUS ADDRESS BIT 17.
000400      TS.A16==400        ;BUS ADDRESS BIT 16.
000200      TS.SSR==200        ;UNIT READY BIT.
000100      TS.OFL==100        ;OFF LINE.
177717      TSC.FCC==177717    ;FATAL CLASS CODE MASK.
177761      TSC.TCC==177761    ;TERMINATION CLASS CODE MASK.
    
```

```

2221      ;THE FOLLOWING ARE BIT DEFINITIONS FOR THE COMMAND WORD
2222
2223      100000      ACK.C==100000      ;ACKNOWLEDGE BIT
2224      040000      CVC.C==40000      ;CLEAR VOLUME CHECK.
2225      020000      OPP.C==20000      ;OPPOSITE BIT
2226      010000      SWB.C==10000      ;SWAP BYTE BIT
2227      004000      MOD.C3==4000      ;MODE BIT 3
2228      004000      BRP.C==4000      ;BYTE/RECORD/FILE COUNT FLAG BIT. NOT USED
2229      ;BY TK25 BUT USED INTERNALLY BY THIS PROGRAM ONLY.
2230      002000      MOD.C2==2000      ;MODE BIT 2
2231      001000      MOD.C1==1000      ;MODE BIT 1
2232      000400      MOD.C0==400       ;MODE BIT 0
2233      000200      IE.C==200         ;INTERRUPT ENABLE
2234      000100      FMT.C1==100      ;FORMAT BIT 1
2235      000100      VFY.C==100       ;WRITE VERIFY FLAG BIT. INTERNAL USE ONLY.
2236      ;NOT USED BY TK25.
2237      000040      FMT.C0==40       ;FORMAT BIT 0.
2238      000040      JMP.C==40        ;JUMP BIT-TO DIRECT THIS PROGRAM TO JUMP TO
2239      ;A CERTAIN LOCATION IN THE COMMAND SEQUENCE
2240      ;TABLE. INTERNAL USE ONLY.
2241      000020      CMD.C4==20        ;COMMAND BIT 4
2242      000020      DLY.C==20        ;INSERT DELAY. INTERNAL USE ONLY.
2243      000010      CMD.C3==10        ;COMMAND BIT 3
2244      000004      CMD.C2==4         ;COMMAND BIT 2
2245      000002      CMD.C1==2         ;COMMAND BIT 1
2246      000001      CMD.C0==1         ;COMMAND BIT 0
2247
2248      ; BIT DEFINITIONS FOR DEVICE CHARACTERISTICS.
2249
2250      000200      CH.ESS==200        ;ENABLE SKIP TAPE MARKS STOP (STOP AT LOGICAL EOT).
2251      000040      CH.EAI==40        ;ENABLE ATTENTION INTERRUPTS.
2252      000020      CH.ERI==20        ;ENABLE MESSAGE BUFFER RELEASE INTERRUPTS.
2253      000040      DFTSCH==CH.EAI    ;DEFAULT CHARACTERISTICS CODE.
2254
2255      ;THE FOLLOWING INDICATES THE RELATIVE POSITIONS OF THE STATUS WORDS
2256      ;IN THE MESSAGE BUFFER.
2257
2258      000004      MS.RFC==4          ;RESIDUAL FRAME COUNT.
2259      000006      MS.XS0==6         ;EXT STATUS REG 0
2260      000010      MS.XS1==10        ;EXT STATUS REG 1
2261      000012      MS.XS2==12        ;EXT STATUS REG 2
2262      000014      MS.XS3==14        ;EXT STATUS REG 3
2263
2264      ;THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0.
2265
2266      100000      XC.TMK==100000     ;TAPE MARK.
2267      040000      XO.RLS==40000     ;RECORD LENGTH SHORT.
2268      020000      XO.LET==20000     ;LOGICAL EOT.
2269      010000      XO.RLL==10000     ;RECORD LENGTH LONG.
2270      000100      XO.ONL==100       ;ON LINE BIT.
2271      000002      XO.BOT==2         ;BOT BIT.
2272      000001      XO.EOT==1         ;EOT BIT.
2273
2274      ;THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 1.
2275
2276      000010      X1.EWN==BIT3
2277

```

```

2278 ;THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2.
2279
2280 100000 X2.OPM==100000 ;OPERATION IN PROGRESS, TAPE MOVING
2281
2282 ;THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3.
2283
2284 000010 X3.DCK==10 ;DENSITY CHECK.
2285 157400 X3.RNY==157400 ;CAPSTAN RUNAWAY UDIAG ERROR CODE.
2286
2287 ;THE FOLLOWING DEFINITIONS SHOW THE RELATIVE POSITIONS OF THE COMMAND
2288 ;PACKET ENTRIES.
2289
2290 000000 CP.CMD==0 ;CMDPKT+0==TK25 COMMAND.
2291 000002 CP.ADL==2 ;CMDPKT+2==BUFFER ADDRESS LOW.
2292 000004 CP.ADH==4 ;CMDPKT+4==BUFFER ADDRESS HIGH.
2293 000006 CP.CNT==6 ;CKDPKT+6==BYTE/FILE/RECORD COUNT
2294
2295 ; MISCELLANEOUS DEFINITIONS.
2296
2297 000340 INTPRI==PRI07 ;PRIORITY TO BE USED IN INTERRUPT STATE.
2298 000010 SCHCNT==10 ;ARBITRARY BYTE LENGTH FOR CHARACTERISTIC
2299 ;BUFFER LENGTH. (EVEN #)
2300 000016 MSGCNT==16 ;MESSAGE BUFFER LENGTH IN BYTES. (EVEN #)
2301 000020 DIACNT==20 ;DIAGNOSTIC COMMAND BUFFER EXTENT.
2302 010000 DATCNT==4096. ;MAXIMUM RECORD LENGTH IN BYTES.
2303 ;THIS COUNT SHOULD BE A MULTIPLE OF 256 TO INSURE
2304 ;PROPER READ/WRITE BUFFER ALLOCATION BY THE SUPER.
2305 177740 RNOPSC==177740 ;RANDOM # OF OPERATIONS MASK.
2306 000007 RANP==7 ;CODE TO SELECT RANDOM PATTERN.
2307 000020 RRECL==16. ;READ RECOVERY ATTEMPT LIMIT.
2308 000020 WRECL==16. ;WRITE RECOVERY ATTEMPT LIMIT.
2309 153624 RANBC==153624 ;CONSTANT USED TO RESET RANDOM # GENERATOR BASE.
2310 032561 RANSC==32561 ;CONSTANT USED TO RESET RANDOM # SAVE LOCATION.
2311 177774 NINUSE==177774 ;NOT IN USE CODE FOR DEVICE STATE TABLE.
2312 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
2313 ;NOT "COMMAND" BITS.
2314
2315 ;THE FOLLOWING DEFINES THE COMMAND WORD FOR EACH TK25 COMMAND.
2316
2317 100013 DRI== ACK.C!CMD.C3!CMD.C1!CMD.CO ;DRIVE INIT.
2318
2319 104001 RDF== ACK.C!BRF.C!CMD.CO ;READ FORWARD
2320
2321
2322 104401 RDR== ACK.C!BRF.C!MOD.CO!CMD.CO ;READ REVERSE
2323
2324
2325 104005 WRT== ACK.C!BRF.C!CMD.CO!CMD.C2 ;WRITE COMMAND
2326
2327
2328 104105 WTV== ACK.C!BRF.C!VFY.C!CMD.CO!CMD.C2 ;WRITE VERIFY
2329
2330
2331 104010 SRF== ACK.C!BRF.C!CMD.C3 ;SPACE RECORD FORWARD
2332
2333
2334

```

2335	104410	SRR==	ACK.C!BRF.C!MOD.CO!CMD.C3	
2336				;SPACE RECORD REVERSE
2337				
2338	105401	RNR==	ACK.C!BRF.C!MOD.C1!MOD.CO!CMD.CO	
2339				;READ REV RETRY1 - REREAD NEXT REVERSE, IE. SPACE FWD, READ REVERSE
2340				
2341	125401	RNF==	ACK.C!BRF.C!OPP.C!MOD.C1!MOD.CO!CMD.CO	
2342				;READ REV RETRY2 - REREAD NEXT FORWARD, IE.READ FORWARD, SPACE REVERSE
2343				
2344	105001	RPF==	ACK.C!BRF.C!MOD.C1!CMD.CO	
2345				;READ FWD RETRY1 - REREAD PREVIOUS FORWARD, IE. SPACE REVERSE, READ FORWARD
2346				
2347	125001	RPR==	ACK.C!BRF.C!OPP.C!MOD.C1!CMD.CO	
2348				;READ FWD RETRY2 - REREAD PREVIOUS REVERSE, IE. READ REVERSE, SPACE FORWARD
2349				
2350	105005	WRR==	ACK.C!MOD.C1!BRF.C!CMD.C2!CMD.CO	
2351				;WRITE RETRY
2352				
2353	102010	RWD==	ACK.C!MOD.C2!CMD.C3	
2354				;REWIND COMMAND
2355				
2356	100012	MBR==	ACK.C!CMD.C3!CMD.C1	
2357				;MESSAGE BUFFER RELEASE
2358				
2359	100011	WTM==	ACK.C!CMD.C3!CMD.CO	
2360				;WRITE TAPE MARK.
2361				
2362	101011	WTR==	ACK.C!MOD.C1!CMD.C3!CMD.CO	
2363				;WRITE TAPE MARK RETRY.
2364				
2365	105010	SFF==	ACK.C!BRF.C!MOD.C1!CMD.C3	
2366				;SPACE FILE FORWARD
2367				
2368	105410	SFR==	ACK.C!BRF.C!MOD.CO!MOD.C1!CMD.C3	
2369				;SPACE FILE REVERSE
2370				
2371	100017	GES==	ACK.C!CMD.CO!CMD.C1!CMD.C2!CMD.C3	
2372				;GET EXTENDED STATUS
2373				
2374	100411	ERS==	ACK.C!MOD.CO!CMD.C3!CMD.CO	
2375				;ERASE 3 INCHES OF TAPE
2376				
2377	100412	UNL==	ACK.C!MOD.CO!CMD.C3!CMD.C1	
2378				;UNLOAD COMMAND
2379				
2380	101012	CLN==	ACK.C!MOD.C1!CMD.C3!CMD.C1	
2381				;ERASE TAPE.
2382				
2383	140004	SCH==	ACK.C!CVC.C!CMD.C2	
2384				;SET DEVICE CHARACTERISTICS.
2385	100006	DIA==	ACK.C!CMD.C2!CMD.C1	
2386				;DIAGNOSTICS.
2387	000040	JMP==	JMP.C	
2388				;JUMP TO "N"TH COMMAND
2389	000020	DLY==	DLY.C	
2390				;DELAY "N" MS.
2391	177777	END==	177777	
				;END OF COMMAND SEQUENCES

```

2393      .SBTTL GLOBAL DATA SECTION
2394
2395      ;**
2396      ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
2397      ; IN MORE THAN ONE TEST.
2398      ;--
2399
2400      ;      COMMAND PACKET.
2401
2402      =      .+3&177774      ;MUST BE ON MOD 4 BOUNDARY.
2403 002314 000000      CMDPKT:: 0      ;1ST WORD IS TK25 COMMAND.
2404 002316 000000      0      ;2ND WORD IS THE BUFFER LOW ADDRESS.
2405 002320 000000      0      ;3RD WORD IS THE BUFFER HIGH ADDRESS.
2406 002322 000000      0      ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.
2407
2408      ;      GET STATUS COMMAND PACKET.
2409
2410      =      .+3&177774      ;MUST BE ON MOD 4 BOUNDARY.
2411 002324 100017      GSCP:: .WORD  GES
2412
2413      ;      MESSAGE BUFFER RELEASE COMMAND PACKET.
2414
2415      =      .+3&177774      ;MUST BE ON MOD 4 BOUNDARY.
2416 002330 002330      BRCPK:: .WORD  MBR
2417 002330 100012
2418
2419      ;      REWIND COMMAND PACKET (USED IN ERROR RECOVERY ONLY)
2420
2421      =      .+3&177774      ;MUST BE ON A MODULE 4 BOUNDARY.
2422 002334 102010      RWCPK:: .WORD  RWD
2423 002336 000001      .WORD  1
2424
2425      ;      WORK AREA FOR ANALYSIS OF MESSAGE PACKET CONTENTS.
2426
2427 002340      MSGPKT:: .BLKW 7      ;1ST WORD:: MESSAGE TYPE.
2428      ;2ND WORD:: DATA FIELD LENGTH.
2429      ;3RD WORD:: RESIDUAL FRAME COUNT.
2430      ;4TH WORD:: XSTAT0
2431      ;5TH WORD:: XSTAT1
2432      ;6TH WORD:: XSTAT2
2433      ;7TH WORD:: XSTAT3
2434
2435      ;      MESSAGE PACKETS.
2436
2437 002356      MSGPK0:: .BLKW 7      ;MESSAGE PACKET FOR DEVICE #0
2438 002374      MSGPK1:: .BLKW 7      ;MESSAGE PACKET FOR DEVICE #1
2439 002412      MSGPK2:: .BLKW 7      ;MESSAGE PACKET FOR DEVICE #2
2440 002430      MSGPK3:: .BLKW 7      ;MESSAGE PACKET FOR DEVICE #3

```

```

2442      ;      SET CHARACTERISTIC BLOCK.
2443
2444 002446 002356  SCHBK:: MSGPK0      ;1ST WORD:: MSGPKT ADDR LO(SET UP BY EXECUTE ROUTINE).
2445 002450 000000      0      ;2ND WORD:: MSGPKT ADDR HI.
2446 002452 000016      MSGCNT      ;3RD WORD:: MSG BUFFER LENGTH (BYTES)
2447 002454 000040      CH.EAI      ;4TH WORD:: CHARACTERISTICS WORD(SET BY SETUP ROUTINE).
2448
2449      ;      TK25 REGISTER ADDRESSES.
2450
2451 002456      TSDB:: .BLKW 4      ;TK25 DATA BUFFER ADDRESSES.
2452 002466      TSSR:: .BLKW 4      ;TK25 STATUS REGISTER ADDRESSES.
2453 002476      TSVCT:: .BLKW 4      ;TK25 VECTOR ADDRESSES.
2454      002456      TSBA=TSDB      ;DATA BUFFER ADDRESS REGISTER.
2455
2456      ;      ADDRESSES OF MESSAGE PACKETS.
2457
2458 002506 002356  MSGPKA:: MSGPK0      ;DEVICE 0.
2459 002510 002374      MSGPK1      ;DEVICE 1.
2460 002512 002412      MSGPK2      ;DEVICE 2.
2461 002514 002430      MSGPK3      ;DEVICE 3.
2462
2463      ;      ADDRESSES OF INTERRUPT HANDLING ROUTINES.
2464
2465 002516 006552  TS4INT:: TS4IN0      ;DEVICE 0.
2466 002520 006560      TS4IN1      ;DEVICE 1.
2467 002522 006566      TS4IN2      ;DEVICE 2.
2468 002524 006574      TS4IN3      ;DEVICE 3.
2469
2470      ;      TK25 CODE LEVELS, WILL BE STORED AFTER SCH CMD IN BASIC FUNCTION TEST
2471
2472 002526 000000  TS4CL:: 0      ;DEVICE 0
2473 002530 000000      0      ;DEVICE 1
2474 002532 000000      0      ;DEVICE 2
2475 002534 000000      0      ;DEVICE 3
2476
2477      ;      UNIT NUMBERS OF ALL DEVICES BEING TESTED(1-4).
2478      ;      WHEN DEVICE IS NOT IN USE, IT,S LOCATION WILL = -3.
2479      ;      R5 WILL ALWAYS CONTAIN THE PRESENT LOGICAL UNIT NUMBER X 2.
2480
2481 002536 177774  DEVTBL:: .WORD NINUSE
2482 002540 177774      .WORD NINUSE
2483 002542 177774      .WORD NINUSE
2484 002544 177774      .WORD NINUSE
2485 002546 177777      .WORD END
2486
2487      ;      BAD TAPE TABLE POINTER: USED BY WRITE RETRY ROUTINE
2488      ;      "WRTY" TO LOG BAD TAPE SPOTS ON UNITS UNDER TEST
2489
2490 002550 003000  BTADDR:: BT0
2491 002552 003052      BT1
2492 002554 003124      BT2
2493 002556 003176      BT3

```

```

2495          ;          COUNTER AREA.
2496
2497          002560
2498 002560   CNTBGN=.
2499 002620   WRBC:: .BLKW  20      ;BYTES WRITTEN.
2500 002660   RRBC:: .BLKW  20      ;BYTES READ REV.
2501 002720   RFBC:: .BLKW  20      ;BYTES READ FWD.
2502 002730   WRREC:: .BLKW  4       ;RECOVERABLE WRITE ERRORS.
2503 002740   WRUNR:: .BLKW  4       ;UNRECOVERABLE WRITE ERRORS.
2504 002750   RRREC:: .BLKW  4       ;RECOVERABLE READ REV ERRORS.
2505 002760   RRUNR:: .BLKW  4       ;UNRECOVERABLE READ REV ERRORS.
2506 002770   RFREC:: .BLKW  4       ;RECOVERABLE READ FWD ERRORS.
2507 003000   RFUNR:: .BLKW  4       ;UNRECOVERABLE READ FWD ERRORS.
2508 003052   BTO:: .BLKW  21.      ;UNIT 0 BAT TAPE SPOTS LOG
2509 003124   BT1:: .BLKW  21.      ;UNIT 1 BAT TAPE SPOTS LOG
2510 003176   BT2:: .BLKW  21.      ;UNIT 2 BAT TAPE SPOTS LOG
2511 003250   BT3:: .BLKW  21.      ;UNIT 3 BAT TAPE SPOTS LOG
2512 003260   WRTYCT:: .BLKW  4      ;WRITE RETRY COUNTER
2513 003270   PASCNT:: .BLKW  4      ;PASS COUNT.
2514 003300   SCCNT:: .BLKW  4      ;SPECIAL CONDITION COUNT.
2515 003310   VFYCNT:: .BLKW  4      ;COUNT OF TK25 DATA COMPARE ERRORS.
2516 003320   HRDCNT:: .BLKW  4      ;COUNT OF HARD ERRORS.
2517          CNTEND=.
2518 003330   RECNT:: .BLKW  4      ;COUNT OF FATAL ERRORS.
2519          ;END OF STATICTICAL COUNTERS.
2520          ;NUMBER OF RECORDS FROM BOT; CLEARED ON REWIND
2521          ;AND WHEN RESTARTING OR CONTINUING TEST 2.
2522          ;LENGTH OF STATISTICAL COUNTER AREA.
2523          CNTLEN==CNTEND-CNTBGN
2524          000550
2525          HERE: .BYTE 0          ;TEST 3 ASCII SEMAPHORE
2526          ; THE FOLLOWING ARE THE DEFINITIONS OF VARIABLES
2527          ; USED BY THE PROGRAM.
2528          .EVEN
2529 003340   000   RAMHLD: .WORD 0          ;RAM ADDR HOLDER 1ST ADDRESS
2530 003342   000000 RAMR5H: .WORD 0          ;HOLDS R5 FOR LATER
2531 003344   000000 RAMDATA:: .BLKW 16.      ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
2532 003406   000000 RAMSIZ: .WORD 0          ;RAM DATA SIZE FOR PRAMPKT ROUTINE
2533 003410   000000 CMPDAT: .WORD 0          ;COUNTS # OF READS (TEST 3) BEFORE ALLOWING A DATA COMPARE
2534 003412   000003 DATRAT: .WORD 3          ;CONTROLS THE DATA COMPARE RATIO
2535 003414   000000 DATAWT: .WORD 0          ;WRITE BUFFER ADDRESS.
2536 003416   000000 DATARD: .WORD 0          ;READ BUFFER ADDRESS.
2537 003420   000000 NCNT: .WORD 0          ;STORAGE FOR VALUE OF N.
2538 003422   000000 NCNT1: .WORD 0          ;TEMP STORAGE FOR VALUE OF N.
2539 003424   000000 BRFCNT: .WORD 0          ;STORAGE FOR BPCR VALUE.
2540 003426   177777 CMDWRD: .WORD END        ;CONTAINS COMMAND WORD BEING EXECUTED PRESENTLY.
2541 003430   177777 CMDSAV: .WORD END        ;SAVE LOCATION FOR CMD WORD DURING ERROR RECOVERY
2542 003432   177777 PCMDWD: .WORD END        ;CONTAINS PREVIOUS COMMAND WORD.
2543 003434   000000 CMDLG: .WORD 0          ;CURRENT COMMAND LOGGING CODE.
2544 003436   000000 LENMSK: .WORD 0          ;RANDOM WRITE LENGTH MASK, TO BE SET UP BY TESTS
2545 003440   153624 RANB: .WORD 153624      ;RANDOM # GENERATOR BASE.
2546 003442   032561 RANS: .WORD 32561      ;RANDOM # SAVE LOCATION.
2547 003444   000000 TIME1: .WORD 0          ;TIME COUNT 1.
2548 003446   000000 TIME2: .WORD 0          ;TIME COUNT 2.
2549 003450   000000 JLOOP: .WORD 0          ;JMP COMMAND LOOP COUNT.
2550 003452   000000 JLOC: .WORD 0          ;JMP COMMAND LOCATION COUNT.
2551 003454   000000 PATERN:: .WORD 0          ;PATTERN SELECT CODE.

```

```

2551 003456 000000      CTCC:: .WORD 0      ;CURRENT TERMINATION CLASS CODE.
2552 003460 000000      R5SAVE:: .WORD 0    ;LOCATION FOR SAVING CURRENT DEVICE POINTER.
2553 003462 000000      TSSREG:: .WORD 0    ;CURRENT STATUS REGISTER.
2554          003414      DIABLK="DATAWT     ;WRITE BUFFER ALSO USED FOR DIAG CMD.
2555
2556          ;          ERROR FLAG AREA, THESE FLAGS ARE CLEARED DURING INITIALIZATION AND
2557          ;          AFTER EACH COMMAND IS COMPLETED.
2558
2559          003464      BGNFLG=.
2560 003464 000000      RETRYC:: .WORD 0    ;# OF RECOVERY ATTEMPTS EXECUTED.
2561 003466          000      RPTCNT:: .BYTE 0    ;WRITE REPEAT ON SAME SPOT CNTR: 4 PER WRITE RETRY
2562 003467          000      WRTYFG:: .BYTE 0    ;WRITE RETRY ON SAME SPOT IN PROGRESS FLAG
2563 003470          000      WRTYER:: .BYTE 0    ;WRITE RETRY ON SAME SPOT ERROR FLAG
2564 003471          000      RECLOG:: .BYTE 0    ;RECORD COUNT HAS BEEN UPDATED FOR THIS RECORD.
2565 003472          000      ERLOG:: .BYTE 0    ;DATA BYTES AND ERRORS HAVE BEEN LOGGED FOR THIS RECORD.
2566 003473          000      RWERR:: .BYTE 0    ;READ/WRITE ERROR HAS OCCURED.
2567 003474          000      UNREC:: .BYTE 0    ;UNRECOVERABLE ERROR HAS OCCURED.
2568 003475          000      ERRREC:: .BYTE 0   ;ERROR RECOVERY MODE.
2569          .EVEN
2570          003476      ENDERF=.
2571
2572          ;          ADDITIONAL FLAGS, THESE FLAGS ARE CLEARED DURING INITIALIZATION.
2573
2574 003476          INTFLG:: .BLKW 4    ;INTERRUPT OCCURRED FLAGS FOR EACH DEVICE.
2575 003506          EOTFLG:: .BLKW 4    ;EOT/BOT FLAGS FOR EACH DEVICE (XSTATO).
2576 003516 000000      BTPT:: .WORD 0    ;BAD TAPE SPOT POINTER TO BTO-BT3 VIA BTADDR
2577 003520          000      EXPBOT:: .BYTE 0    ;BOT IS EXPECTED, DO NOT ABORT ON BOT/FUNC RTI.
2578 003521          000      RANDOM:: .BYTE 0   ;RANDOM EVERYTHING FLAG.
2579 003522          000      VFYFLG:: .BYTE 0   ;SET DURING WRITE/VERIFY COMMAND.
2580 003523          000      RPTFLG:: .BYTE 0   ;PERFORMANCE REPORT HAS BEEN REQUESTED.
2581 003524          000      SWBFLG:: .BYTE 0   ;ENABLES SWAP BYTE FUNCTION WHEN NOT EQUAL TO ZERO.
2582 003525          000      LOOPCT:: .BYTE 0   ;WRITE LOOP CONTROL (TEST 3)
2583 003526          000      IRE:: .BYTE 0    ;INHIBIT RESIDUAL FRAME COUNT ERROR REPORT.
2584 003527          000      DROPED:: .BYTE 0   ;CURRENT UNIT HAS BEEN DROPPED
2585 003530          000      T1SWB:: .BYTE 0   ;TEST1 SWAP BYTES FLAG
2586 003531          000      ALLEOT:: .BYTE 0   ;ALL UNITS @ EOT FLAG
2587 003532          000      STREAM:: .BYTE 0   ;INDICATES TEST ONE UNIT AT A TIME, COMPLETELY.
2588 003533          000      ERSFLG:: .BYTE 0   ;ERASE FLAG: DO ERASE AFTER A SPACE REV TO DELETE
2589          .EVEN      ;BADLY WRITTEN RECORD. 1 TO 4 ERASES LEAVING
2590          ;          A 3 TO 12 INCH GAP MAY RESULT.
2591
2592          003534      ENDFLG=.
2593
2594          ;          ADDITIONAL FLAGS, THESE FLAGS ARE CLEARED ONLY AFTER BEING CHECKED.
2595
2596 003534          000      STAFLG:: .BYTE 0   ;START FLAG - SET BY INIT CODE IF STARTING.
2597 003535          000      PWRFLG:: .BYTE 0   ;POWER FAILURE FLAG - SET ONLY DURING INIT.
2598 003536          000      TRAPD4:: .BYTE 0   ;TRAPED AT 4 FLAG
2599 003537          000      MISCFG:: .BYTE 0   ;MISCELLANEOUS FLAG
2600

```



```

2602      ; OPERATOR FLAG SETTINGS PASSED BY DIAG. SUPERVISOR IN A 16 BIT WORD
2603      ; SEE GLOBAL EQUATES SECTION FOR FLAG BIT LIST
2604
2605 003540 000000 OPFLAG:: .WORD 0 ;READ ONLY OPERATOR FLAG WORD
2606      .EVEN
2607
2608      ;THE FOLLOWING IS THE COMMAND SEQUENCE TABLE. THE TABLE
2609      ;HAS DEFAULT VALUES AT PROGRAM LOAD AS SHOWN. THESE VALUES
2610      ;CAN BE UPDATED BY A TEST OR BY OPERATOR INPUT.
2611
2612 003542 140004 CMDSEQ:: .WORD SCH ;SET CHARACTERISTICS.
2613 003544 000040 .WORD CH.EAI
2614 003546 000001 .WORD 1
2615 003550 000000 .WORD 0
2616 003552 102010 CMDSE2:: .WORD RWD ;REWIND.
2617 003554 000001 .WORD 1 ;BYTE COUNT.
2618 003556 000001 .WORD 1 ;ONCE.
2619 003560 000007 .WORD RANP ;PATTERN.
2620 003562 104005 .WORD WRT ;WRITE.
2621 003564 010000 .WORD DATCNT ;MAX BUFFER LENGTH.
2622 003566 035230 .WORD 15000. ;15000 RECORDS.
2623 003570 000007 .WORD RANP ;PATTERN - RANDOM DATA
2624 003572 102010 .WORD RWD ;REWIND
2625 003574 000001 .WORD 1 ;BYTE COUNT
2626 003576 000001 .WORD 1 ;ONE ITERATION
2627 003600 000007 .WORD RANP ;RANDOM DATA
2628 003602 104001 .WORD RDF ;READ FWD.
2629 003604 010000 .WORD DATCNT ;MAX BUFFER LENGTH.
2630 003606 035230 .WORD 15000. ;15000 RECORDS.
2631 003610 000007 .WORD RANP ;RANDOM DATA
2632 003612 102010 .WORD RWD ;REWIND
2633 003614 000001 .WORD 1 ;BYTE COUNT.
2634 003616 000001 .WORD 1 ;ONCE.
2635 003620 000007 .WORD RANP ;PATTERN.
2636 003622 .BLKW 4 ;EXTENSION TO HOLD 1 MORE CMD.
2637 003632 177777 SEQEND:: .WORD END ;SOFT END OF SEQUENCE TABLE.
2638 003634 177777 .WORD END
2639 003636 177777 .WORD END
2640 003640 177777 .WORD END
2641 003642 177777 .WORD END ;HARD END OF SEQUENCE TABLE.

```

```

2643                                     ;THE FOLLOWING IS THE TK25 COMMAND TABLE
2644
2645 003644 100013          CMDTBL:: .WORD DRI          ;DRIVE INIT.
2646 003646 104001          .WORD RDF          ;READ FORWARD.
2647 003650 104401          .WORD RDR          ;READ REVERSE.
2648 003652 104005          .WORD WRT          ;WRITE
2649 003654 104105          .WORD WTV          ;WRITE/VERIFY. (WRITE ALL RECORDS, RDR AND
2650                                     ;CHECK DATA ON ALL RECORDS, RDF AND
2651                                     ;CHECK DATA ON ALL RECORDS.)
2652 003656 104010          .WORD SRF          ;SPACE "N" RECORDS FORWARD.
2653 003660 104410          .WORD SRR          ;SPACE "N" RECORDS REVERSE.
2654 003662 105401          .WORD RNR          ;READ NEXT REVERSE. I.E., SPACE FWD, READ REVERSE.
2655 003664 125401          .WORD RNF          ;READ NEXT FORWARD. I.E., READ FORWARD, SPACE REVERSE.
2656 003666 105001          .WORD RPF          ;READ PREVIOUS FORWARD. I.E., SPACE REVERSE, READ FORWARD
2657 003670 125001          .WORD RPR          ;READ PREVIOUS REVERSE. I.E., READ REVERSE, SPACE FORWARD
2658 003672 105005          .WORD WRR          ;WRITE RETRY.
2659 003674 102010          .WORD RWD          ;REWIND.
2660 003676 100012          .WORD MBR          ;MESSAGE BUFFER RELEASE
2661 003700 100011          .WORD WTM          ;WRITE TAPE MARK
2662 003702 101011          .WORD WTR          ;WRITE TAPE MARK RETRY.
2663 003704 105010          .WORD SFF          ;SPACE "N" FILES FORWARD.
2664 003706 105410          .WORD SFR          ;SPACE "N" FILES REVERSE.
2665 003710 100017          .WORD GES          ;GET EXTENDED STATUS.
2666 003712 100411          .WORD ERS          ;ERASE 3 INCHES OF TAPE.
2667 003714 100412          .WORD UNL          ;REWIND AND UNLOAD.
2668 003716 101012          .WORD CLN          ;CLEAR TAPE.
2669 003720 140004          .WORD SCH          ;SET CHARACTERISTICS.
2670 003722 100006          .WORD DIA          ;DIAGNOSTIC COMMAND.
2671 003724 000040          .WORD JMP          ;JUMP TO THE NTH COMMAND IN THE SEQUENCE.
2672 003726 000020          .WORD DLY          ;DELAY "N" MS.
2673 003730 177777          .WORD END          ;END OF COMMAND TABLE
2674

```

```

2676                                     ; THE FOLLOWING TABLE CONTAINS THE ASCII FOR EACH COMMAND.
2677
2678 003732      104      122      111  CMDASC:: .ASCII /DRI/      ;DRIVE INIT.
2679 003735      122      104      106      .ASCII /RDF/      ;READ FORWARD.
2680 003740      122      104      122      .ASCII /RDR/      ;READ REVERSE.
2681 003743      127      122      124      .ASCII /WRT/      ;WRITE
2682 003746      127      124      126      .ASCII /WTV/      ;WRITE/VERIFY. (WRITE ALL RECORDS, RDR AND CHECK DATA
2683                                     ;ON ALL RECORDS, RDF AND CHECK DATA ON ALL RECORDS.)
2684 003751      123      122      106      .ASCII /SRF/      ;SPACE "N" RECORDS FORWARD.
2685 003754      123      122      122      .ASCII /SRR/      ;SPACE "N" RECORDS REVERSE.
2686 003757      122      116      122      .ASCII /RNR/      ;READ NEXT REVERSE. I.E., SPACE FWD READ REVERSE.
2687 003762      122      116      106      .ASCII /RNF/      ;READ NEXT FORWARD, I.E., READ FORWARD, SPACE REVERSE.
2688 003765      122      120      106      .ASCII /RPF/      ;READ PREVIOUS FORWARD. IE., SPACE REVERSE, READ FORWARD
2689 003770      122      120      122      .ASCII /RPR/      ;READ PREVIOUS REVERSE. IE., READ REVERSE, SPACE FORWARD
2690 003773      127      122      122      .ASCII /WRR/      ;WRITE RETRY.
2691 003776      122      127      104      .ASCII /RWD/      ;REWIND.
2692 004001      115      102      122      .ASCII /MBR/      ;MESSAGE BUFFER RELEASE
2693 004004      127      124      115      .ASCII /WTM/      ;WRITE TAPE MARK
2694 004007      127      124      122      .ASCII /WTR/      ;WRITE TAPE MARK RETRY.
2695 004012      123      106      106      .ASCII /SFF/      ;SPACE "N" FILES FORWARD.
2696 004015      123      106      122      .ASCII /SFR/      ;SPACE "N" FILES REVERSE.
2697 004020      107      105      123      .ASCII /GES/      ;GET EXTENDED STATUS.
2698 004023      105      122      123      .ASCII /ERS/      ;ERASE 3 INCHES OF TAPE.
2699 004026      125      116      114      .ASCII /UNL/      ;REWIND AND UNLOAD.
2700 004031      103      114      116      .ASCII /CLN/      ;CLEAN TAPE.
2701 004034      123      103      110      .ASCII /SCH/      ;SET CHARACTERISTICS. WHERE BRF=200, 40, 20, 0.
2702                                     ;SEE TK25 PROGRAMMING SPECIFICATION FOR DESCRIPTION.
2703 004037      104      111      101      .ASCII /DIA/      ;DIAGNOSTICS. SEE TK25 PROGRAMMING SPECIFICATION
2704                                     ;FOR DESCRIPTION. ODT MUST BE USED TO LOAD DIAGNOSTIC DATA
2705                                     ;INTO THE WRITE BUFFER BEFORE THIS CMD IS ISSUED.
2706 004042      112      115      120      .ASCII /JMP/      ;JUMP TO THE NTH COMMAND IN THE COMMAND
2707                                     ;SEQUENCE TABLE, WHERE N IS DEFINED IN
2708                                     ;THE # OF OPERATIONS.
2709 004045      104      114      131      .ASCII /DLY/      ;DELAY "N" MS, WHERE N IS DEFINED IN
2710                                     ;THE # OF OPERATIONS.
2711 004050      105      116      104      .ASCII /END/      ;END OF COMMAND SEQUENCE.
2712 .EVEN
2713
2714
2715

```

```

2717 .SBTTL GLOBAL TEXT SECTION
2718
2719
2720 ;**
2721 ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
2722 ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
2723 ; MORE THAN ONE TEST.
2724 ;--
2725
2732 ;
2733 ; FORMAT STATEMENTS USED IN PRINT CALLS
2734 ;
2735
2736 .NLIST BEX
2737
2738 004054 045 116 045 CODELM:: .ASCIZ /%N%AUNIT %D1%A TK25 CODE LEVEL P%03%N%N/
2739 .EVEN
2740 004124 130 130 130 HALTM:: .ASCIZ /XXX CMD - TYPE <CR> TO CONTINUE/
2741 004164 103 115 104 CMDPKM:: .ASCIZ /CMD PACKET ADR NOT ON MODULO 4 BOUNDARY: RELOAD!/
2742 .EVEN
2743 004246 104 101 124 WTVERM:: .ASCIZ /DATA COMPARE ERROR/
2744 004271 116 117 040 TOERM:: .ASCIZ /NO TK25 RESPONSE/
2745 004312 125 116 104 SCERM:: .ASCIZ /UNDEFINED SPEC COND/
2746 004336 122 106 103 RFCERM:: .ASCIZ /RFC NON ZERO/
2747 004353 124 113 062 NSSRM:: .ASCIZ /TK25 NOT READY/
2748 004372 122 105 124 RLEXM:: .ASCIZ /RETRY LIMIT EXCEEDED/
2749 004417 125 116 111 ATTNM:: .ASCIZ /UNIT OFF LINE/
2750 004435 106 125 116 FUNRM:: .ASCIZ /FUNCTION REJECT/
2751 004455 106 101 124 FATSM:: .ASCIZ /FATAL SUBSYSTEM ERROR/
2752 004503 116 117 040 NOINTM:: .ASCIZ /NO INTERRUPT/
2753 004520 124 101 120 TSAM:: .ASCIZ /TAPE STATUS ALERT/
2754 004542 124 117 117 TOOMM:: .ASCIZ /TOO MANY INTERRUPTS/
2755 004566 103 101 120 RNYM:: .ASCIZ /CAPSTAN RUNAWAY-GET STATUS RESULTS:/
2756 004632 122 105 103 RERM:: .ASCIZ /RECOVERABLE ERROR/
2757 004654 125 116 122 URERM:: .ASCIZ /UNRECOVERABLE ERROR/
2758 004700 045 116 045 DROPDM:: .ASCIZ /%N%ADROPPED UNIT %D1%N/
2759 004727 045 116 045 AUDRPM:: .ASCIZ /%N%AALL UNITS DROPPED%N%N/
2760 004761 045 116 045 DTAER2:: .ASCIZ "%N%ABYTE:%D4%S2%AWAS:%B8%S2%AS/B:%B8%N"
2761 005030 045 104 064 DTAER3:: .ASCIZ "%D4%A BYTES IN ERROR OUT OF %D4%N"
2762 005072 045 101 116 DTAER4:: .ASCIZ /%ANO DATA READ%N/
2763 005113 045 101 122 DTAER5:: .ASCIZ /%ARECORD TOO LONG: >%04%A BYTES%N/
2764 005155 045 101 122 NURTY1:: .ASCIZ /%ARECOVERED ON RETRY %D2%N/
2765 005211 045 101 125 OFLINM:: .ASCIZ /%AUNIT %D1%A OFF LINE%N/
2766 005241 045 101 107 GETSTM:: .ASCIZ /%AGET STATUS CMD RESULTS:%N/
2767 005275 045 116 000 CRLF:: .ASCIZ /%N/
2768 005300 045 116 045 CRLFSP:: .ASCIZ /%N%S7/
2769 005306 045 116 045 RAMFHR:: .ASCIZ '%N%A ***** CONTROLLER RAM DUMP *****'
2770 005365 045 116 045 RAMIOP:: .ASCIZ '%N%A RAM ADDRESS (OCTAL) = %03%A - %03%N'
2771 005436 045 101 040 RAMPD:: .ASCIZ '%A %03%A '
2772 005450 045 116 045 RAMLIN:: .ASCIZ '%N%N%N'
2773 .LIST BEX
2774 .EVEN
2775

```

```

2777      .SBTTL GLOBAL ERROR REPORT SECTION
2778
2779      ;**
2780      ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX CALLS
2781      ; THAT ARE USED IN MORE THAN ONE TEST. IT ALSO INCLUDES THE ASCII MESSAGES
2782      ; THAT ARE USED BY THE PRINTB AND PRINTX CALLS..
2783      ;--
2784
2785
2786      BGNMSG DTAERM
005460
005460
2792      DTAERM::
005460      PRINTB @STAER1,DEVTBL(R5),PASCNT(R5),RECCNT(R5)
005460      MOV RECCNT(R5),-(SP)
005464      016546 003330      MOV PASCNT(R5),-(SP)
005464      016546 003260      MOV DEVTBL(R5),-(SP)
005470      016546 002536      MOV @STAER1, -(SP)
005474      012746 006140      MOV @4, -(SP)
005500      012746 000004      MOV SP,RO
005504      010600      TRAP C$PNTB
005506      104414      ADD @12,SP
005510      062706 000012
2793      PRINTB @STAER7
005514      MOV @STAER7, -(SP)
005514      012746 006232      MOV @1, -(SP)
005520      012746 000001      MOV SP,RO
005524      010600      TRAP C$PNTB
005526      104414      ADD @4,SP
005530      062706 000004
2794      005534      LET RECCD := R2      ;SAVE R2      MOV R2,RECCD
005534      010237 006546
2795      005540      LET TIME1 := R3      ;SAVE R3      MOV R3,TIME1
005540      010337 003444
2796      005544      LET TIME2 := R4      ;SAVE R4      MOV R4,TIME2
005544      010437 003446
2797      005550      JSR PC,RECTAP      ;RETRIEVE RECORD READ
005550      004737 006602      ;RESTORE R2
2798      005554      LET R2 := RECCD      MOV RECCD,R2
005554      013702 006546
2799      005560      LET RECCD := R3      ;SAVE RECORD READ      MOV R3,RECCD
005560      010337 006546
2800      005564      LET R3 := TIME1      ;RESTORE R3      MOV TIME1,R3
005564      013703 003444
2801      005570      LET R4 := TIME2      ;RESTORE R4      MOV TIME2,R4
005570      013704 003446
2802      005574      PRINTB @STAER6,RECCD      ;PRINT RECORD READ
005574      013746 006546      MOV RECCD, -(SP)
005600      012746 006262      MOV @STAER6, -(SP)
005604      012746 000002      MOV @2, -(SP)
005610      010600      MOV SP,RO
005612      104414      TRAP C$PNTB
005614      062706 000006      ADD @6,SP
2803      005620      EXIT MSG
005620      000167
005622      000000      .WORD J$JMP
2804      .EVEN      .WORD L10002-2-
2805
2806      ENDMSG
005624
005624      L10002:
005624      104423      TRAP C$MSG
2807

```

2808	005626			BGNMSG STAERM			
	005626			STAERM: :			
2809	005626			PRINTB #STAER1,DEVTBL(R5),PASCNT(R5),RECCNT(R5)			
	005626	016546	003330			MOV	RECCNT(R5),-(SP)
	005632	016546	003260			MOV	PASCNT(R5),-(SP)
	005636	016546	002536			MOV	DEVTBL(R5),-(SP)
	005642	012746	006140			MOV	#STAER1, -(SP)
	005646	012746	000004			MOV	#4, -(SP)
	005652	010600				MOV	SP,RO
	005654	104414				TRAP	C\$PNTB
	005656	062706	000012			ADD	#12,SP
2810	005662			PRINTB #STAER7			
	005662	012746	006232			MOV	#STAER7, -(SP)
	005666	012746	000001			MOV	#1, -(SP)
	005672	010600				MOV	SP,RO
	005674	104414				TRAP	C\$PNTB
	005676	062706	000004			ADD	#4,SP
2811	005702			LET R2 := CMDPKT CLR.BY #177740			
	005702	013702	002314			MOV	CMDPKT,R2
	005706	042702	177740			BIC	#177740,R2
2812	005712			LET R2 := R2 - #1			
	005712	005302				DEC	R2
2813	005714			IF R2 EQ #0 THEN			
	005714	005702					
	005716	001016				TST	R2
	005720	004737	006602			BNE	50000\$
2814	005720	004737	006602	JSR PC,RECTAP			
2815	005724			LET RECRED := R3			
	005724	010337	006546				
2816	005730			PRINTB #STAER6,RECRED			
	005730	013746	006546			MOV	RECRED, -(SP)
	005734	012746	006262			MOV	#STAER6, -(SP)
	005740	012746	000002			MOV	#2, -(SP)
	005744	010600				MOV	SP,RO
	005746	104414				TRAP	C\$PNTB
	005750	062706	000006			ADD	#6,SP
2817	005754			ENDIF			
	005754						50000\$:
2818	005754			PRINTX #STAER2			
	005754	012746	006316			MOV	#STAER2, -(SP)
	005760	012746	000001			MOV	#1, -(SP)
	005764	010600				MOV	SP,RO
	005766	104415				TRAP	C\$PNTX
	005770	062706	000004			ADD	#4,SP
2819	005774			PRINTX #STAER3,CMDPKT,@TSDB(R5),MSGPKT*MS.RFC,TSSREG,CTCC			
	005774	013746	003456			MOV	CTCC, -(SP)
	006000	013746	003462			MOV	TSSREG, -(SP)
	006004	013746	002344			MOV	MSGPKT*MS.RFC, -(SP)
	006010	017546	002456			MOV	@TSDB(R5), -(SP)
	006014	013746	002314			MOV	CMDPKT, -(SP)
	006020	012746	006375			MOV	#STAER3, -(SP)
	006024	012746	000006			MOV	#6, -(SP)
	006030	010600				MOV	SP,RO
	006032	104415				TRAP	C\$PNTX
	006034	062706	000016			ADD	#16,SP
2820	006040			PRINTX #STAER4,CMDPKT*2,CMDPKT*4,CMDPKT*6			
	006040	013746	002322			MOV	CMDPKT*6, -(SP)
	006044	013746	002320			MOV	CMDPKT*4, -(SP)


```

2842 .SBTTL GLOBAL SUBROUTINES SECTION
2843 ;**
2844 ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
2845 ; THAT ARE USED IN MORE THAN ONE TEST.
2846 ;--
2847
2848 ;     MODULES TO HANDLE TK25 INTERRUPTS.
2849
2850 006552          BGNSRV TS4INO          ;DEVICE 0.
2851 006552          TS4INO::            LET INTFLG := INTFLG + #1      ;SET INTERRUPT OCCURRED FLAG.
2852 006552 005237 003476          INC          INTFLG
2853 006556          ENDSRV
2854 006556          L10004:
2855 006556          RTI
2856 000002
2857
2858 006560          BGNSRV TS4IN1          ;DEVICE 1.
2859 006560          TS4IN1::            LET INTFLG+2 := INTFLG+2 + #1  ;SET INTERRUPT OCCURRED FLAG.
2860 006560 005237 003500          INC          INTFLG+2
2861 006564          ENDSRV
2862 006564          L10005:
2863 006564          RTI
2864 000002
2865
2866 006566          BGNSRV TS4IN2          ;DEVICE 2.
2867 006566          TS4IN2::            LET INTFLG+4 := INTFLG+4 + #1  ;SET INTERRUPT OCCURRED FLAG.
2868 006566 005237 003502          INC          INTFLG+4
2869 006572          ENDSRV
2870 006572          L10006:
2871 006572          RTI
2872 000002
2873
2874 006574          BGNSRV TS4IN3          ;DEVICE 3.
2875 006574          TS4IN3::            LET INTFLG+6 := INTFLG+6 + #1  ;SET INTEPRUPT OCCURRED FLAG.
2876 006574 005237 003504          INC          INTFLG+6
2877 006600          ENDSRV
2878 006600          L10007:
2879 006600          RTI
2880 000002

```



```

2866
2867 ; SUBROUTINE TO RETRIEVE RECORD COUNT READ FROM TAPE FOR ERROR
2868 ; PRINTS.
2869 ; INPUTS:
2870 ; OUTPUTS: R3 = RECORD COUNT READ
2871 ; REGISTERS: R2, R3, R4
2872 ; CALLS:
2873
2874 006602 RECTAP::IF #MOD.CO SETIN CMDWRD THEN ;READ REV FETCH
      006602 032737 000400 003426 BIT #MOD.CO,CMDWRD
      006610 001430 BEQ 50001$
2875 006612 LET R2 := MSGPKT*MS.RFC + DATARD ;FIND LAST READ AD.
      006612 013702 002344 MOV MSGPKT*MS.RFC,R2
      006616 063702 003416 ADD DATARD,R2
2876 006622 IF #BIT00 SETIN R2 THEN ;ODD AD., REASSEMBLE
      006622 032702 000001 BIT #BIT00,R2
      006626 001417 BEQ 50002$
2877 006630 LET R2 := R2 + #1 ;REC COUNT STARTING
      006630 005202 INC R2
2878 006632 LET R3 :B= (R2) CLR.BY #177400 ;WITH UPPER BYTE FETCH
      006632 111203 MOVB (R2),R3
      006634 142703 177400 BICB #177400,R3
2879 006640 LET R3 := SWAP R3 ;
      006640 000303 SWAB R3
2880 006642 LET R2 := R2 - #1 ;LOWER BYTE AD.
      006642 005302 DEC R2
2881 006644 IFB SWBFLG NE #0 THEN
      006644 105737 003524 TSTB SWBFLG
      006650 001401 BEQ 50003$
2882 006652 LET R2 := R2 - #1 ;LOWER BYTE AD.
      006652 005302 ON SWAP DEC R2
2883 006654 ENDIF
2884 006654 LET R4 :B= (R2) CLR.BY #177400 ;FETCH LOWER
      006654 111204 MOVB (R2),R4
      006656 142704 177400 BICB #177400,R4
2885 006662 LET R3 := R3 OR R4 ;MERGE BYTES
      006662 050403 BIS R4,R3
2886 006664 ELSE
      006664 000401 BR 50004$
      006666 50002$:
2887 006666 LET R3 := (R2) ;EVEN AD. FETCH
      006666 011203 MOV (R2),R3
2888 006670 ENDIF
      006670 50004$:
2889 006670 ELSE
      006670 000402 BR 50005$
      006672 50001$:
2890 006672 LET R3 := @DATARD ;READ FWD FETCH
      006672 017703 174520 MOV @DATARD,R3
2891 006676 ENDIF
      006676 50005$:
2892
2893 006676 000207 RTS PC

```

```

2896      :      SUBROUTINE TO STORE A SET CHARACTERISTIC COMMAND AS
2897      :      THE FIRST ENTRY IN THE SEQUENCE TABLE.
2898      :      INPUTS:
2899      :      OUTPUTS:
2900      :      REGISTERS:
2901      :      CALLS:
2902
2903 006700      SETCH:: LET R1 := #CMDSEQ      ;INIT COMMAND SEQUENCE TABLE POINTER.
      006700      012701      003542      MOV      #CMDSEQ,R1
2904 006704      012721      140004      MOV      #SCH,(R1)+      ;THIS CODE SETS UP A SET CHARACTERISTIC
2905 006710      012721      000040      MOV      #DFTSCH,(R1)+ ;COMMAND AS THE FIRST COMMAND IN THE
2906 006714      012721      000001      MOV      #1,(R1)+      ;SEQUENCE TABLE.
2907 006720      005721      TST      (R1)+      ;SKIP PATTERN LOCATION.
2908 006722      000207      RTS      PC
2909
2910
2911
2912
2913      :      SUBROUTINE TO STORE A REWIND COMMAND IN THE SEQUENCE TABLE
2914      :      INPUTS:
2915      :      OUTPUTS:
2916      :      REGISTERS:
2917      :      CALLS:
2918
2919 006724      SETRW:: LET (R1)+ := #RWD      ;CMD = REWIND.
      006724      012721      102010      MOV      #RWD,(R1)+
2920 006730      LET (R1)+ := #1      ;BRF.
      006730      012721      000001      MOV      #1,(R1)+
2921 006734      LET (R1)+ := #1      ;# OF OPERATIONS.
      006734      012721      000001      MOV      #1,(R1)+
2922 006740      005721      TST      (R1)+      ;SKIP PATTERN.
2923 006742      000207      RTS      PC      ;RETURN

```

```

2925      ; SUBROUTINE TO EXECUTE ALL COMMANDS IN THE SEQUENCE TABLE ON ALL
2926      ; DEVICES.
2927      ; INPUTS:
2928      ; OUTPUTS:      R2 = TERMINATION INDICATOR (0=END OF TABLE,1=EOT)
2929      ; REGISTERS:
2930      ; CALLS:      CMDAC,SETUP,EXSUB,CKHAE,NEXTU,FIRSTU,VFYDAT.
2931
2932      EXALL:: LET R1 := @CMDSEQ      ;INIT SEQUENCE TABLE POINTER.
                MOV      @CMDSEQ,R1
2933      006744 012701 003542      WHILE (R1) NE @END DO      ;WHILE THERE ARE CMDs IN THE SEQUENCE TABLE.
                50006$:
                006750      021127 177777      CMP      (R1),@END
                006754 001527      BEQ      50007$
2934      006756 004737 007706      JSR PC,SETUP      ;GO SETUP THE COMMAND BLOCK.
2935      006762      WHILE NCNT LT NCNT1 DO      ;WHILE THERE ARE RECORDS REMAINING:
                006762 023737 003420 003422      50010$:
                006770 002116      CMP      NCNT,NCNT1
                006772 004737 007600      BGE      50011$
2936      006772 004737 007600      JSR PC,CMDAC      ;STORE CMD ASCII IN ERROR MESSAGE.
2937      006776 105737 003521      IFB RANDOM NE @0 THEN      ;IF IN RANDOM MODE:
                007002 001435      TSTB     RANDOM
                007004      BEQ      50012$
2938      007004 023727 003426 104005      IF CMDWRD EQ @WRT THEN      ;IF CMD IS A WRITE THEN:
                007012 001031      CMP      CMDWRD,@WRT
                007014      BNE      50013$
2939      007014 105737 003522      IFB VFYFLG EQ @0 THEN      ;IF DATA IS NOT
                007020 001026      TSTB     VFYFLG      TO BE VERIFIED THEN:
                007022      BNE      50014$
2940      007022 063737 003442 003440      LET RANB := RANB + RANS      ;GENERATE
                007030 063737 003440 003442      LET RANS := RANS + RANB      ;RANDOM
                007036 013737 003442 003424      LET BRFCNT := RANS      ;LENGTH
                007044 043737 003436 003424      LET BRFCNT := BRFCNT CLR.BY LENMSK ;MASK RANDOM LENGTH.
                007052 023727 003424 000022      IF BRFCNT LT @18. THEN      ;DO NOT ALLOW BYTE COUNT OF LESS THAN 18.
                007060 002003      CMP      BRFCNT,@18.
                007062      BGE      50015$
2945      007062 012737 000022 003424      LET BRFCNT := @18. ;CHANGE COUNT OF 0-17 TO 18.
                007070      MOV      @18.,BRFCNT
2946      007070      ENDF
                50015$:
2947      007070 013737 003424 002322      LET CMDPKT.CP.CNT := BRFCNT ;MOVE BRFCNT TO CMD PACKET.
                007076      MOV      BRFCNT,CMDPKT.CP.CNT
2948      007076      ENDF
                50014$:
2949      007076      ENDF
                50013$:
2950      007076      ENDF
                50012$:
2951      007076 004737 007240      JSR PC,EXSUB      ;ISSUE CMD TO ALL,AWAIT INTS,CHECK STATUS.
2952      007102 004737 017706      JSR PC,CKHAE      ;CHECK HALT AFTER EACH CMD FLAG.
2953      007106 012702 000001      LET R2 := @1      ;SET ALL UNITS AT BOT/EOT.
                007112 004737 017300      MOV      @1,R2
2954      007112 004737 017300      JSR PC,FIRSTU      ;FIND FIRST UNIT.
2955      007116      WHILE DEVTBL(R5) NE @END DO ;WHILE THERE ARE MORE UNITS:

```

007116						50016\$:
007116	026527	002536	177777			CMP DEVTBL(R5),#END
007124	001426					BEQ 50017\$
2956 007126					IF #MOD.CO SETIN CMDWRD THEN ;IF CMD IS REVERSE THEN:	
007126	032737	000400	003426			BIT #MOD.CO,CMDWRD
007134	001406					BEQ 50020\$
2957 007136					IF #X0.BOT NOTSETIN EOTFLG(R5) THEN ;IF NOT AT BOT THEN:	
007136	032765	000002	003506			BIT #X0.BOT,EOTFLG(R5)
007144	001001					BNE 50021\$
2958 007146					LET R2 := #0	;CLEAR EOT/BOT FLAG.
007146	005002					CLR R2
2959 007150					ENDIF	
007150					ELSE	50021\$:
2960 007150						;ELSE IF CMD IS NOT REVERSE:
007150	000411					BR 50022\$
007152						50020\$:
2961 007152					IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN	
007152	032765	000001	003506			BIT #X0.EOT,EOTFLG(R5)
007160	001404					BEQ 50023\$
007162	032737	000001	003426			BIT #CMD.CO,CMDWRD
007170	001001					BNE 50024\$
007172						50023\$:
2962						;IF NOT AT EOT OR NOT A MOTION CMD THEN:
2963 007172					LET R2 := #0	;CLEAR EOT/BOT FLAG.
007172	005002					CLR R2
2964 007174					ENDIF	
007174					ENDIF	50024\$:
2965 007174						50022\$:
007174					JSR PC,NEXTU	;FIND NEXT UNIT
2966 007174	004737	017346			ENDDO	
2967 007200						BR 50016\$
007200	000746					50017\$:
007202					IF R2 EQ #1 THEN	;IF ALL UNIT ARE AT EOT/BOT THEN:
2968 007202						CMP R2,#1
007202	020227	000001				BNE 50025\$
007206	001001				BR EXARTN	;RETURN WITH R2 = #1.
2969 007210	000412				ENDIF	
2970 007212						50025\$:
007212					LET NCNT := NCNT + #1	;UPDATE RECORD COUNT.
2971 007212						INC NCNT
007212	005237	003420			LET PCMDWD := CMDWRD	;SAVE PREVIOUS COMMAND WORD.
2972 007216						MOV CMDWRD,PCMDWD
007216	013737	003426	003432		ENDDO	
2973 007224						BR 50010\$
007224	000656					50011\$:
007226					JSR PC,VFYDAT	;IF LAST CMD WAS A WRITE VERIFY, THEN GO
2974 007226	004737	016142				;VERIFY THE LAST N RECORDS OF DATA.
2975					ENDDO	
2976 007232						BR 50006\$
007232	000646					50007\$:
007234					LET R2 := #0	;SET NORMAL RETURN INDICATOR.
2977 007234						CLR R2
007234	005002					
2978 007236	000207				EXARTN: RTS PC	;RETURN.
2979						
2980						
2981						

```

2982      ; SUBROUTINE TO ISSUE COMMAND TO ALL DEVICES, WAIT FOR
2983      ; ALL INTERRUPTS, AND CHECK ALL STATUS.
2984      ; INPUTS:
2985      ; OUTPUTS:
2986      ; REGISTERS:
2987      ; CALLS: EXCUTE,GOWAIT,NEXTU,FIRSTU.
2988
2989 007240 004737 017300 EXSUB:: JSR PC,FIRSTU ;SET UP FOR FIRST UNIT.
2990 007244 WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
007244 50026$:
007244 CMP DEVTBL(R5),#END
007252 001465 BEQ 50027$
2991 007254 IF #MOD.CO SETIN CMDWRD THEN ;IF CMD IS REVERSE THEN:
007254 032737 000400 003426 BIT #MOD.CO,CMDWRD
007262 001421 BEQ 50030$
2992 007264 IF #XO.BOT NOTSETIN EOTFLG(R5) THEN ;IF NOT AT BOT
007264 032765 000002 003506 BIT #XO.BOT,EOTFLG(R5)
007272 001014 BNE 50031$
2993 007274 IF #XO.EOT SETIN EOTFLG(R5) THEN ;BUT IF AT EOT
007274 032765 000001 003506 BIT #XO.EOT,EOTFLG(R5)
007302 001406 BEQ 50032$
2994 007304 IFB ALLEOT NE #0 THEN ;AND ALL OTHERS AT EOT
007304 105737 003531 TSTB ALLEOT
007310 001402 BEQ 50033$
2995 007312 004737 010654 JSR PC,EXCUTE ;THEN EXECUTE REV CMD
2996 007316 ENDIF ;IF NOT ALL AT EOT, FREEZE UNIT(S) AT EOT
007316 50033$:
2997 007316 ELSE ;IF NOT AT BOT AND
007316 000402 BR 50034$
007320 50032$:
2998 007320 004737 010654 JSR PC,EXCUTE ;NOT AT EOT, EXEC REV CMD
2999 007324 ENDIF
007324 50034$:
3000 007324 ENDIF
007324 50031$:
3001 007324 ELSE ;ELSE IF CMD IS NOT REVERSE:
007324 000435 BR 50035$
007326 50030$:
3002 007326 IF CMDLG EQ #2 AND #XO.BOT SETIN EOTFLG(R5) THEN
007326 023727 003434 000002 CMP CMDLG,#2
007334 001011 BNE 50036$
007336 032765 000002 003506 BIT #XO.BOT,EOTFLG(R5)
007344 001405 BEQ 50036$
3003 ;CLEAR BAD SPOT COUNTS WHEN WRITING FROM BOT
3004 007346 LET BTPT := BTADDR(R5)
007346 016537 002550 003516 MOV BTADDR(R5),BTPT
3005 007354 LET #BTPT := #0
007354 005077 174136 CLR #BTPT
3006 007360 ENDIF
007360 50036$:
3007 007360 IF #XO.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN
007360 032765 000001 003506 BIT #XO.EOT,EOTFLG(R5)
007366 001404 BEQ 50037$
007370 032737 000001 003426 BIT #CMD.CO,CMDWRD
007376 001003 BNE 50040$
007400 50037$:
3008 ;IF NOT AT EOT OR NOT A MOTION CMD THEN:

```

```

3009 007400 004737 010654      JSR PC,EXCUTE      ;ISSUE CMD TO TK25.
3010 007404      ELSE
      007404 000405      BR      50041$
      007406      50040$:
3011 007406      IFB ALLEOT NE #0 THEN
      007406 105737 003531      TSTB     ALLEOT
      007412 001402      BEQ      50042$
3012 007414 004737 010654      JSR PC,EXCUTE
3013 007420      ENDIF
      007420      50042$:
3014 007420      ENDIF
      007420      50041$:
3015 007420      ENDIF
      007420      50035$:
3016 007420 004737 017346      JSR      PC,NEXTU      ;FIND NEXT UNIT IN TEST CYCLE.
3017 007424      ENDDO
      007424 000707      BR      50026$
      007426      50027$:
3018 007426      IFB RPTFLG NE #0 THEN      ;IF REPORT HAS BEEN REQUESTED THEN:
      007426 105737 003523      TSTB     RPTFLG
      007432 001403      BEQ      50043$
3019 007434      LET RPTFLG :B= #0      ;CLR THE FLAG.
      007434 105037 003523      CLRB     RPTFLG
3020 007440      DORPT      ;PRINT THE PERFORMANCE REPORT.
      007440 104424      TRAP     C$DRPT
3021 007442      ENDIF      ;
      007442      50043$:
3022 007442 004737 017300      JSR PC,FIRSTU      ;SET UP FOR FIRST UNIT.
3023 007446      WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
      007446      50044$:
      007446 026527 002536 177777      CMP      DEVTBL(R5),#END
      007454 001450      BEQ      50045$
3024 007456      IF #MOD.CO SETIN CMDWRD THEN ;IF CMD IS REVERSE THEN:
      007456 032737 000400 003426      BIT      #MOD.CO,CMDWRD
      007464 001421      BEQ      50046$
3025 007466      IF #X0.BOT NOTSETIN EOTFLG(R5) THEN ;IF NOT AT BOT
      007466 032765 000002 003506      BIT      #X0.BOT,EOTFLG(R5)
      007474 001014      BNE      50047$
3026 007476      IF #X0.EOT SETIN EOTFLG(R5) THEN ;BUT IF AT EOT
      007476 032765 000001 003506      BIT      #X0.EOT,EOTFLG(R5)
      007504 001406      BEQ      50050$
3027 007506      IFB ALLEOT NE #0 THEN      ;AND ALL OTHERS AT EOT
      007506 105737 003531      TSTB     ALLEOT
      007512 001402      BEQ      50051$
3028 007514 004737 011274      JSR PC,GOWAIT      ;THEN WAIT FOR CMD END
3029 007520      ENDIF      ;IF NOT ALL AT EOT, DO NOT WAIT
      007520      50051$:
3030 007520      ELSE      ;NOT AT BOT, AND NOT AT EOT
      007520 000402      BR      50052$
      007522      50050$:
3031 007522 004737 011274      JSR PC,GOWAIT      ;WAIT FOR INT,CHECK STATUS.
3032 007526      ENDIF
      007526      50052$:
3033 007526      ENDIF
      007526      50047$:
3034 007526      ELSE      ;ELSE IF CMD IS FORWARD:
      007526 000420      BR      50053$

```

```

007530
3035 007530 032765 000001 003506 IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN
007536 001404 BIT #X0.EOT,EOTFLG(R5)
007540 032737 000001 003426 BEQ 50054$
007546 001003 BIT #CMD.CO,CMDWRD
007550 BNE 50055$
50054$:
3036 ;IF NOT AT EOT OR NOT A MOTION CMD THEN:
3037 007550 004737 011274 JSR PC,GOWAIT ;WAIT FOR INT,CHECK STATUS.
3038 007554 000405 ELSE
007556 BR 50056$
3039 007556 IFB ALLEOT NE #0 THEN 50055$:
007556 105737 003531 TSTB ALLEOT
007562 001402 BEQ 50057$
3040 007564 004737 011274 JSR PC,GOWAIT
3041 007570 ENDF
007570 ENDF 50057$:
3042 007570 ENDF 50056$:
007570 ENDF
3043 007570 ENDF
007570
3044 007570 004737 017346 JSR PC,NEXTU ;FIND NEXT UNIT IN TEST CYCLE.
3045 007574 000724 ENDDO
007576 BR 50044$
3046 007576 000207 RTS PC ;RETURN.
50045$:

```

```

3048 ; THIS SUBROUTINE STORES THE ASCII FOR THE CURRENT COMMAND AND PREVIOUS
3049 ; COMMAND IN THE STANDARD ERROR MESSAGE. ON ENTRY LOCATION CMDWRD
3050 ; CONTAINS CURRENT CMD AND LOCATION PCMDWD CONTAINS PREVIOUS CMD.
3051 ; INPUTS:
3052 ; OUTPUTS:
3053 ; REGISTERS: R3, R4.
3054 ; CALLS: GCMDA
3055
3056 CMDAC:: LET R4 := CMDWRD ;R4 = CMD BINARY.
                                MOV CMDWRD,R4
007600 013704 003426 JSR PC,GCMDA ;GET CMD ASCII.
007600 004737 007654 MOVB (R3)+,STAER1+2 ;MOVE CMD ASCII
3057 007604 004737 007654 MOVB (R3)+,STAER1+3 ;
3058 007610 112337 006142 MOVB (R3)+,STAER1+4 ;INTO MSG.
3059 007614 112337 006143 LET R4 := PCMDWD ;R4 = PREVIOUS CMD BINARY.
3060 007620 111337 006144 JSR PC,GCMDA ;GET CMD ASCII.
                                MOV PCMDWD,R4
3061 007624 013704 003432 NOP
3062 007630 004737 007654 LET STAER7+24 :B= (R3)+ ;MOVE CMD ASCII
                                MOVB (R3)+,STAER7+24
3063 007634 000240 LET STAER7+25 :B= (R3)+ ;
3064 007636 112337 006256 LET STAER7+26 :B= (R3) ;INTO MSG.
                                MOVB (R3)+,STAER7+25
3065 007642 112337 006257 RTS PC ;RETURN. GO EXECUTE NEXT FUNCTION.
3066 007646 111337 006260
3067 007652 000207
3068
3069
3070
3071 ; SUBROUTINE TO FIND THE ASCII EQUIVILENT OF THE COMMAND IN R4.
3072 ; ADDRESS OF ASCII 1ST WORD IS RETURNED IN R3.
3073 ; INPUTS: R4 = PRESENT COMMAND WORD.
3074 ; OUTPUTS: R3 = ADDRESS OF PRESENT COMMAND ASCII.
3075 ; REGISTERS:
3076 ; CALLS:
3077
3078 GCMDA:: LET R3 := #0 ;INIT CMD TBL POINTER.
                                CLR R3
3079 007654 005003 WHILE CMTDBL(R3) NE R4 DO ;UNTIL CURRENT CMD IS FOUND:
                                50060$:
007656 026304 003644 CMP CMTDBL(R3),R4
007662 001403 BEQ 50061$
3080 007664 062703 000002 LET R3 := R3 + #2 ;SEARCH CMD TABLE.
                                ADD #2,R3
3081 007670 000772 ENDDO BR 50060$
                                50061$:
3082 007672 010304 LET R4 := R3 MOV R3,R4
3083 007674 006203 LET R3 := R3 SHIFT -1 ;POINT TO ASCII FOR THAT COMMAND
                                ASR R3
3084 007676 060403 ADD R4,R3
3085 007700 062703 003732 ADD #CMDASC,R3
3086 007704 000207 RTS PC ;RETURN.

```



```

3088      ; THIS SUBROUTINE LOADS THE TK25 COMMAND PACKET FROM ONE
3089      ; ENTRY IN THE SEQUENCE TABLE.
3090      ; INPUTS:
3091      ; OUTPUTS:
3092      ; REGISTERS:      R2, R3.
3093      ; CALLS:          GENPAT.
3094
3095      SETUP:: LET CMDLG := #0                ;CLR CMD LOGGING CODE(DISABLES LOGGING)
          007706 005037 003434                CLR      CMDLG
          007706 012137 002314                MOV      (R1)+,CMDPKT
3096      007712 012137 002314                MOV      (R1),CMDPKT+CP.CNT
3097      007716 011137 002322                MOV      (R1),BRFCNT
3098      007722 011137 003424                MOV      CMDPKT,R2
3099      007726 013702 002314                BIC      @NCMD.C,R2
3100      007732 042702 177740                MOV      R2,R3
3101      007736 010203                        SUB      @CMD.C3,R3
3102      007740 162703 000010                BNE      2$
3103      007744 001003                        MOV      (R1),CMDPKT+2
3104      007746 011137 002316                BR       3$
3105      007752 000461                        2$: IF CMDPKT EQ @WTM THEN
3106      007754                                ;IF CMD IS A WRITE TAPE MARK THEN:
          007754 023727 002314 100011        CMP      CMDPKT,@WTM
          007762 001003                        BNE      50062$
3107      007764                                ;WTM LOGGING CODE IS 2.
          007764 012737 000002 003434        MOV      @2,CMDLG
3108      007772                                50062$:
          007772                                ;IS IT A READ?
3109      007772 010203                        MOV      R2,R3
3110      007774 162703 000001                SUB      @CMD.CO,R3
3111      010000 001017                        BNE      1$
3112      010002 013737 003416 002316        MOV      DATARD,CMDPKT+CP.ADL
3113      010010 032737 000400 002314        IF @MOD.CO SETIN CMDPKT THEN
          010016 001404                                ;LOGGING CODE IS 4.
3114      010020                                ;ELSE - IF CMD IS A READ FWD:
          010020 012737 000004 003434        LET CMDLG := #4
          010026 000403                        BR       50063$
3115      010030                                ;LOGGING CODE IS 6.
          010030 012737 000006 003434        LET CMDLG := #6
          010036 000403                        BR       50064$
3116      010036                                ;CONTINUE.
          010036 000427                        ;IS IT
3117      010036 010203                        MOV      R2,R3
3118      010040 162703 000004                SUB      @CMD.C2,R3
3119      010042 001011                        BNE      4$
3120      010046 001011                        LET CMDPKT+CP.ADL := @SCHBK
          010050 012737 002446 002316        MOV      @SCHCNT,CMDPKT+CP.CNT
          010050 012737 000010 002322        LET SCHBK+6 := (R1)
          010056 011137 002454                ;CONTINUE.
          010064 000412                        ;IS IT
3125      010070                                ;A DIAGNOSTIC (DIA) CMD?
3126      010072 010203                        MOV      R2,R3
3127      010074 162703 000006                SUB      @CMD.C1!CMD.C2,R3
3128      010100 001006                        BNE      3$
3129      010102 012737 000020 002322        MOV      @DIACNT,CMDPKT+CP.CNT
3130      010110 012737 003414 002316        MOV      @DIABLK,CMDPKT+CP.ADL
          010110 003414 002316                ;LOAD BUFFER EXTENT.
          010110 003414 002316                ;LOAD BUFFER ADR LOW.

```

```

3131 010116 005721          3$:  TST      (R1)+      ;POINT TO N (NUMBER OF TIMES TO EXECUTE THIS INSTRUC
3132 010120          LET NCNT1 := (R1)+  ;SAVE NUMBER OF OPERATIONS
                                MOV      (R1)+,NCNT1
3133 010124          LET NCNT := #0      ;CLEAR OPERATION COUNTER.
                                CLR      NCNT
3134 010130          MOV      (R1)+,PATERN  ;SAVE PATTERN CODE FOR CURRENT CMD.
3135 010134          MOV      R2,R3        ;IS IT
3136 010136          SUB      @CMD.CO!CMD.C2,R3 ;A WRITE?
3137 010142          BNE      5$           ;BR IF NOT.
3138 010144          MOV      DATAWT,CMDPKT+CP.ADL ;LOAD WRITE BUFFER LO ORDER.
3139 010152          JSR      PC,GENPAT     ;GO GENERATE THE WRITE PATTERN.
3140 010156          LET CMDLG := #2      ;WRITE LOGGING CODE IS 2.
                                MOV      #2,CMDLG
3141 010164          5$:  IF @Vfy.C SETIN CMDPKT THEN ;IF DATA VERIFICATION IS REQUIRED:
                                BIT      @Vfy.C,CMDPKT
3142 010174          LET VfyFLG :B= #1     ;SET VERIFY FLAG.
                                BEQ      50065$
3143 010202          BIC      @Vfy.C,CMDPKT ;CLEAR VERIFY BIT(NOT USED BY HARDWARE).
3144 010210          ELSE                 ;IF DATA VERIFICATION IS NOT REQUIRED:
                                BR       50066$
3145 010212          LET VfyFLG :B= #0     ;CLR VERIFY FLAG.
3146 010216          ENDIF                CLRB   VfyFLG
3147 010216          LET PCMDWD := CMDWRD   ;SAVE PREVIOUS CMD WORD.
3148 010224          LET CMDWRD := CMDPKT   ;SAVE PRESENT CMD WORD.
3149 010232          IFB SWBFLG NE #0 THEN ;IF SWAP BYTES IS ENABLED:
3150 010240          LET CMDPKT := CMDPKT SET.BY @SWB.C ;SET SWAP BIT IN COMMAND.
3151 010246          ENDIF                BIS   @SWB.C,CMDPKT
3152 010246          BIC      @BRF.C,CMDPKT ;CLR BRF BIT (INTERNAL ONLY).
3153 010254          LET CMDSAV := CMDPKT   ;SAVE 1ST WORD OF COMMAND PACKET.
3154 010262          RTS      PC           MOV   CMDPKT,CMDSAV

```

```

3156 ; THIS SUBROUTINE SETS UP AND CALLS THE APPROPRIATE SUBROUTINE TO GENERATE
3157 ; THE DESIRED PATTERN FOR THE WRITE AND WRITE/VERIFY COMMANDS.
3158 ; INPUTS:
3159 ; OUTPUTS:
3160 ; REGISTERS: R2, R3, R4.
3161 ; CALLS: PATRO - PATR7
3162
3163 GENPAT:: LET R3 := PATERN SHIFT 1 ;SETUP PATTERN ROUTINE POINTER
          010264 013703 003454 MOV PATERN,R3
          010264 006303 ASL R3
3164 010272 LET R4 := BRFCNT * #1 ;SET LENGTH OF WRITE BFR
          010272 013704 003424 MOV BRFCNT,R4
          010276 005204 INC R4
3165 010300 LET R4 := R4 CLR.BY #1 ;ROUNDED UP TO NEXT WORD
          010300 042704 000001 BIC #1,R4
3166 010304 LET R4 := R4 - #2 ;WITH FIRST WORD RESERVED
          010304 162704 000002 SUB #2,R4
3167 010310 LET R2 := DATAWT * #2 ;FOR RECORD COUNT
          010310 013702 003414 MOV DATAWT,R2
          010314 062702 000002 ADD #2,R2
3168 010320 JSR PC,@PATTBL(R3) ;GO GENERATE THE APPROPRIATE PATTERN.
3169 010324 RTS PC ;RETURN TO SETUP SUBROUTINE.
3170
3171 ;TK25 WRITE PATTERN LOOKUP TABLE. USED TO JSR TO THE
3172 ;CORRECT DATA PATTERN GENERATING ROUTINE.
3173
3174 PATTBL: PATRO ; INCREMENTING PATTERN, 0 - 377
3175 PATR1 ; ALL ONES PATTERN
3176 PATR2 ; ALL ZEROES PATTERN
3177 PATR3 ; '1' BIT SHIFT, RIGHT TO LEFT
3178 PATR4 ; '0' BIT SHIFT, RIGHT TO LEFT
3179 PATR5 ; ALTERNATE '0' & '1' WITH ALT. BYTES COMPL.
3180 PATR6 ; ALTERNATE BYTES OF 000 AND 377
3181 PATR7 ; RANDOM PATTERN.
3182 PATR8 ; DUMMY. NO PATTERN, JUST EXITS.
3183
3184 ;INCREMENTING PATTERN. 0 - 377.
3185
3186 PATRO:: LET R3 := #400
          010350 012703 000400 MOV #400,R3
3187 010350 012703 000400
3188 1$: LET R4 := R4 - #2 ;DECREMENT WORD COUNT.
          010354 162704 000002 SUB #2,R4
          010360 100411 BMI 2$ ;BR IF DONE.
3189 010362 LET (R2)* := R3 ;STORE DATA WORD.
          010362 010322 MOV R3,(R2)*
3190 010362 010322
3191 LET R3 := R3 + #1002 ;UPDATE PATTERN.
          010364 062703 001002 ADD #1002,R3
          010364 062703 001002
3192 IF R3 EQ #1000 THEN ;IF PATTERN HAS WRAPPED AROUND THEN:
          010370 020327 001000 CMP R3,#1000
          010374 001002 BNE 50070$
3193 LET R3 := #400 ;INIT THE PATTERN AGAIN.
          010376 012703 000400 MOV #400,R3
          010376 012703 000400
3194 010402 ENDF
          010402 50070$:
3195 010402 000764 BR 1$ ;DO IT AGAIN.
3196 010404 000207 RTS PC ;RETURN.
          010404 000207

```

```

3197
3198
3199
3200 010406 012703 177777
3201 010412 162704 000002
    010412 100402
3202 010416 010322
3203 010420 000773
3204 010422 000207
3205
3206 010424 000207

```

```

;ALL ONE'S PATTERN.
PATR1:: MOV    #1,R3
ZROPAT: LET R4 := R4 - #2
        BMI    1$
        MOV    R3,(R2)+
        BR     ZROPAT
1$:     RTS    PC

```

```

;ALL ONES PATTERN;.
;DECREMENT BYTE COUNT.
        SUB    #2,R4
;DONE?,BR IF YES.
;IF NOT LOAD NEXT BYTE WITH PATTERN.
;DO IT AGAIN.
;RETURN.

```

```

3208                                     ;ALL ZEROES PATTERN.
3209
3210 010426 005003                                     PATR2:: CLR    R3                ;CLR PATTERN REGISTER.
3211 010430 004737 010412                          JSR    PC,ZROPAT            ;GO GENERATE IT.
3212 010434 000207                                     RTS    PC                    ;RETURN.
3213
3214                                     ;ONE BIT WALKING FROM R TO L IN A FIELD OF ZEROES.
3215
3216 010436 012703 000401                          PATR3:: MOV    #401,R3        ;INIT PATTERN REGISTER.
3217 010442                                WLKZRO: LET R4 := R4 - #2    ;DECREMENT WORD COUNT.
3218 010446 162704 000002                                SUB    #2,R4
3219 010450 010322                                BMI    1$                    ;BR IF DONE.
3220 010452 006303                                MOV    R3,(R2)+             ;LOAD DATA.
3221 010454 005503                                ASL    R3                    ;SHIFT PATTERN.
3222 010456 000771                                ADC    R3                    ;ADD CARRY BACK INTO PATTERN.
3223 010460 000207 1$: RTS    PC                    ;DO IT AGAIN.
3224                                     ;RETURN.
3225                                     ;ZERO BIT WALKING FROM R TO L IN A FIELD OF 1'S.
3226
3227 010462 012703 177376                          PATR4:: MOV    #177376,R3    ;INIT PATTERN REGISTER.
3228 010466 004737 010442                          JSR    PC,WLKZRO            ;GO GENERATE ;IT.
3229 010472 000207                                     RTS    PC                    ;RETURN.
3230
3231                                     ;ALTERNATING ONE AND ZERO BITS WITH ALTERNATE BYTES
3232                                     ;COMPLEMENTED.
3233
3234 010474 012703 125125                          PATR5:: MOV    #125125,R3    ;INIT PATTERN REGISTER.
3235 010500 004737 010412                          JSR    PC,ZROPAT            ;GO GENERATE IT.
3236 010504 000207                                     RTS    PC                    ;RETURN.
3237
3238                                     ;ALTERNATING BYTES OF 000 AND 377.
3239
3240 010506 012703 177400                          PATR6:: MOV    #177400,R3    ;INIT PATTERN REGISTER.
3241 010512                                1$: LET R4 := R4 - #2    ;DECREMENT WORD COUNT.
3242 010516 162704 000002                                SUB    #2,R4
3243 010520 010322                                BMI    2$                    ;BR IF DONE.
3244 010522 000773                                MOV    R3,(R2)+             ;LOAD DATA.
3245 010524 000207                                BR     1$                    ;DO IT AGAIN.
3246                                2$: RTS    PC                    ;RETURN.
3247                                     ;RANDOM PATTERN GENERATOR
3248
3249 010526                                PATR7::
3250 010526 012737 001233 010644                          MOV    #RS1,RAN1            ;SET UP THE SEED
3251 010534 012737 007622 010646                          MOV    #RS2,RAN2            ;SET UP THE SEED
3252 010542 012737 000000 010650                          MOV    #RS3,RAN3            ;SET UP THE SEED
3253 010550                                1$: LET R4 := R4 - #2    ;DECREMENT WORD COUNT
3254 010554 162704 000002                                SUB    #2,R4
3255 010556 100432                                BMI    GIT                    ;BR IF DONE.
3256 010560 010346                                MOV    R3,-(SP)              ;MOVE THE FIRST SEED INTO R3
3257 010564 000241 010644                                MOV    RAN1,R3               ;CLEAR THE CARRY FLAG
3258 010566 005337 010650                                CLC                            ;DECREMENT THE THIRD SEED
3259 010572 006103                                DEC    RAN3
3260 010574 006103                                ROL    R3
3261 010576 063703 010646                                ROL    R3
                                                ADD    RAN2,R3                ;ADD THE SECOND SEED TO R3

```

```

3262 010602 010337 010644      MOV     R3,RAN1      ;PUT IT ALL IN THE FIRST SEED
3263 010606 063703 010650      ADD     RAN3,R3      ;PUT THE THIRD SEED INTO R3
3264 010612 006103              ROL     R3           ;
3265 010614 006103              RCL     R3           ;
3266 010615 063703 010646      ADD     RAN2,R3      ;ADD THE SECOND SEED TO R3
3267 010622 006103              ROL     R3           ;
3268 010624 006103              ROL     R3           ;
3269 010626 010337 010646      MOV     R3,RAN2      ;PUT IT IN THE SECONG SEED
3270 010632 012603              MOV     (SP)+,R3     ;RESTORE R3
3271 010634 013722 010646      MOV     RAN2,(R2)+   ;PUT # IN BUFFER
3272 010640 000743              BR      1$          ;CONTINUE
3273 010642 000207      GIT:    RTS         PC          ;RETURN
3274
3275 010644 000000      RAN1::  .WORD      0
3276 010646 000000      RAN2::  .WORD      0
3277 010650 000000      RAN3::  .WORD      0
3278              001233      RS1     ==         1233
3279              007622      RS2     ==         7622
3280              000000      RS3     ==         0
3281
3282              ;          NO PATTERN GENERATION.
3283
3284 010652 000207      PATR8:: RTS         PC          ;RETURN.

```

```

3286 ; THIS SUBROUTINE INITIATES TK25 COMMAND EXECUTION
3287 ; AND CHECKS FOR TK25 RESPONSE.
3288 ; INPUTS:
3289 ; OUTPUTS:
3290 ; REGISTERS: R2, R3.
3291 ; CALLS: DROPU, MOVMSG, FIRSTU, NEXTU, WSSR.
3292
3293 010654 EXCUTE::LET TIME1 := #-1 ;INIT TIMEOUT COUNTER.
010654 012737 177777 003444 MOV #-1,TIME1
3294 010662 REPEAT ;WAIT -
010662 ;UPDATE TIMEOUT COUNTER.
3295 010662 LET TIME1 := TIME1 - #1 ;50071$:
010662 005337 003444 DEC TIME1
3296 010666 IF TIME1 EQ #0 THEN ;IF TIMED OUT:
010666 005737 003444 TST TIME1
010672 001011 BNE 50072$
3297 010674 004737 011726 JSR PC,MOVMSG ;MOVE CURRENT PACKET MSG.
3298 010700 104455 ERRDF 2,NSSRM,STAERM ;REPORT TK25 NOT READY
010700 000002 TRAP C$ERDF
010702 004353 .WORD 2
010704 005626 .WORD NSSRM
010706 005626 .WORD STAERM
3299 010710 004737 017402 JSR PC,DROPU ;DROP THE UNIT.
3300 010714 000566 BR EXCRTN ;RETURN.
3301 010716 ENDIF
3302 010716 UNTIL #TS.SSR SETIN @TSSR(R5) ;WAIT UNTIL DEVICE IS READY.
010716 032775 000200 002466 BIT #TS.SSR,@TSSR(R5)
010724 001756 BEQ 50071$
3303 010726 IF CMDWRD EQ #SCH THEN ;IF WE ARE DOING A SET CHAR CMD THEN:
010726 023727 003426 140004 CMP CMDWRD,#SCH
010734 001022 BNE 50073$
3304 010736 LET R5SAVE := R5 ;SAVE CURRENT DEVICE POINTER.
010736 010537 003460 MOV R5,R5SAVE
3305 010742 004737 017300 JSR PC,FIRSTU ;FIND FIRST UNIT.
3306 010746 WHILE DEVTBL(R5) NE #END DO
010746 026527 002536 177777 CMP DEVTBL(R5),#END
010754 001405 BEQ 50075$
3307 010756 004737 011672 JSR PC,WSSR ;WAIT FOR UNIT READY OR TIME OUT.
3308 010762 004737 017346 JSR PC,NEXTU ;FIND NEXT UNIT.
3309 010766 000767 BR 50074$
010770 50075$:
3310 010770 LET R5 := R5SAVE ;RESTORE CURRENT DEVICE POINTER.
010770 013705 003460 MOV R5SAVE,R5
3311 010774 LET SCHBK := MSGPKA(R5) ;SET UP ADR OF MSG PKT IN SCH BLOCK.
010774 016537 002506 002446 MOV MSGPKA(R5),SCHBK
3312 011002 ENDIF
011002 50073$:
3313 011002 LET R3 := MSGPKA(R5) ;ADR OF THIS UNIT'S MSG PACKET.
011002 016503 002506 MOV MSGPKA(R5),R3
3314 011006 LET R2 := #0 ;CLR COUNTER.
011006 005002 CLR R2
3315 011010 WHILE R2 NE #MSGCNT DO ;WHILE THERE ARE MORE LOCATIONS:
011010 020227 000016 50076$:
011010 CMP R2,#MSGCNT

```

3316	011014	001405				BEQ	50077\$
	011016			LET (R3) := # -1		; INIT THE MSG PACKET WITH ALL 1'S	
3317	011016	012723	177777			MOV	# -1, (R3)
	011022			LET R2 := R2 + #2		; UPDATE COUNTER.	
3318	011022	062702	000002			ADD	#2, R2
	011026			ENDDO		BR	50076\$
	011026	000770				50077\$:	
3319	011030	105737	002212	TSTB DINT		; ARE INTERRUPTS DISABLED.	
3320	011034	001023		BNE 1\$; BR IF YES.	
3321	011036			IFB INTFLG(R5) GT #1 THEN		; IF MORE THAN ONE INTERRUPT HAS OCCURED:	
	011036	126527	003476			CMPB	INTFLG(R5), #1
	011044	003412				BLE	50100\$
3322	011046			LET TSSREG := @TSSR(R5)		; FREEZE THE CURRENT STATUS REG FOR PRINT	
	011046	017537	002466			MOV	@TSSR(R5), TSSREG
3323	011054			ERRDF 15, TOOMM, STAERM		; REPORT TOO MANY INTERRUPTS.	
	011054	104455				TRAP	C#ERRDF
	011056	000017				.WORD	15
	011060	004542				.WORD	TOOMM
	011062	005626				.WORD	STAERM
3324	011064	004737	017402	JSR PC, DROPU		; DROP THE UNIT	
3325	011070	000500		BR EXCRTN		; RETURN - UNIT HAS BEEN DROPPED.	
3326	011072			ENDIF			
	011072					50100\$:	
3327	011072			LET INTFLG(R5) := #0		; CLR INTERRUPT FLAG FOR THIS DEV.	
	011072	005065	003476			CLR	INTFLG(R5)
3328	011076	052737	000200	002314		; SET INT ENABLE BIT.	
3329	011104			BIS #IE.C, CMDPKT		; IF NOT RETRYING	
	011104	105737	003475	IFB ERRREC EQ #0 THEN			
	011110	001005				TSTB	ERRREC
3330	011112			LET RECCNT(R5) := RECCNT(R5) + #1		BNE	50101\$
	011112	005265	003330			INC	RECCNT(R5)
3331	011116			LET @DATAWT := RECCNT(R5)		; THEN UPDATE REC COUNT TO WRITE IT ON TAPE	
	011116	016577	003330			MOV	RECCNT(R5), @DATAWT
3332	011124			ENDIF			
	011124					50101\$:	
3333	011124			IF L\$TEST EQ #3 AND CMDWRD EQ #WRT THEN			
	011124	023727	002114			CMP	L\$TEST, #3
	011132	001040				BNE	50102\$
	011134	023727	003426			CMP	CMDWRD, #WRT
	011142	001034				BNE	50102\$
3334	011144			LET LOOPCT := LOOPCT + 1			
	011144	005046				CLR	-(SP)
	011146	113716	003525			MOVB	LOOPCT, (SP)
	011152	063716	000001			ADD	1, (SP)
	011156	111637	003525			MOVB	(SP), LOOPCT
	011162	062706	000002			ADD	#2, SP
3335	011166			IFB LOOPCT GT 3 THEN			
	011166	123737	003525			CMPB	LOOPCT, 3
	011174	003417				BLE	50103\$
3336	011176			DELAY 21.			
	011176	012727	000025			MOV	#21, (PC)
	011202	000000				.WORD	0
	011204	013727	002116			MOV	L\$DLY, (PC)
	011210	000000				.WORD	0
	011212	005367	177772			DEC	-6(PC)
	011216	001375				BNE	-4


```

011220 005367 177756
011224 001367
3337 011226 LET LOOPCT :B= 0
011226 113737 000000 003525 MOV 0,LOOPCT
3338 011234 ENDIF
011234 50103$:
3339 011234 ENDIF
011234 50102$:
3340 011234 012775 002314 002456 MOV @CNDPKT,@TSDB(R5) ;LOAD TSDB WITH CNDPKT ADDRESS
3341 ;THIS INITIATES COMMAND EXECUTION.
3342 011242 IF @TS.SSR SETIN @TSSR(R5) THEN ;IF READY DID NOT DROP THEN:
011242 032775 000200 002466 BIT @TS.SSR,@TSSR(R5)
011250 001410 BEQ 50104$
3343 011252 004737 011726 JSR PC,MOVMSG ;MOVE CURRENT MESSAGE PACKET TO COMMON.
3344 011256 ERRDF 3,TOERM,STAERM ;REPORT NO TK25 RESPONSE.
011256 104455 TRAP C$ERDF
011260 000003 .WORD 3
011262 004271 .WORD TOERM
011264 005626 .WORD STAERM
3345 011266 004737 017402 JSR PC,DROPU ;DROP THE UNIT
3346 011272 ENDIF
011272 50104$:
3347 011272 000207 EXCRTN: RTS PC ;RETURN.

```

```

3349      ; THIS SUBROUTINE WAITS FOR THE TK25 INERRUPT OR DONE BIT TO SET AND ALLOWS THE
3350      ; OPERATOR TO TRANSFER CONROL TO THE SUPERVISOR.
3351      ; UPON APPEARANCE OF THE INTERRUPT OR DONE, CHECK TSSR FOR STATUS ERRORS,
3352      ; LOG BYTES AND ERRORS AND PERFORM ERROR RECOVERY IF NESSASARY.
3353      ; INPUTS:
3354      ; OUTPUTS:
3355      ; REGISTERS:      R2, R3.
3356      ; CALLS:          DROPU, MOVMSG, RECUD, CHKERR, LOG, CLRERR.
3357
3358      GOWAIT::IF DEVTBL(R5) EQ #NINUSE THEN
3359      011274 026527 002536 177774      CMP      DEVTBL(R5),#NINUSE
3360      011274 001001                    BNE      50105$
3361      011304 000563                    BR       1$
3362      011306                    ENDIF
3363      011306                    LET TIME1 := #-1      ;INIT TIME OUT COUNTER.
3364      012737 177777 003444      REPEAT      ;REPEAT UNTIL INTERRUPT OCCURES:
3365      011314                    BREAK      ;HONOR SUPERVISOR BREAKS
3366      011314 104422                    IF PCMDWD EQ #RWD AND CMDWRD EQ #WRT THEN
3367      011316 023727 003432 102010      CMP      PCMDWD,#RWD
3368      011324 001020                    BNE      50107$
3369      011326 023727 003426 104005      CMP      CMDWRD,#WRT
3370      011334 001014                    BNE      50107$
3371      011336                    ;POSSIBLE FIRST WRITE ON TAPE
3372      012727 000050      DELAY 40.
3373      011342 000000      MOV      #40.,(PC)+
3374      011344 013727 002116      .WORD 0
3375      011350 000000      MOV      L$DLY,(PC)+
3376      011352 005367 177772      .WORD 0
3377      011356 001375      DEC      -6(PC)
3378      011360 005367 177756      BNE      -4
3379      011364 001367      DEC      -22(PC)
3380      011366                    BNE      -20
3381      011366      ENDIF      ;SO DELAY TO CALIBRATE TAPE
3382      011366      IF CMDWRD EQ #RWD THEN      ;IF COMMAND WAS REWIND THEN:
3383      011366 023727 003426 102010      50107$:
3384      011374 001014      CMP      CMDWRD,#RWD
3385      011376      DELAY 10.      ;WAIT EXTRA 10 MSECS EACH LOOP.
3386      011376 012727 000012      MOV      #10.,(PC)+
3387      011402 000000      .WORD 0
3388      011404 013727 002116      MOV      L$DLY,(PC)+
3389      011410 000000      .WORD 0
3390      011412 005367 177772      DEC      -6(PC)
3391      011416 001375      BNE      -4
3392      011420 005367 177756      DEC      -22(PC)
3393      011424 001367      BNE      -20
3394      011426      ENDIF
3395      011426      50110$:
3396      011426      IF CMDWRD EQ #SFF OR CMDWRD EQ #SFR THEN
3397      011426 023727 003426 105010      CMP      CMDWRD,#SFF
3398      011434 001404      BEQ      50111$
3399      011436 023727 003426 105410      CMP      CMDWRD,#SFR
3400      011444 001014      BNE      50112$

```

```

011446
3372 011446          DELAY 12.
011446 012727 000014
011452 000000
011454 013727 002116
011460 000000
011462 005367 177772
011466 001375
011470 005367 177756
011474 001367
3373 011476          ENDIF
011476
3374 011476          IFB DINT EQ #0 THEN
011476 105737 002212
011502 001003
3375 011504          LET R2 := INTFLG(R5)
011504 016502 003476
3376 011510          ELSE
011510 000406
011512
3377 011512          LET R3 := COMP #TS.SSR
011512 012703 000200
011516 005103
3378 011520          LET R2 := @TSSR(R5) CLR.BY R3 ;FETCH DONE BIT.
011520 017502 002466
011524 040302
3379 011526          ENDIF
011526
3380 011526          LET TIME1 := TIME1 - #1 ;UPDATE TIMEOUT COUNTER.
011526 005337 003444
3381 011532          UNTIL R2 NE #0 OR TIME1 EQ #0 ;REPEAT UNTIL INTERRUPT OR READY OCCURES.
011532 005702
011534 001003
011536 005737 003444
011542 001264
011544
3382 011544          IF TIME1 EQ #0 THEN ;IF TIME OUT HAS OCCURRED:
011544 005737 003444
011550 001022
3383 011552          LET @DATAWT := RECCNT(R5) - #1 ;RE-ADJUST REC COUNT DOWN
011552 016577 003330 171634
011560 005377 171630
3384 011564          JSR PC,MOVMSG ;MOVE CURRENT MSG PACKET TO COMMON AREA.
3385 011570          ERRDF 4,NOINTM,STAERM ;REPORT NO INTERRUPT.
011570 104455
011572 000004
011574 004503
011576 005626
3386 011600          JSR PC,DROPU ;DROP THE UNIT.
3387 011604          LET R3 := #ENDERF
011604 012703 003476
3388 011610          JSR PC,CLRERR ;CLEAR ALL ERROR FLAGS
3389 011614          ELSE
011614 000417
011616
3390 011616          JSR PC,MOVMSG ;MOVE CURRENT MSG. PACKET TO COMMON AREA.
3391 011622          JSR PC,RECUD ;UPDATE THE RECORD COUNT.

```

50111\$:

```

;ADD DELAY FOR SPACE TAPE MARK COMMANDS
MOV #12.,(PC)
.WORD 0
MOV L$DLY,(PC)
.WORD 0
DEC -6(PC)
BNE -.4
DEC -22(PC)
BNE .-20

```

50112\$:

```

;IF INTERRUPTS ARE ENABLED.
TST DINT
BNE 50113$
;FETCH INTERRUPT OCCURRED FLAG.
MOV INTFLG(R5),R2
;IF IN BRUTUS MODE:
BR 50114$

```

50113\$:

```

;SET UP A MASK FOR THE DONE BIT.
MOV #TS.SSR,R3
COM R3
;FETCH DONE BIT.
MOV @TSSR(R5),R2
BIC R3,R2

```

50114\$:

```

;UPDATE TIMEOUT COUNTER.
DEC TIME1
;REPEAT UNTIL INTERRUPT OR READY OCCURES.
TST R2
BNE 50115$
TST TIME1
BNE 50106$

```

50115\$:

```

;IF TIME OUT HAS OCCURRED:
TST TIME1
BNE 50116$
;RE-ADJUST REC COUNT DOWN
MOV RECCNT(R5),@DATAWT
DEC @DATAWT

```

TRAP

```

C$ERDF
.WORD 4
.WORD NOINTM
.WORD STAERM

```

MOV

#ENDERF,R3

BR

50117\$

50116\$:

```

;MOVE CURRENT MSG. PACKET TO COMMON AREA.
;UPDATE THE RECORD COUNT.

```

```

3392 011626 004737 012160      JSR  PC,CHKERR      ;CHECK FOR STATUS ERRORS.
3393 011632      IFB  WRTYFG EQ #0 THEN ;
      011632 105737 003467      ;
      011636 001006      TSTB  WRTYFG
3394 011640 004737 015436      JSR  PC,LOG        ;LOG BYTES AND ERRORS.
3395 011644      LET  R3 := #ENDERF      BNE  50120#
      011644 012703 003476      ;
3396 011650 004737 011656      JSR  PC,CLRERR      MOV  #ENDERF,R3
3397 011654      ENDIF      ;CLEAR ALL ERROR FLAGS
      011654      ;
3398 011654      ENDIF      50120#:
      011654      ;
3399 011654 000207      1#: RTS  PC      50117#:
      ;RETURN IF DONE.

```

```

3401 ; SUBROUTINE TO CLEAR FLAGS.
3402 ; INPUTS: R3 = LWA TO BE CLEARED + 2.
3403 ; OUTPUTS:
3404 ; REGISTERS: R2
3405 ; CALLS:
3406
3407 011656 CLRERR:: LET R2 := #BGNFLG
011656 012702 003464 MOV #BGNFLG,R2
3408 011662 REPEAT
011662 50121$:
3409 011662 LET (R2)+ := #0
011662 005022 CLR (R2)+
3410 011664 UNTIL R2 EQ R3
011664 020203 CMP R2,R3
011666 001375 BNE 50121$
3411 011670 000207 RTS PC
3412
3413
3414
3415 ; SUBROUTINE TO WAIT UNTIL CURRENT UNIT IS READY OR UNTIL TIME OUT.
3416 ; INPUTS:
3417 ; OUTPUTS:
3418 ; REGISTERS:
3419 ; CALLS:
3420
3421 011672 WSSR:: LET TIME1 := #-1 ;INIT TIMEOUT COUNTER.
011672 012737 177777 003444 MOV #-1,TIME1
3422 011700 REPEAT ;REPEAT UNTIL DEV READY OR TIMEOUT:
011700 50122$:
3423 011700 BREAK ;BREAK TO THE SUPERVISOR.
011700 104422 TRAP C$BRK
3424 011702 LET TIME1 := TIME1 - #1 ;UPDATE TIMEOUT COUNTER.
011702 005337 003444 DEC TIME1
3425 011706 UNTIL #TS.SSR SETIN @TSSR(R5) OR TIME1 EQ #0
011706 032775 000200 002466 BIT #TS.SSR,@TSSR(R5)
011714 001003 BNE 50123$
011716 005737 003444 TST TIME1
011722 001366 BNE 50122$
011724 50123$:
3426 ;REPEAT UNTIL DEV READY OR TIMEOUT.
3427 011724 000207 RTS PC ;RETURN.
3428
3429

```

```

3431
3432      ;      SUBROUTINE TO MOVE THE CURRENT MESSAGE PACKET TO THE COMMON AREA AND
3433      ;      TO UPDATE THE CURRENT TERMINATION CLASS CODE.
3434      ;      INPUTS:
3435      ;      OUTPUTS:
3436      ;      REGISTERS:      R2, R3.
3437      ;      CALLS:
3438
3439 011726      MOVMSG:: LET TSSREG := @TSSR(R5)      ;FREEZE THE STATUS REG CONTENTS
011726 017537 002466 003462      MOV      @TSSR(R5),TSSREG
3440 011734      LET R2 := TSSREG CLR.BY #TSC.TCC ;EXTRACT THE TERMINATION CLASS CODE.
011734 013702 003462      MOV      TSSREG,R2
011740 042702 177761      BIC      #TSC.TCC,R2
3441 011744      LET CTCC := R2 SHIFT -1      ;AND SAVE IT
011744 010237 003456      MOV      R2,CTCC
011750 006237 003456      ASR      CTCC
3442 011754      LET R3 := MSGPKA(R5)      ;ADR OF THIS DEVICE'S MSG.
011754 016503 002506      MOV      MSGPKA(R5),R3
3443 011760      LET R2 := #0      ;CLR COUNTER.
011760 005002      CLR      R2
3444 011762      WHILE R2 NE #MSGCNT DO      ;WHILE THERE ARE MORE LOCATIONS:
011762      50124$:
011762 020227 000016      CMP      R2,#MSGCNT
011766 001405      BEQ      50125$
3445 011770      LET MSGPKT(R2) := (R3).      ;MOVE MSG TO COMMON AREA.
011770 012362 002340      MOV      (R3).,MSGPKT(R2)
3446 011774      LET R2 := R2 + #2      ;UPDATE COUNTER.
011774 062702 000002      ADD      #2,R2
3447 012000      ENDDO
012000 000770      BR      50124$
012002      50125$:
3448 012002      LET EOTFLG(R5) := MSGPKT*MS.XSO ;MOVE XSTATO TO EOT FLAG.
012002 013765 002346 003506      MOV      MSGPKT*MS.XSO,EOTFLG(R5)
3449 012010      RTS      PC
000207

```

```

3451      ;      SUBROUTINE TO ADJUST THE RECORD COUNT.
3452      ;      INPUTS:
3453      ;      OUTPUTS:
3454      ;      REGISTERS:
3455      ;      CALLS:
3456
3457      RECUD:: IFB RECLOG EQ #0 THEN          ;IF RECORD HAS NOT BEEN LOGGED:
          012012      105737  003471          TSTB  RECLOG
          012016      001057          BNE   50126$
3458      012020          LET RECcnt(R5) := RECcnt(R5) - #1
          012020      005365  003330          DEC   RECcnt(R5)
3459      012024          IF #BITO NOTSETIN CTCC AND #X2.OPM SETIN MSGPKT+MS.XS2 THEN ;IF TAPE MOVED THEN:
          012024      032737  000001  003456          BIT   #BITO,CTCC
          012032      001046          BNE   50127$
          012034      032737  100000  002352          BIT   #X2.OPM,MSGPKT+MS.XS2
          012042      001442          BEQ   50127$
3460      012044          LET RECLOG :B= RECLOG + #1 ;SET RECORD LOGGED,
          012044      105237  003471          INCB  RECLOG
3461      012050          IF CMDWRD EQ #RWD THEN          ;IF THIS IS A REWIND CMD:
          012050      023727  003426  102010          CMP   CMDWRD,#RWD
          012056      001003          BNE   50130$
3462      012060          LET RECcnt(R5) := #0          ;CLEAR RECORD COUNT,
          012060      005065  003330          CLR   RECcnt(R5)
3463      012064          ELSE
          012064      000431          BR    50131$
          012066          50130$:
3464      012066          IF #BRF.C SETIN CMDWRD THEN          ;IF BRF USED, UPDATE RECORD COUNT.
          012066      032737  004000  003426          BIT   #BRF.C,CMDWRD
          012074      001425          BEQ   50132$
3465      012076          IF #MOD.CO NOTSETIN CMDWRD THEN          ;IF A FORWARD CMD:
          012076      032737  000400  003426          BIT   #MOD.CO,CMDWRD
          012104      001007          BNE   50133$
3466      012106          IF #MOD.CO NOTSETIN PCMDWD THEN          ;IF PREV CMD WAS A FWD ALSO:
          012106      032737  000400  003432          BIT   #MOD.CO,PCMDWD
          012114      001002          BNE   50134$
3467      012116          LET RECcnt(R5) := RECcnt(R5) + #1          ;INCREMENT RECORD COUNT.
          012116      005265  003330          INC   RECcnt(R5)
3468      012122          ENDIF
          012122          50134$:
3469      012122          ELSE          ;IF REVERSE CMD:
          012122      000412          BR    50135$
          012124          50133$:
3470      012124          IF #MOD.CO SETIN PCMDWD THEN          ;IF PREVIOUS CMD WAS A REV ALSO:
          012124      032737  000400  003432          BIT   #MOD.CO,PCMDWD
          012132      001406          BEQ   50136$
3471      012134          IF #X0.BOT NOTSETIN EOTFLG(R5) THEN          ;WHEN NOT AT BOT THEN
          012134      032765  000002  003506          BIT   #X0.BOT,EOTFLG(R5)
          012142      001002          BNE   50137$
3472      012144          LET RECcnt(R5) := RECcnt(R5) - #1          ;DECREMENT RECORD COUNT.
          012144      005365  003330          DEC   RECcnt(R5)
3473      012150          ENDIF
          012150          50137$:
3474      012150          ENDIF
          012150          50136$:
3475      012150          ENDIF
          012150          50135$:
3476      012150          ENDIF

```

```
012150
3477 012150          ENDIF
012150
3478 012150          ENDIF
012150
3479 012150          LET @DATAWT := RECCNT(R5)
012150 016577 003330 171236
3480 012156          ENDIF
012156
3481 012156 000207   RTS      PC          ;RETURN.

50132$:
50131$:
50127$:
MOV     RECCNT(R5),@DATAWT
50126$:
```



```

3483      :      THIS IS THE ERROR CHECK SUBROUTINE.  AFTER INTERRUPT THIS
3484      :      SUBROUTINE IS CALLED TO CHECK THE TK25 STATUS.
3485      :      IF SPECIAL COND IS SET THEN THE TCC HANDLING SUBROUTINE IS ENTERED.
3486      :      IF THE RFC IS NON ZERO FOR A COMMAND REQUIRING A BPCR,
3487      :      THEN AN ERROR RFC IS REPORTED,
3488      :      INPUTS:
3489      :      OUTPUTS:
3490      :      REGISTERS:      R2, R4.
3491      :      CALLS:          TCC0-TCC7.
3492
3493 012160      :      CHKERR::IF DEVTBL(R5) NE @NINUSE THEN
012160      026527      002536      177774      CMP      DEVTBL(R5),@NINUSE
012166      001556      BEQ      50140$
3494 012170      :      IF @X1.EWN SETIN MSGPKT*MS.XS1 THEN      ;IF EARLY WARNING IS SET AND
012170      032737      000010      002350      BIT      @X1.EWN,MSGPKT*MS.XS1
012176      001447      BEQ      50141$
3495 012200      :      IFB EWRPNT NE @0 THEN      ;IF END OF TRACK STATUS PRINTS ALLOWED
012200      105737      002216      TSTB    EWRPNT
012204      001421      BEQ      50142$
3496 012206      :      LET R2 := MSGPKT*MS.XS1 CLR.BY @177417
012206      013702      002350      MOV      MSGPKT*MS.XS1,R2
012212      042702      177417      BIC      @177417,R2
3497 012216      :      LET R2 := R2 SHIFT -4
012216      006202      ASR      R2
012220      006202      ASR      R2
012222      006202      ASR      R2
012224      006202      ASR      R2
3498 012226      :      PRINTF @EWMMSG,R2      ;PRINT END OF TRACK MESSAGE
012226      010246      MOV      R2,-(SP)
012230      012746      012526      MOV      @EWMMSG,-(SP)
012234      012746      000002      MOV      @2,-(SP)
012240      010600      MOV      SP,R0
012242      104417      TRAP    C$PNTF
012244      062706      000006      ADD      @6,SP
3499 012250      :      ENDIF
3500 012250      :      IF CTCC EQ @4 THEN      ;IF TCC4 AND WE DID A WRITE
012250      023727      003456      000004      CMP      CTCC,@4
012256      001017      BNE     50143$
3501 012260      :      IF CMDWRD EQ @WRT OR CMDWRD EQ @WTM THEN
012260      023727      003426      104005      CMP      CMDWRD,@WRT
012266      001404      BEQ     50144$
012270      023727      003426      100011      CMP      CMDWRD,@WTM
012276      001007      BNE     50145$
012300      :      50144$:
3502 012300      :      IFB IREC EQ @0 THEN
012300      105737      002210      TSTB    IREC
012304      001004      BNE     50146$
3503 012306      :      JSR      PC,EWRTRY      ;DO EW RETRY
3504 012312      000137      012524      JMP      1$      ;ALAS, A "GOTO" STATEMENT, EH?
3505 012316      :      ENDIF
012316      :      50146$:
3506 012316      :      ENDIF
012316      :      50145$:
3507 012316      :      ENDIF
012316      :      50143$:
3508 012316      :      ENDIF

```

```

012316
3509 012316 032737 100000 003462 IF #TS.SC SETIN TSSREG THEN ;IF SPECIAL COND STATUS IS SET THEN:
012316 032737 100000 003462 ;IF TCC IS NOT 2 THEN:
012324 001441 ;IF TCC IS NOT 2 THEN:
3510 012326 023727 003456 000002 IF CTCC NE #2 THEN ;IF TCC IS NOT 2 THEN:
012326 023727 003456 000002 ;IF NOT IN ERROR RECOVERY:
012334 001405 ;IF NOT IN ERROR RECOVERY:
3511 012336 105737 003475 IFB ERRREC EQ #0 THEN ;IF NOT IN ERROR RECOVERY:
012336 105737 003475 ;IF NOT IN ERROR RECOVERY:
012342 001002 ;IF NOT IN ERROR RECOVERY:
3512 012344 005265 003270 INC SCCNT(R5) ;INC SC COUNTER.
3513 012350 ENDIF
012350
3514 012350 ENDIF
012350
3515 012350 IF #TS.NXM SETIN TSSREG OR #TS.UPE SETIN TSSREG THEN ;WHEN NON-EXISTANT MEMO
012350 032737 004000 003462 ;WHEN NON-EXISTANT MEMO
012356 001004 ;WHEN NON-EXISTANT MEMO
012360 032737 040000 003462 ;WHEN NON-EXISTANT MEMO
012366 001412 ;WHEN NON-EXISTANT MEMO
012370
3516 012370 IF #X2.OPM NOTSETIN MSGPKT+MS.XS2 THEN ;AND TAPE NOT MOVED
012370 032737 100000 002352 ;AND TAPE NOT MOVED
012376 001003 ;AND TAPE NOT MOVED
3517 012400 LET R2 := #5 ;SET TCC5 INDEX
012400 012702 000005 ;SET TCC5 INDEX
3518 012404 ELSE
012404 000402 ;SET TCC5 INDEX
012406
3519 012406 LET R2 := #4 ;TAPE MOVED, SET TCC4 INDEX
012406 012702 000004 ;TAPE MOVED, SET TCC4 INDEX
3520 012412 ENDIF
012412
3521 012412 ELSE
012412 000402 ;TAPE MOVED, SET TCC4 INDEX
012414
3522 012414 LET R2 := CTCC ;SET DETECTED TCC INDEX
012414 013702 003456 ;SET DETECTED TCC INDEX
3523 012420 ENDIF
012420
3524 012420 LET R2 := R2 SHIFT 1 ;CURRENT TCC X 2.
012420 006302 ;CURRENT TCC X 2.
3525 012422 004772 012612 JSR PC,@TCCRA(R2) ;GO TO THE TCC HANDLING SUBROUTINE.
3526 012426 ELSE
012426 000430 ;GO TO THE TCC HANDLING SUBROUTINE.
012430
3527 012430 IF #BRF.C SETIN CMDWRD THEN ;IF BRF IS USED IN THIS CMD THEN:
012430 032737 004000 003426 ;IF BRF IS USED IN THIS CMD THEN:
012436 001424 ;IF BRF IS USED IN THIS CMD THEN:
3528 012440 IF MSGPKT+MS.RFC NE #0 THEN ;IF THERE IS AN RFC THEN:
012440 005737 002344 ;IF THERE IS AN RFC THEN:
012444 001421 ;IF THERE IS AN RFC THEN:
3529 012446 IFB RANDOM EQ #0 ORB VFYFLG NE #0 THEN
012446 105737 003521 ;IF THERE IS AN RFC THEN:
012452 001403 ;IF THERE IS AN RFC THEN:
012454 105737 003522 ;IF THERE IS AN RFC THEN:
012460 001413 ;IF THERE IS AN RFC THEN:

```

```

3530 012462
3531 012462 105737 003526 IFB IRE EQ #0 THEN ;IF NOT IN RANDOM OR IF CMD IS WTV:
012466 001010 ;IF RFC ERROR REPORTS ARE ALLOWED:
3532 012470 005265 003310 LET HRDCNT(R5) := HRDCNT(R5) + #1 ;UPDATE HARD ERROR COUNT
012470 005265 003310 ERRHRD 13,RFCERM,STAERM ;REPORT RFC ERROR
3533 012474 104456 TRAP C$ERHRD
012476 000015 .WORD 13
012500 004336 .WORD RFCERM
012502 005626 .WORD STAERM
3534 012504 004737 014050 JSR PC,RAMDUM ;GO DO RAM DUMP
3535 012510 ENDIF
3536 012510 ENDIF
3537 012510 ENDIF
3538 012510 ENDIF
3539 012510 ENDIF
3540 012510 105737 003473 IFB RWERR NE #0 THEN ;IF A READ/WRITE ERROR HAS OCCURRED THEN:
012514 001403 TSTB RWERR
3541 012516 013737 003430 002314 LET CMDPKT := CMDSAV ;RESTORE CMD PACKET AFTER ERROR RECOV.
012516 013737 003430 002314 BEQ 50165$
3542 012524 ENDIF MOV CMDSAV,CMDPKT
3543 012524 ENDIF
3544 012524 000207 1$: RTS PC ;RETURN.
3545
3546 012526 045 116 045 EWMSG: .ASCIZ /%N%AEARLY WARNING DURING WRITE AT END OF TRACK #D2/
012531 101 105 101
012534 122 114 131
012537 040 127 101
012542 122 116 111
012545 116 107 040
012550 104 125 122
012553 111 116 107
012556 040 127 122
012561 111 124 105
012564 040 101 124
012567 040 105 116
012572 104 040 117
012575 106 040 124
012600 122 101 103
012603 113 040 045
012606 104 062 000
3547 .EVEN

```

3549 ; ADDRESSES OF TCC HANDLING ROUTINES FOR TERMINATION CLASS CODES 0 - 7.
3550
3551 012612 012740 TCCRA: TCC0
3552 012614 012762 TCC1
3553 012616 013000 TCC2
3554 012620 013146 TCC3
3555 012622 013170 TCC4
3556 012624 013670 TCC5
3557 012626 013772 TCC6
3558 012630 014026 TCC7

```

3560      ;      SUBROUTINE TO HANDLE EARLY WARNING WRITE ERRORS.
3561      ;
3562      ;      THIS ROUTINE WILL SIMPLY TAKE CARE OF EARLY WARNING WRITE ERRORS
3563      ;      BY REISSUING THE COMMAND WHICH FAILED WITH THE RETRY MODIFIER SET.
3564      ;      NOTE THAT NO ERROR FLAGS, COUNTS, ETC ARE CHANGED.
3565      ;      (EW ERRORS WILL NOT SHOW UP IN ERROR TALLIES).
3566
3567 012632 EWRTRY::
3568 012632      LET -(SP) := CNDPKT          ;SAVE THE COMMAND PACKET
3569 012632      013746 002314      MOV      CNDPKT, -(SP)
3570 012636      052737 001000 002314      LET CNDPKT := CNDPKT SET.BY #MOD.C1 ;SET THE RETRY MODIFIER IN COMMAND
3571 012644      112737 000001 003475      BIS      #MOD.C1,CNDPKT
3572 012652      004737 010654      LET ERRREC :B= #1          ;SHOW ERROR RECOVERY IN PROCESS
3573 012656      105037 003475      MOV      #1,ERRREC
3574 012662      012637 002314      JSR      PC,EXCUTE      ;DO THE COMMAND
3575 012666      012602      LET ERRREC :B= #0      ;NO MORE ERROR RECOVERY
3576 012670      012746 000012      CLRB     ERRREC
3577 012674      012727 000372      LET CNDPKT := (SP).    ;RESTORE THE ORIGINAL COMMAND STATUS
3578 012700      000000      LET R2 := (SP).    ;MODIFY THE RETURN ADDRESS
3579 012702      013727 002116      MOV      (SP).,R2
3580 012706      000000      LET -(SP) := #10.    ;OVERALL LOOP CONTROL
3581 012710      005367 177772      REPEAT      ;LOOP
3582 012714      001375      DELAY 250.    ;WAIT FOR THE UNIT TO STOP
3583 012716      005367 177756      MOV      #250.,(PC).
3584 012722      001367      .WORD 0
3585 012724      005316      MOV      L#DLY,(PC).
3586 012726      005716      .WORD 0
3587 012730      001361      DEC      -6(PC)
3588 012732      062706 000002      BNE      .-4
3589 012736      000207      DEC      -22(PC)
3590      BNE      .-20
3591      LET (SP) := (SP) - #1      ;ONE LESS ITERATION
3592      UNTIL (SP) EQ #0          ;UNTIL = 0
3593      DEC      (SP)
3594      TST      (SP)
3595      BNE      50166$
3596      LET SP := SP + #2        ;CORRECT THE STACK
3597      ADD      #2,SP
3598      RTS      PC              ;RETURN TO CALLER

```

```

3583      ;      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 0, UNDEFINED SPECIAL
3584      ;      CONDITION ERROR.
3585      ;      INPUTS:
3586      ;      OUTPUTS:
3587      ;      REGISTERS:
3588      ;      CALLS:
3589
3590 012740 TCC0:: LET HRDCNT(R5) := HRDCNT(R5) + #1 ;UPDATE HARD ERROR COUNT.
012740 005265 003310      INC      HRDCNT(R5)
3591 012744      ERRHRD 5,SCERM,STAERM      ;REPORT SPECIAL CONDITION ERROR.
012744 104456      TRAP      C#ERHRD
012746 000005      .WORD      5
012750 004312      .WORD      SCERM
012752 005626      .WORD      STAERM
3592 012754 004737 014050      JSR      PC,RAMDUM      ;GO DO RAM DUMP
3593 012760 000207      RTS      PC      ;RETURN.
3594
3595
3596
3597
3598
3599      ;      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 1, ATTENTION CONDITION.
3600      ;      THIS TCC INDICATES THAT THE DRIVE HAS UNDERGONE A STATUS CHANGE
3601      ;      SUCH AS GOING OFFLINE OR COMING ONLINE.
3602      ;      INPUTS:
3603      ;      OUTPUTS:
3604      ;      REGISTERS:      R2,R4
3605      ;      CALLS:      DROPU
3606
3607 012762 TCC1:: ERRDF 6,ATTNM,STAERM      ;REPORT ATTENTION-UNIT OFF LINE.
012762 104455      TRAP      C#ERDF
012764 000006      .WORD      6
012766 004417      .WORD      ATTNM
012770 005626      .WORD      STAERM
3608 012772 004737 014050      JSR      PC,RAMDUM      ;GO DO RAM DUMP
3609 012776 000207      RTS      PC      ;RETURN.

```

```

3611      ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 2, TAPE STATUS ALERT.
3612      ; A STATUS CONDITION HAS BEEN ENCOUNTERED THAT MAY HAVE SIGNIFICANCE
3613      ; TO THE PROGRAM. BITS OF INTEREST INCLUDE TMK, RLS, LET, RLL, BOT, EOT.
3614      ; INPUTS:
3615      ; OUTPUTS:
3616      ; REGISTERS:
3617      ; CALLS:
3618
3619 013000 TCC2:: IF @X0.BOT SETIN MSGPKT*MS.XSO ANDB EXPBOT NE #0 THEN
013000      BIT @X0.BOT,MSGPKT*MS.XSO
013006      BEQ 50167#
013010      TSTB EXPBOT
013014      BEQ 50167#
3620      ;IF AT BOT AND BOT IS EXPECTED:
3621 013016 000452 BR TC2RTN ;RETURN-TCC2 CAUSED BY EXPECTED BOT.
3622 013020 ENDF
013020
3623 013020 IF L#TEST EQ #3 THEN 50167#:
013020      CMP L#TEST,#3
013026      BNE 50170#
3624 013030 IF PCMDWD EQ #WTM AND CMDWRD EQ #SRR THEN
013030      CMP PCMDWD,#WTM
013036      BNE 50171#
013040      CMP CMDWRD,#SRR
013046      BNE 50171#
3625 013050 000435 BR TC2RTN
3626 013052 ENDF
013052
3627 013052 ENDF 50171#:
013052
3628 013052 IF @X0.RLS!X0.RLL!X0.TMK!X0.LET!X0.BOT SETIN MSGPKT*MS.XSO THEN 50170#:
013052      BIT @X0.RLS!X0.RLL!X0.TMK!X0.LET
013060      BEQ 50172#
3629      ;IF TCC2 CAUSED BY ANYTHING BUT EOT:
3630 013062 IFB RANDOM EQ #0 ORB VFYFLG NE #0 THEN
013062      TSTB RANDOM
013066      BEQ 50173#
013070      TSTB VFYFLG
013074      BEQ 50174#
013076      50173#:
3631      ;IF NOT IN RANDOM OR IF CMD IS WTV:
3632 013076 IFB IRE EQ #0 THEN ;IF RFC ERROR REPORTS ARE ALLOWED:
013076      TSTB IRE
013102      BNE 50175#
3633 013104 IFB ERRREC NE #0 THEN ;IF WE ARE IN ERROR RECOVERY THEN:
013104      TSTB ERRREC
013110      BEQ 50176#
3634 013112 LET UNREC :B= UNREC * #1 ;SET UNRECOVERABLE FLAG FOR LOG.
013112      INCB UNREC
3635 013116 ELSE ;ELSE - IF NOT IN ERROR RECOVERY:
013116      BR 50177#
013120      50176#:
3636 013120 LET SCCNT(R5) := SCCNT(R5) * #1 ;INCREMENT THE SPEC COND COUNTER.
013120      INC SCCNT(R5)
3637 013124 ENDF
013124
3638 013124 LET HRDCNT(R5) := HRDCNT(R5) * #1 ;UPDATE HARD ERROR COUNT.

```

```

3639 013124 005265 003310
      013130
      013130 104456
      013132 000007
      013134 004520
      013136 005626
3640 013140 004737 014050
3641 013144
      013144
3642 013144
      013144
3643 013144
      013144
3644 013144 000207
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658 013146
      013146 104455
      013150 000010
      013152 004435
      013154 005626
3659 013156 004737 014050
3660 013162 004737 017402
3661 013166 000207

      ERRHRD 7,TSAM,STAERM
      JSR PC,RAMDUM
      ENDIF
      ENDIF
      ENDIF
TC2RTN: RTS PC
      ; THE SPECIFIED FUNCTION WAS NOT INITIATED. BITS OF INTEREST ARE
      ; RMR, OFL, VCK, BOT, ILC, WLE, ILA, AND NBA.
      ; INPUTS:
      ; OUTPUTS:
      ; REGISTERS: R2,R4
      ; CALLS: DROPU
TCC3:: ERRDF 8,FUNRM,STAERM
      JSR PC,RAMDUM
      JSR PC,DROPU
      RTS PC

      INC HRDCNT(R5)
      ;REPORT TAPE STATUS ALERT.
      TRAP C$ERHRD
      .WORD 7
      .WORD TSAM
      .WORD STAERM
      ;GO DO RAM DUMP
      50175$:
      50174$:
      50172$:
      ;RETURN.

      ;REPORT FUNCTION REJECT.
      TRAP C$ERDF
      .WORD 8
      .WORD FUNRM
      .WORD STAERM
      ;GO DO RAM DUMP
      ;DROP THE UNIT.
      ;RETURN.

```



```

3663      ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 4, RECOVERABLE ERROR.
3664      ; TAPE POSITION IS ONE RECORD BEYOND WHAT ITS POSITION WAS WHEN
3665      ; THE FUNCTION WAS INITIATED. RECOVERY PROCEDURE IS TO LOG THE
3666      ; ERROR AND ISSUE THE APPROPRIATE RETRY COMMAND.
3667      ; 2 WRITE-ERROR-RECOVERY ALGORITHMS CAN BE SELECTED:
3668      ; THE FIRST ONE, VIA BADTSW SWITCH, DOES DETECT BAD SPOTS ON TAPE.
3669      ; IT CALLS A WRITE RETRY SUBR UNTIL THE RECORD IS RECOVERED
3670      ; OR 20 BAD SPOTS HAVE BEEN LOGGED. AFTER LOGGING 256 BAD
3671      ; SPOTS, A BAD TAPE OVERFLOW MSG IS PRINTED AND THE
3672      ; UNIT DROPPED.
3673      ; THE SECOND ALGORITHM ISSUES THE TK25 WRITE RETRY COMMAND
3674      ; UP TO 16 TIMES BEFORE DROPPING THE UNIT OR PROCEEDING
3675      ; WITH THE NEXT RECORD ON RECOVERY.
3676      ; INPUTS:
3677      ; OUTPUTS:
3678      ; REGISTERS:      R2,R4.
3679      ; CALLS:          RTLE, EXECUTE, GOWAIT, DROPU, WRTY
3680
3681 013170      ; TCC4:: IF DEVTBL(R5) EQ #NINUSE THEN
013170 026527 002536 177774      CMP      DEVTBL(R5),#NINUSE
013176 001002      BNE      50200$
3682 013200      JMP      3$
3683 013204      ENDIF
013204
3684 013204      IF CMDLG EQ #2 ANDB BADTSW NE #0 THEN
013204 023727 003434 000002      CMP      CMDLG,#2
013212 001142      BNE      50201$
013214 105737 002211      TSTB     BADTSW
013220 001537      BEQ      50201$
3685 013222      IFB ERRREC EQ #0 ANDB ERCVER NE #0 THEN
013222 105737 003475      TSTB     ERRREC
013226 001011      BNE      50202$
013230 105737 002207      TSTB     ERCVER
013234 001406      BEQ      50202$
3686 013236      ERRSOFT 9,RERM,STAERM ;
013236 104457      TRAP     C$ERSOFT
013240 000011      .WORD    9
013242 004632      .WORD    RERM
013244 005626      .WORD    STAERM
3687 013246 004737 014050      JSR      PC,RAMDUM      ;GO DO RAM DUMP
3688 013252      ENDIF
013252
3689 013252      IFB IREC EQ #0 THEN
013252 105737 002210      TSTB     IREC
013256 001115      BNE      50203$
3690 013260      LET ERRREC :B= ERRREC + #1 ;RETRY FLAG FOR EXECUTE SUBR: DON'T UPDATE REC CNT
013260 105237 003475      INCB     ERRREC
3691 013264      LET WRTYER :B= WRTYER + #1 ;REWRITE ERROR FLAG FOR WRTY SUBR
013264 105237 003470      INCB     WRTYER
3692 013270      IFB WRTYFG EQ #0 THEN ;FIRST RETRY ON THIS RECORD: SUBSEQUENT
013270 105737 003467      TSTB     WRTYFG
013274 001105      BNE      50204$
3693      ;RETRIES WITH TCC4 ERRORS BY-PASS THIS SECTION
3694 013276      LET WTYWRD := CMDWRD ;SAVE WRITE COMMAND PACKET
013276 013737 003426 014730      MOV      CMDWRD,WTYWRD
3695 013304      LET WTYCMD := CMDPKT ;
013304 013737 002314 014726      MOV      CMDPKT,WTYCMD

```

3696	013312				LET WTYBRF := CMDPKT.CP.CNT		
	013312	013737	002322	014732			MOV CMDPKT.CP.CNT,WTYBRF
3697	013320				LET RWERR :B= RWERR + #1	;LOG SUBR FLAG:	COUNT WRT ERRORS
	013320	105237	003473				INCB RWERR
3698	013324				LET WRTYFG :B= WRTYFG + #1	;RETRY IN PROGRESS FLAG	
	013324	105237	003467				INCB WRTYFG
3699	013330				REPEAT		
	013330						50205\$:
3700	013330	005265	003250		LET WRTYCT(R5) := WRTYCT(R5) + #1		;COUNT GLOBAL WRITE RETRIES
	013330						INC WRTYCT(R5)
3701	013334	005037	003464		LET RETRYC := #0	;CLEAR # OF RETRIES PER RECORD	
	013334						CLR RETRYC
3702	013340				LET RPTCNT :B= #0	;CLEAR # OF REPEATS	
	013340	105037	003466				CLRB RPTCNT
3703	013344	004737	014350		JSR PC,WRTY	;CALL WRITE RETRY	
3704	013350				IF DEVTBL(R5) EQ #NINUSE THEN		
	013350	026527	002536	177774			CMP DEVTBL(R5),#NINUSE
	013356	001001					BNE 50206\$
3705	013360	000542			BR 3\$		
3706	013362				ENDIF		
	013362						50206\$:
3707	013362				UNTILB WRTYER EQ #0 OR #BTPT HIS #40.		;REPEAT RETRIES ON SAME RECORD
	013362	105737	003470				TSTB WRTYER
	013366	001404					BEQ 50207\$
	013370	027727	170122	000050			CMP #BTPT,#40.
	013376	103754					BLO 50205\$
	013400						50207\$:
3708							;UNTIL RECOVERED OR 20 BAD SPOTS
3709	013400				IF #BTPT HIS #40. THEN		;WHEN 20 BAD SPOTS LOGGED (TWR)
	013400	027727	170112	000050			CMP #BTPT,#40.
	013406	103406					BLO 50210\$
3710	013410	004737	015146		JSR PC,BORERS	;ERASE BAD RECORD	
3711	013414				LET RECCNT(R5) := RECCNT(R5) - #1		
	013414	005365	003330				DEC RECCNT(R5)
3712	013420	004737	014050		JSR PC,RAMDUM	;GO DO RAM DUMP	
3713	013424				ENDIF		
	013424						50210\$:
3714	013424				IF #BTPT HIS #512. THEN		
	013424	027727	170066	001000			CMP #BTPT,#512.
	013432	103417					BLO 50211\$
3715	013434				PRINTB #BTMSG2	;BAD TAPE OVERFLOW MESSAGE	
	013434	012746	015021				MOV #BTMSG2,-(SP)
	013440	012746	000001				MOV #1,-(SP)
	013444	010600					MOV SP,RO
	013446	104414					TRAP C\$PNTB
	013450	062706	000004				ADD #4,SP
3716	013454	004737	017402		JSR PC,DROPU	;DROP THE UNIT.	
3717	013460				LET RECCNT(R5) := #0		
	013460	005065	003330				CLR RECCNT(R5)
3718	013464				LET #TSDB(R5) := #RWCPK	;REWIND UNIT	
	013464	012775	002334	002456			MOV #RWCPK,#TSDB(R5)
3719	013472				ENDIF		
	013472						50211\$:
3720	013472				LET WRTYFG :B= #0	;RETRY COMPLETE	FLAG
	013472	105037	003467				CLR WRTYFG
3721	013476				LET MISCFG :B= MISCFG + #1	;DO NOT	HALT ON THIS CMD FLG
	013476	105237	003537				INCB MISCFG

```

3722 013502          LET PCMDWD := WTYWRD          ;RESTORE ORIGINAL WRT CMD AFTER RECOVERY
013502 013737 014730 003432          MOV          WTYWRD,PCMDWD
3723 013510          ENDIF
013510          50204$:
3724 013510          ELSE
013510 000402          BR          50212$
013512          50203$:
3725 013512          LET UNREC :B= UNREC * #1      ;
013512 105237 003474          INCB          UNREC
3726 013516          ENDIF
013516          50212$:
3727 013516          ELSE
013516 000463          BR          50213$
013520          50201$:
3728 013520 004737 014212          JSR PC,RTLE          ;CHECK FOR RETRY LIMIT EXCEEDED.
3729 013524          IF CMDLG GT #2 THEN          ;IF READ CMD THEN:
013524 023727 003434 000002          CMP          CMDLG,#2
013532 003411          BLE          50214$
3730 013534          LET R2 := #RRECL SHIFT -1      ;R2=READ RETRY COUNT LIMIT / 2
013534 012702 000020          MOV          #RRECL,R2
013540 006202          ASR          R2
3731 013542          IF RETRYC GE R2 THEN          ;IF RETRY COUNT IS MORE THAN HALF LIMIT:
013542 023702 003464          CMP          RETRYC,R2
013546 002403          BLT          50215$
3732 013550          LET CNDPKT := CNDPKT SET.BY #OPP.C
013550 052737 020000 002314          ;SET OPPOSITE BIT FOR RETRY2.
3733 013556          ENDIF          BIS          #OPP.C,CNDPKT
013556          50215$:
3734 013556          ENDIF
013556          50214$:
3735 013556          IF RETRYC EQ #0 ANDB ERCVER NE #0 THEN ;IF THIS IS THE ORIGINAL ERROR THEN:
013556 005737 003464          TST          RETRYC
013562 001011          BNE          50216$
013564 105737 002207          TSTB         ERCVER
013570 001406          BEQ          50216$
3736 013572          ERRSOFT 9,RERM,STAERM          ;REPORT RECOVERABLE ERROR
013572 104457          TRAP          C#ERSOFT
013574 000011          .WORD          9
013576 004632          .WORD          RERM
013600 005626          .WORD          STAERM
3737 013602 004737 014050          JSR PC,RAMDUM          ;GO DO RAM DUMP
3738 013606          ENDIF          ;PROVIDED OPERATOR HAS ENABLED THE REPORT
013606          50216$:
3739 013606          LET RETRYC := RETRYC * #1      ;UPDATE RETRY COUNT.
013606 005237 003464          INC          RETRYC
3740 013612          LET CNDPKT := CNDPKT SET.BY #MOD.C1 ;SET RETRY BIT IN CMD PACKET.
013612 052737 001000 002314          BIS          #MOD.C1,CNDPKT
3741 013620          IFB IREC EQ #0 THEN          ;IF ERROR RECOVERY ENABLED:
013620 105737 002210          TSTB         IREC
013624 001016          BNE          50217$
3742 013626          IF DEVTBL(R5) EQ #NINUSE THEN
013626 026527 002536 177774          CMP          DEVTBL(R5),#NINUSE
013634 001001          BNE          50220$
3743 013636          BR 3$
000413          ENDIF
3744 013640          50220$:
3745 013640          LET ERRREC :B= ERRREC * #1      ;SET ERROR RECOVERY FLAG.

```

```

013640 105237 003475
3746 013644                POP R2,R2                ;POP 2 RTN ADRS FROM STACK.
013644 012602                MOV      (SP),R2
013646 012602                MOV      (SP),R2
3747 013650 004737 010654    JSR PC,EXCUTE    ;GO EXECUTE THE RETRY COMMAND.
3748 013654 000137 011274    JMP GOWAIT      ;GO WAIT FOR INTERRUPT * CHECK STATUS.
3749 013660                ELSE                ;ELSE IF ERROR RECOVERY IS NOT ENABLED:
013660 000402                BR      50217$
013662                50217$:
3750 013662                LET UNREC :B= UNREC * #1    ;SET UNRECOVERABLE ERROR FLAG.
013662 105237 003474                INCB   UNREC
3751 013666                ENDIF
013666                50221$:
3752 013666                ENDIF
013666                50213$:
3753 013666 000207          3$: RTS PC                ;RETURN

```

```

3755      :      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 5, RECOVERABLE ERROR.
3756      :      TAPE POSITION HAS NOT CHANGED. RECOVERY PROCEDURE IS TO LOG THE
3757      :      ERROR AND RE-ISSUE THE ORIGINAL COMMAND.
3758      :      INPUTS:
3759      :      OUTPUTS:
3760      :      REGISTERS:      R2,R4.
3761      :      CALLS:          RTLE, EXCUTE, GOWAIT, DROPU.
3762
3763 013670 004737 014212      TCC5:: JSR PC,RTLE          ;CHECK FOR RETRY LIMIT EXCEEDED
3764 013674 005737 003464      IF RETRYC EQ #0 THEN      ;IF THIS IS THE ORIGINAL ERROR THEN:
                                TST      RETRYC
                                BNE      50222$
3765 013702 104457 014050      ERRSOFT 10,RERM,STAERM    ;REPORT RECOVERABLE ERROR.
                                TRAP     C$ERSOFT
                                .WORD    10
                                .WORD    RERM
                                .WORD    STAERM
3766 013712 004737 014050      JSR      PC,RANDUM        ;GO DO RAM DUMP
3767 013716 005237 003464      ENDIF
                                50222$:
3768 013716 005237 003464      LET RETRYC := RETRYC + #1 ;UPDATE RETRY COUNTER.
                                INC      RETRYC
3769 013722 105737 002210      IFB IREC EQ #0 THEN      ;IF ERROR RECOVERY IS ENABLED:
                                TSTB    IREC
                                BNE     50223$
3770 013730 105237 003475      LET ERRREC :B= ERRREC + #1 ;SET ERROR RECOVERY FLAG.
                                INCB    ERRREC
3771 013734 005265 003330      LET RECCNT(R5) := RECCNT(R5) + #1 ;UPDATE REC COUNT
                                INC     RECCNT(R5)
3772 013740 016577 003330 167446 LET @DATAWT := RECCNT(R5) ;AND INSERT IT INTO WRT BFR
                                MOV     RECCNT(R5),@DATAWT
3773 013746 012602 012602      POP R2,R2                ;POP 2 RTN ADRS FROM STACK.
                                MOV     (SP)+,R2
                                MOV     (SP)+,R2
3774 013752 004737 010654      JSR PC,EXCUTE            ;GO RE-ISSUE THE COMMAND.
3775 013756 000137 011274      JMP GOWAIT              ;GO WAIT FOR INTERRUPT + CHECK STATUS.
3776 013762 000402 011274      ELSE                    ;ELSE IF ERROR RECOVERY IS NOT ENABLED:
                                BR      50224$
                                50223$:
3777 013764 105237 003474      LET UNREC :B= UNREC + #1 ;SET UNRECOVERABLE ERROR FLAG.
                                INCB    UNREC
3778 013770 000207 000207      ENDIF
3779 013770 000207 000207      RTS PC                  ;RETURN.
                                50224$:
3780
3781

```

```

3783      ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 6, UNRECOVERABLE ERROR.
3784      ; TAPE POSITION HAS BEEN LOST. THE ONLY VALID RECOVERY PROCEDURE
3785      ; IS TO REWIND AND START OVER AT BOT UNLESS THE TAPE HAS LABELS OR
3786      ; SEQUENCE NUMBERS. THIS DIAGNOSTIC WILL REWIND AND RETRY THE
3787      ; COMMAND ONLY IF DENSITY CHECK IS SET, OTHERWISE THE UNIT WILL BE
3788      ; DROPPED FROM THE TEST SEQUENCE.
3789      ; INPUTS:
3790      ; OUTPUTS:
3791      ; REGISTERS: R2, R4
3792      ; CALLS: RTLE, WSSR, EXECUTE, GOWAIT, DROPU
3793
3794 013772 TCC6:: LET @TSDB(R5) := @RWCPK ;ISSUE A REWIND COMMAND.
          013772 012775 002334 002456 ;MOV @RWCPK,@TSDB(R5)
3795 014000 004737 011672 JSR PC,WSSR ;WAIT FOR SUBSYSTEM READY,
3796 014004 104455 ERRDF 11,URERM,STAERM ;REPORT UNRECOVERABLE ERROR.
          014006 000013 ;TRAP C$ERDF
          014010 004654 ;.WORD 11
          014012 005626 ;.WORD URERM
          014014 004737 014050 JSR PC,RAMDUM ;GO DO RAM DUMP
          014020 004737 017402 JSR PC,DROPU ;DROP THE UNIT.
3799 014024 000207 RTS PC ;RETURN

```

```

3801      ;      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 7, FATAL SUBSYSTEM
3802      ;      ERROR.  THE SUBSYSTEM IS INCAPABLE OF PROPERLY PERFORMING
3803      ;      COMMANDS OR AT LEAST ITS INTEGRITY IS SERIOUSLY QUESTIONABLE.
3804      ;      REFER TO THE FATAL CLASS CODE FIELD IN THE TSSR REGISTER FOR
3805      ;      ADDITIONAL INFORMATION ON THE TYPE OF FATAL ERROR.
3806      ;      INPUTS:
3807      ;      OUTPUTS:
3808      ;      REGISTERS:      R2, R4
3809      ;      CALLS:
3810
3811      TCC7::  ERRDF 12,FATSM,STAERM      ;REPORT FATAL SUBSYSTEM ERROR.
3812      014026 104455                      TRAP      C$ERDF
3813      014030 000014                      .WORD    12
3814      014032 004455                      .WORD    FATSM
3815      014034 005626                      .WORD    STAERM
3816
3817      3812 014036 004737 014050      JSR      PC,RAMDUM      ;GO DO RAM DUMP
3818      3813 014042 004737 017402      JSR      PC,DROPU      ;DROP THE UNIT.
3819      3814 014046 000207                      RTS PC      ;RETURN.
3820
3821
3822      RAMDUM::
3823      IFB RAMWRT NE #0 THEN
3824
3825      PRINTX #RAMFHR
3826
3827      TSTB RAMWRT
3828      BEQ 50225$
3829
3830      MOV #RAMFHR, -(SP)
3831      MOV #1, -(SP)
3832      MOV SP, R0
3833      TRAP C$PNTX
3834      ADD #4, SP
3835
3836      MOV #8.,RAMSIZ      ;RAM FIELD IS 8 BYTES LONG
3837      MOV #20,RAMHLD     ;COMMAND PACKET ADDRESS
3838      JSR PC,RAMER      ;READ AND PRINT THEM
3839      MOV #200,RAMHLD   ;CHARACTERISTICS PACKET ADDRESS
3840      JSR PC,RAMER      ;READ AND PRINT THEM
3841      MOV #14.,RAMSIZ   ;RAM FIELD IS 8 BYTES LONG
3842      MOV #214,RAMHLD  ;MESSAGE PACKET ADDRESS
3843      JSR PC,RAMER      ;READ AND PRINT THEM
3844      MOV #16.,RAMSIZ   ;RAM FIELD IS SIXTEEN BYTES LONG
3845      MOV #60,RAMHLD   ;SENCE BYTES ADDRESS
3846      JSR PC,RAMER      ;READ AND PRINT THEM
3847
3848      PRINTX #RAMLIN
3849
3850      MOV #RAMLIN, -(SP)
3851      MOV #1, -(SP)
3852      MOV SP, R0
3853      TRAP C$PNTX
3854      ADD #4, SP
3855
3856      3834 014210                      ENDIF
3857      3835 014210 000207                      RTS      PC
3858      3836
3859
3860      50225$:

```

```

3838      :      SUBROUTINE TO CHECK FOR RETRY LIMIT EXCEEDED. PRINTS ERROR MESSAGE
3839      :      IF EXCEEDED AND DROP UNIT UNLESS COMMAND IS A READ.
3840      :      INPUTS:
3841      :      OUTPUTS:
3842      :      REGISTERS:      R2, R4.
3843      :      CALLS:          DROPU
3844
3845
3846      014212      :      RTLE:: IF CMDLG EQ #0 THEN          ;IF CMD IS NOT A READ OR WRITE THEN:
3847      014212      005737      003434          TST      CMDLG
3848      014216      001012          BNE      50226$
3849      014220      ERRDF 11,URERM,STAERM          ;REPORT UNRECOVERABLE ERROR.
3850      014220      104455          TRAP      C$ERDF
3851      014222      000013          .WORD    11
3852      014224      004654          .WORD    URERM
3853      014226      005626          .WORD    STAERM
3854      014230      004737      014050          JSR      PC,RAMDUM          ;GO DO RAM DUMP
3855      014234      004737      017402          JSR      PC,DROPU          ;DROP THE UNIT.
3856      014240      POP R2
3857      014240      012602          MOV      (SP)+,R2
3858      014242      000441          BR RTLRTN          ;AND RETURN.
3859      014244      ENDIF
3860      014244          LET RWERR :B= RWERR + #1          ;SET READ/WRITE ERROR FLAG.
3861      014244      105237      003473          INCB    RWERR
3862      014250          IF CMDLG EQ #2 THEN          ;IF CMD IS A WRT OR WTM:
3863      014250      023727      003434      000002          CMP      CMDLG,#2
3864      014256      001020          BNE      50227$
3865      014260          IF RETRYC EQ #WRECL THEN          ;IF RETRY COUNT HAS REACHED LIMIT:
3866      014260      023727      003464      000020          CMP      RETRYC,#WRECL
3867      014266      001013          BNE      50230$
3868      014270          LET UNREC :B= UNREC + #1          ;SET UNRECOVERABLE FLAG
3869      014270      105237      003474          INCB    UNREC
3870      014274          ERRDF 14,RLEXM,STAERM          ;REPORT RETRY LIMIT EXCEEDED.
3871      014274      104455          TRAP      C$ERDF
3872      014276      000016          .WORD    14
3873      014300      004372          .WORD    RLEXM
3874      014302      005626          .WORD    STAERM
3875      014304      004737      014050          JSR      PC,RAMDUM          ;GO DO RAM DUMP
3876      014310      004737      017402          JSR      PC,DROPU          ;DROP THE UNIT.
3877      014314      012602          MOV      (SP)+,R2
3878      014316          ENDIF
3879      014316          ELSE
3880      014316          50230$:
3881      014316      000413          ;ELSE - CMD IS A READ:
3882      014320          BR      50231$
3883      014320          50227$:
3884      014320      023727      003464      000020          ;IF RETRY COUNT HAS REACHED LIMIT:
3885      014326      001007          CMP      RETRYC,#RRECL
3886      014330          BNE      50232$
3887      014330          LET UNREC :B= UNREC + #1          ;SET UNRECOVERABLE FLAG
3888      014330      105237      003474          INCB    UNREC
3889      014334          ERRHRD 14,RLEXM,STAERM          ;REPORT RECOVERABLE ERROR.
3890      014334      104456          TRAP      C$ERHRD
3891      014336      000016          .WORD    14
3892      014340      004372          .WORD    RLEXM
3893      014342      005626          .WORD    STAERM

```



```
3866 014344          POP R2
      014344 012602
3867 014346          ENDIF
      014346
3868 014346          ENDIF
      014346
3869 014346 000207   RTLRTN: RTS PC          ;RETURN

MOV    (SP)+,R2
50232$:
50231$:
```

```

3871      ; SUBR TO REWRITE A BAD, BUT RECOVERABLE WRITTEN RECORD.
3872      ; REWRITE RECORD ON SAME SPOT; REPEAT 4 TIMES.
3873      ; IF ALL 4 REPEATS GOOD, RECORD IS RECOVERED
3874      ; AND A RECOVERABLE WRITE ERROR IS LOGGED.
3875      ; IF ANY OF 4 REPEATS BAD, ERASE BAD RECORD, LOG SUSPECTED
3876      ; BAD SPOT, RETRY AGAIN, RETRY 4 TIMES, UP TO 4 REPEATS EACH.
3877      ; IF RECORD NOT GOOD AFTER 4 RETRIES, ERASE IT, EXIT WITH
3878      ; ERROR FLAG WRTYER SET, PRINTING RETRY FAILED.
3879      ; THIS ALL SCHEME IS REENTERED 20 TIMES MAX, IE 20 BAD
3880      ; SPOTS MAX ARE ALLOWED.
3881
3882      ; INPUTS:
3883      ; OUTPUTS:
3884      ; REGISTERS:      R3,R4
3885      ; CALLS:          BORERS, REWRT
3886
3887 014350      ; WRTY:: IF DEVTBL(R5) NE #NINUSE THEN                ; IF DRIVE NOT DROPPED
      014350 026527 002536 177774      ;                          CMP      DEVTBL(R5),#NINUSE
      014356 001537                    ;                          BEQ      50233$
3888 014360      ; BEGIN RETRY
3889 014360      ; REPEAT
      014360                    ;                          50235$:
3890 014360      ; BEGIN REPEAT
3891 014360      ; REPEAT
      014360                    ;                          50237$:
3892 014360 004737 015146      ; JSR PC,BORERS                ;BACKSPACE/ERASE ONE RECORD
3893 014364      ; LET WRTYER :B= #0                ;CLEAR WRITE RETRY ERROR
      014364 105037 003470      ;                          CLRB   WRTYER
3894 014370 004737 015322      ; JSR PC,REWRT                ;REWRITE RECORD ON SAME SPOT
3895 014374      ; IF DEVTBL(R5) EQ #NINUSE THEN
      014374 026527 002536 177774      ;                          CMP      DEVTBL(R5),#NINUSE
      014402 001003                    ;                          BNE      50240$
3896 014404      ; LET RPTCNT :B= #3
      014404 112737 000003 003466      ;                          MOVB   #3,RPTCNT
3897 014412      ; ENDIF
3898 014412      ; LET RPTCNT :B= RPTCNT * #1        ;COUNT REPEATS
      014412 105237 003466      ;                          INCB   RPTCNT
3899 014416      ; UNTILB RPTCNT EQ #4 ORB WRTYER NE #0 ;LIMIT: 4 REPEATS OR RECOVERED
      014416 123727 003466 000004      ;                          CMPB   RPTCNT,#4
      014424 001403                    ;                          BEQ      50241$
      014426 105737 003470      ;                          TSTB   WRTYER
      014432 001752                    ;                          BEQ      50237$
      014434                    ;                          50241$:
3900 014434      ; END REPEAT
3901 014434      ; LET RETRYC := RETRYC * #1        ;COUNT RETRIES
      014434 005237 003464      ;                          INC     RETRYC
3902 014440      ; IF DEVTBL(R5) EQ #NINUSE THEN
      014440 026527 002536 177774      ;                          CMP      DEVTBL(R5),#NINUSE
      014446 001002                    ;                          BNE      50242$
3903 014450 000502      ; BR      3$
3904 014452      ; ELSE
      014452 000400                    ;                          BR      50243$
      014454                    ;                          50242$:
3905 014454      ; ENDIF
      014454                    ;                          50243$:
    
```

3906	014454			IFB WRTYER EQ #0 THEN	:	TSTB	WRTYER	
	014454	105737	003470			BNE	50244\$	
	014460	001001				:	EXIT RETRY LOOP IF RECOVERED	
3907	014462			LEAVE RETRY	:	BR	50234\$	
	014462	000457						
3908	014464			ELSE	:		50244\$:	
	014464							
3909	014464			IFB ERCVER NE #0 THEN	:	TSTB	ERCVER	
	014464	105737	002207			BEG	50246\$	
	014470	001415				:	PRINT SUSPECTED BAD SPOT	
3910	014472			PRINTB @BTMSG1,RETRYC,<B,RPTCNT>	:	CLR	-(SP)	
	014472	005046				BISB	RPTCNT,(SP)	
	014474	153716	003466			MOV	RETRYC, -(SP)	
	014500	013746	003464			MOV	@BTMSG1, -(SP)	
	014504	012746	014734			MOV	#3, -(SP)	
	014510	012746	000003			MOV	SP,R0	
	014514	010600				TRAP	C#PNTB	
	014516	104414				ADD	@10,SP	
	014520	062706	000010					
3911	014524			ENDIF	:		50246\$:	
	014524					:	ON FIRST RETRY, LOGG BAD SPOT	
3912	014524			IF RETRYC EQ #1 THEN	:	CMP	RETRYC,#1	
	014524	023727	003464			BNE	50247\$	
	014532	001021	000001			:	BTPT IS BOTH THE BAD SPOT COUNTER	
3913	014534			LET BTPT := BTADDR(R5)	:	MOV	BTADDR(R5),BTPT	
	014534	016537	002550			:	AND THE LOGGING INDEX	
3914	014542			LET R4 := @BTPT * #2	:	MOV	@BTPT,R4	
	014542	017704	166750			ADD	@2,R4	
	014546	062704	000002			:	MOV R4,@BTPT	
3915	014552			LET @BTPT := R4	:			
	014552	010477	166740			:		
3916	014556			IF R4 LOS #40. THEN	:	CMP	R4,#40.	
	014556	020427	000050			BHI	50250\$	
	014562	101005				:	STORE FIRST 20 BAD SPOTS	
3917	014564			LET R3 := BTPT	:	MOV	BTPT,R3	
	014564	013703	003516			:		
3918	014570			LET R4 := R4 * R3	:	ADD	R3,R4	
	014570	060304				:		
3919	014572			LET (R4) := RECCNT(R5)	:	MOV	RECCNT(R5),(R4)	
	014572	016514	003330					
3920	014576			ENDIF	:		50250\$:	
	014576					:		
3921	014576			ENDIF	:		50247\$:	
	014576					:	ERASE FLAG TO ERASE BAD RECORD	
3922	014576			LET ERSFLG :B= ERSFLG * #1	:	INCB	ERSFLG	
	014576	105237	003533			:	CANCEL "LOG" ERROR FLAG ON FAILING RETRY	
3923	014602			LET RWERR :B= #0	:	CLRB	RWERR	
	014602	105037	003473			:	CLEAR REPEAT COUNT FOR NEXT RETRY	
3924	014606			LET RPTCNT :B= #0	:	CLRB	RPTCNT	
	014606	105037	003466			:		
3925	014612			ENDIF	:		50245\$:	
	014612					:	LIMIT: 4 RETRIES	
3926	014612			UNTIL RETRYC EQ #4	:	CMP	RETRYC,#4	
	014612	023727	003464			BNE	50235\$	
	014620	001257	000004			:		
3927	014622			END RETRY	:		50234\$:	
	014622							

```

3928 014622          IFB WRTYER NE #0 THEN          ;
      014622 105737 003470          ;           TSTB   WRTYER
      014626 001413          ;           BEQ   50251#
3929 014630          IFB ERCVER NE #0 THEN          ;
      014630 105737 002207          ;           TSTB   ERCVER
      014634 001410          ;           BEQ   50252#
3930 014636          PRINTB #BTMSG3          ;PRINT RETRY FAILED
      014636 012746 015076          ;           MOV   #BTMSG3,-(SP)
      014642 012746 000001          ;           MOV   #1,-(SP)
      014646 010600          ;           MOV   SP,R0
      014650 104414          ;           TRAP  C#PNTB
      014652 062706 000004          ;           ADD   #4,SP
3931 014656          ENDIF          ;           50252#:
      014656          ;           50251#:
3932 014656          ENDIF          ;           50233#:
      014656          ;           50233#:
3933 014656          ENDIF          ;           50253#:
3934 014656          3#: LET -(SP) := #10.          ;LOOP CONTROL
      014656 012746 000012          ;           MOV   #10,-(SP)
3935 014662          REPEAT          ;           50253#:
      014662          DELAY 250.          ;WAIT FOR THE UNIT TO STOP
3936 014662          ;           MOV   #250,(PC)+
      014662 012727 000372          ;           .WORD 0
      014666 000000          ;           MOV   L#DLY,(PC)+
      014670 013727 002116          ;           .WORD 0
      014674 000000          ;           DEC   -6(PC)
      014676 005367 177772          ;           BNE   -4
      014702 001375          ;           DEC   -22(PC)
      014704 005367 177756          ;           BNE   -20
      014710 001367          ;
3937 014712          LET (SP) := (SP) - #1          ;ONE LESS ITERATION
      014712 005316          ;           DEC   (SP)
3938 014714          UNTIL (SP) EQ #0          ;'TILL ZERO!
      014714 005716          ;           TST   (SP)
      014716 001361          ;           BNE   50253#
3939 014720          LET SP := SP + #2          ;POP THE STACK
      014720 062706 000002          ;           ADD   #2,SP
3940 014724 000207          RTS PC
3941
3942
3943
3944
3945
3946
3947 014726 000000          WTYCMD: .WORD 0          ;STORAGE FOR WRITE CMD WHILE RETRYING
3948 014730 000000          WTYWRD: .WORD 0          ;STORAGE FOR WRITE CMD WORD WHILE RETRYING
3949 014732 000000          WTYBRF: .WORD 0          ;STORAGE FOR WRITE BPCR WHILE RETRYING
3950
3951
3952 014734          045          101          123          BTMSG1: .ASCIZ /#ASUSPECT BAD SPOT AFTER #D1#A RETRY, #D1#A REPEAT#N/
      014737          125          123          120
      014742          105          103          124
      014745          040          102          101
      014750          104          040          123
      014753          120          117          124
      014756          040          101          106

```

	014761	124	105	122	
	014764	040	045	104	
	014767	061	045	101	
	014772	040	122	105	
	014775	124	122	131	
	015000	054	040	045	
	015003	104	061	045	
	015006	101	040	122	
	015011	105	120	105	
	015014	101	124	045	
	015017	116	000		
3953	015021	045	116	045	BTMSG2: .ASCIZ /#N#ABAD TAPE OVERFLOW: CHANGE CARTRIDGE!#N#N/
	015024	101	102	101	
	015027	104	040	124	
	015032	101	120	105	
	015035	040	117	126	
	015040	105	122	106	
	015043	114	117	127	
	015046	072	040	103	
	015051	110	101	116	
	015054	107	105	040	
	015057	103	101	122	
	015062	124	122	111	
	015065	104	107	105	
	015070	041	045	116	
	015073	045	116	000	
3954	015076	045	101	122	BTMSG3: .ASCIZ /#ARETRY FAILED ON BAD SPOT...ERASED!#N#N/
	015101	105	124	122	
	015104	131	040	106	
	015107	101	111	114	
	015112	105	104	040	
	015115	117	116	040	
	015120	102	101	104	
	015123	040	123	120	
	015126	117	124	056	
	015131	056	056	105	
	015134	122	101	123	
	015137	105	104	041	
3955	015142	045	116	000	.EVEN

```

3957 ; SUBR TO BACSPACE ONE RECORD
3958 ; IF THE ERASE FLAG IS SET, THEN ERASE THAT RECORD
3959 ; INPUTS: ERSFLG 1 = DO ERASE
3960 ;
3961 ; OUTPUTS:
3962 ; REGISTERS:
3963 ; CALLS: EXECUTE, GOWAIT, CKHAE
3964 015146 BORERS:: LET PCMDWD := CMDWRD ;SET COMMAND TO SPACE REV
015146 013737 003426 003432 MOV CMDWRD,PCMDWD
3965 015154 LET CMDWRD := #SRR ;
015154 012737 104410 003426 MOV #SRR,CMDWRD
3966 015162 LET CNDPKT := CMDWRD CLR.BY #BRF.C ;
015162 013737 003426 002314 MOV CMDWRD,CNDPKT
015170 042737 004000 002314 BIC #BRF.C,CNDPKT
3967 015176 LET CMDSAV := CNDPKT ;
015176 013737 002314 003430 MOV CNDPKT,CMDSAV
3968 015204 LET CNDPKT.CP.ADL := #1 ;
015204 012737 000001 002316 MOV #1,CNDPKT.CP.ADL
3969 015212 LET CMDLG := #0 ;
015212 005037 003434 CLR CMDLG
3970 015216 JSR PC,CMDAC ;
3971 015222 JSR PC,EXECUTE ;
3972 015226 JSR PC,GOWAIT ;
3973 015232 JSR PC,CKHAE ;
3974 015236 IFB ERSFLG NE #0 THEN ;WHEN ERASE FLAG IS SET, DO ERASE
015236 105737 003533 TSTB ERSFLG
015242 001426 BEQ 50254$
3975 015244 LET PCMDWD := CMDWRD ;
015244 013737 003426 003432 MOV CMDWRD,PCMDWD
3976 015252 LET CMDWRD := #ERS ;
015252 012737 100411 003426 MOV #ERS,CMDWRD
3977 015260 LET CNDPKT := CMDWRD ;
015260 013737 003426 002314 MOV CMDWRD,CNDPKT
3978 015266 LET CMDSAV := CNDPKT ;
015266 013737 002314 003430 MOV CNDPKT,CMDSAV
3979 015274 JSR PC,CMDAC ;
3980 015300 JSR PC,EXECUTE ;
3981 015304 JSR PC,GOWAIT ;
3982 015310 JSR PC,CKHAE ;
3983 015314 LET ERSFLG :B= #0
015314 105037 003533 CLRB ERSFLG
3984 015320 ENDIF
015320 50254$:
3985 015320 RTS PC
000207
3986 ; SUBR TO REWRITE A BADLY WRITTEN RECORD
3987 ;
3988 015322 REWRT: IF DEVTBL(R5) NE #NINUSE THEN ;IF DRIVE NOT DROPPED
015322 026527 002536 177774 CMP DEVTBL(R5),#NINUSE
015330 001441 BEQ 50255$
3989 015332 LET PCMDWD := CMDWRD ;RESTORE WRITE COMMAND PACKET
015332 013737 003426 003432 MOV CMDWRD,PCMDWD
3990 015340 LET CMDWRD := WTYWRD ;
015340 013737 014730 003426 MOV WTYWRD,CMDWRD
3991 015346 LET CNDPKT := WTYCMD ;
015346 013737 014726 002314 MOV WTYCMD,CNDPKT
3992 015354 LET CMDSAV := CNDPKT ;
015354 013737 002314 003430 MOV CNDPKT,CMDSAV
    
```

3993	015362				LET CMDPKT+CP.ADL := DATAWT	;		
	015362	013737	003414	002316			MOV	DATAWT,CMDPKT+CP.ADL
3994	015370				LET CMDPKT+CP.CNT := WTYBRF	;		
	015370	013737	014732	002322			MOV	WTYBRF,CMDPKT+CP.CNT
3995	015376				LET CMDLG := #2	;		
	015376	012737	000002	003434			MOV	#2,CMDLG
3996	015404	004737	007600		JSR PC,CMDAC			
3997	015410	004737	010654		JSR PC,EXCUTE			
3998	015414				IF DEVTBL(R5) NE #NINUSE THEN			
	015414	026527	002536	177774			CMP	DEVTBL(R5),#NINUSE
	015422	001404					BEQ	50256#
3999	015424	004737	011274		JSR PC,GOWAIT	;		
4000	015430	004737	017706		JSR PC,CKHAE	;		
4001	015434				ENDIF			50256#:
	015434							
4002	015434				ENDIF			50255#:
	015434							
4003	015434	000207			RTS PC			

```

4005      ; SUBROUTINE TO LOG BYTES READ/WRITTEN.
4006      ; ALSO UPDATES READ/WRITE ERROR COUNTERS.
4007      ;
4008      ; INPUTS:
4009      ; OUTPUTS:
4010      ; REGISTERS:      R2, R3, R4.
4011      ; CALLS:
4012 015436 LOG:: IFB ERLOG EQ #0 THEN      ;IF DATA AND ERRORS HAVE NOT BEEN LOGGED THEN:
      015436 105737 003472      TSTB ERLOG
      015442 001126      BNE 50257#
4013 015444      LET ERLOG := ERLOG + #1      ;SET LOG DONE FLAG.
      015444 105237 003472      INCB ERLOG
4014 015450      LET R4 := CMDLG      ;GET CURRENT CMD LOGGING CODE.
      015450 013704 003434      MOV CMDLG,R4
4015 015454      IF R4 NE #0 THEN      ;IF THERE IS A CODE THEN:
      015454 005704      TST R4
      015456 001520      BEQ 50260#
4016 015460      LET R4 := R4 - #2      ;ADJUST THE CODE FOR TABLE INDEX.
      015460 162704 000002      SUB #2,R4
4017 015464      LET R2 := R5 + BINC(R4) + #CNTBGN ;R2 = ADR OF BYTE COUNT LSW.
      015464 010502      MOV R5,R2
      015466 066402 015722      ADD BINC(R4),R2
      015472 062702 002560      ADD #CNTBGN,R2
4018 015476      LET (R2) := (R2) + BRFCNT      ;ADD BRFCNT TO LSW.
      015476 063712 003424      ADD BRFCNT,(R2)
4019 015502      IF MSGPKT*MS.RFC LOS BRFCNT THEN ;IF THE RFC IS LOWER OR THE SAME AS BRFCNT THEN:
      015502 023737 002344 003424      CMP MSGPKT*MS.RFC,BRFCNT
      015510 101002      BHI 50261#
4020 015512      LET (R2) := (R2) - MSGPKT*MS.RFC ;SUBTRACT RFC FROM EXPECTED BRFC.
      015512 163712 002344      SUB MSGPKT*MS.RFC,(R2)
4021 015516      ENDIF
      015516
4022 015516      LET R3 := R2 + #10      ;R3 = ADR OF 2ND WORD.
      015516 010203      MOV R2,R3
      015520 062703 000010      ADD #10,R3
4023 015524      WHILE (R2) GT #999. DO
      015524      015524 021227 001747      50261#:
      015530 003404      CMP (R2),#999.
4024 015532      LET (R2) := (R2) - #1000. ;UPDATE BYTE COUNT
      015532 162712 001750      BLE 50263#
4025 015536      LET (R3) := (R3) + #1 ;2ND WORD.
      015536 005213      SUB #1000.,(R2)
4026 015540      ENDDO
      015540 000771      INC (R3)
      015542      BR 50262#
4027 015542      LET R2 := R3 + #10 ;R2 = ADR OF 3RD WORD.
      015542 010302      MOV R3,R2
      015544 062702 000010      ADD #10,R2
4028 015550      WHILE (R3) GT #999. DO
      015550      015550 021327 001747      50262#:
      015554 003404      CMP (R3),#999.
4029 015556      LET (R3) := (R3) - #1000. ;UPDATE BYTE COUNT
      015556 162713 001750      BLE 50265#
4030 015562      LET (R2) := (R2) + #1 ;3RD WORD.
      015562 005212      SUB #1000.,(R3)
      INC (R2)

```



```

4031 015564          ENDDO
      015564 000771
      015566
4032 015566          LET R3 := R2 * #10          ;R3 = ADR OF 4TH WORD.
      015566 010203
      015570 062703 000010
      015574          WHILE (R2) GT #999. DO
      015574 021227 001747
      015600 003404
      015602          LET (R2) := (R2) - #1000. ;UPDATE BYTE COUNT
      015602 162712 001750
      015606          LET (R3) := (R3) * #1          ;4TH WORD.
      015606 005213
4036 015610          ENDDO
      015610 000771
      015612
4037 015612          IFB RWERR NE #0 THEN          ;IF R/W ERROR, UPDATE ERROR COUNT.
      015612 105737 003473
      015616 001440
      015620          LET R2 := R5 * EINC(R4) * #WRREC ;R2 = ADR OF COUNTER.
      015620 010502
      015622 066402 015730
      015626 062702 002720
      015632          IFB UNREC NE #0 THEN          ;IS THE ERROR UNRECOVERABLE?
4039 015632 105737 003474
      015636 001404
      015640          LET R2 := R2 * #10          ;YES, POINT TO NEXT COUNTER.
4040 015640 062702 000010
      015644          LET (R2) := (R2) * #1          ;UPDATE THE ERROR COUNTER
4041 015644 005212
      015646          ELSE          ;ELSE - IF ERROR IS RECOVERABLE:
4042 015646 000424
      015650
      015650          LET (R2) := (R2) * #1          ;UPDATE THE ERROR COUNTER
4043 015650 005212
      015652          IFB IREC EQ #0 THEN          ;IF ERROR RECOVERY IS ENABLED:
4044 015652 105737 002210
      015656 001020
      015660          IFB DROPED EQ #0 ANDB ERCVER NE #0 THEN ;IF UNIT HAS NOT BEEN DROPPED:
4045 015660 105737 003527
      015664 001015
      015666 105737 002207
      015672 001412
      015674          PRINTB #NURTY1,RETRYC          ;PRINT # OF RETRIES TO RECOVER
4046 015674 013746 003464
      015700 012746 005155
      015704 012746 000002
      015710 010600
      015712 104414
      015714 062706 000006
      015720          ENDDO
4047 015720          ENDDO
      015720
4048 015720          ENDDO
      015720
4049 015720          ENDDO
      015720

```

```

4050 015720          ENDIF
      015720
4051 015720          ENDIF
      015720
4052 015720          ENDIF
      015720
4053 015720 000207
4054
4055 015722 000000      ; BINC:
4056 015724 000040      0
4057 015726 000100      40
4058
4059 015730 000000      ; EINC:
4060 015732 000020      0
4061 015734 000040      20
4062
4063
4064

RTS PC
INDEXES TO BYTE COUNTERS.
;WRITE.
;READ REV.
;READ FWD.
INDEXES TO READ/WRITE ERROR COUNTERS.
;WRITE.
;READ REV.
;READ FWD.

50270$:
50260$:
50257$:

```

```

4066                                     .SBTTL RAMER - READ AND DISPLAY SELECTED RAM
4067                                     ;*
4068                                     ;ROUTINE TO READ THE SELECTED RAM LOCATIONS
4069                                     ;-
4070 RAMER:: MOV      R5,-(SP)
4071          MOV      R4,-(SP)
4072          MOV      R3,-(SP)
4073          MOV      R2,-(SP)
4074          MOV      R1,-(SP)
4075          MOV      @RAMDATA,R1      ;ADDRESS TO SAVE THE RAM DATA
4076          MOV      RAMHLD,R2      ;BYTE ADDRESS OF THE FIRST RAM DATA
4077          MOV      RAMSIZ,R3      ;SET THE SIZE OF THE READ UP
4078          MOV      TSDB(R5),R4      ;MOV THE TSDB ADDRESS INTO R4
4079          INC      R4              ;ADD 1 TO IT
4080          10$: NOP
4081          JSR      PC,WSSR          ;WAIT FOR THE SSR TO SET
4082          MOV      R2,(R4)          ;SELECT NEXT RAM ADDRESS
4083          JSR      PC,WSSR          ;WAIT FOR SSR TO SET
4084          MOV      @TSBA(R5),(R1)+  ;READ THE RAM DATA
4085          20$: ADD      #1,R2      ;ADDRESS OF THE NEXT RAM LOCATION
4086          SOB      R3,10$          ;NUMBER OF LOCATIONS COUNTER
4087          MOV      RAMSIZ,R4      ;GET THE RAM SIZE
4088          MOV      RAMHLD,R2      ;GET THE STARTING RAM ADDRESS
4089          ADD      R2,R4          ;CALCULATE THE END ADDRESS
4090          SUB      #1,R4          ;CORRECT VALUE OF PRINTOUT
4091          PRINTX   @RAMIOP,R2,R4    ;RAM ADDRESS = 10 - 17, ETC.
4092          MOV      R4,-(SP)
4093          MOV      R2,-(SP)
4094          MOV      @RAMIOP,-(SP)
4095          MOV      #3,-(SP)
4096          MOV      SP,R0
4097          TRAP    C$PNTX
4098          ADD      #10,SP
4099          MOV      @RAMDATA,R1      ;ADDRESS OF WHERE RAM DATA IS
4100          MOV      RAMSIZ,R3      ;THE SIZE OF THE RAM FIELD READ
4101          30$: CLR      R4          ;NO EXTRA DATA LEFT OVER
4102          MOV      (R1)+,R4        ;PICK UP BYTE OF RAM DATA
4103          BIC      #177400,R4      ;GET RID OF SIGN EXTEND
4104          PRINTX   @RAMPD,R4        ;"010 211 111 222 377 000 123 134 ETC."
4105          MOV      R4,-(SP)
4106          MOV      @RAMPD,-(SP)
4107          MOV      #2,-(SP)
4108          MOV      SP,R0
4109          TRAP    C$PNTX
4110          ADD      #6,SP
4111          SOB      R3,30$          ;LOOP UNTIL ALL PRINTED
4112          MOV      (SP)+,R1
4113          MOV      (SP)+,R2
4114          MOV      (SP)+,R3
4115          MOV      (SP)+,R4
4116          MOV      (SP)+,R5
4117          50$: RTS      PC          ;RETURN
4118          ; IF A WRITE/VERIFY COMMAND IS ISSUED, CONTROL IS THEN
4119          ; TRANSFERRED TO THIS SUBROUTINE TO READ REVERSE, CHECK DATA,
4120          ; READ FORWARD, CHECK DATA, THEN CONTINUE TO NEXT COMMAND.
4121          ; INPUTS:
4122          ; OUTPUTS:
    
```



```

4131      ;      SUBROUTINE TO EXECUTE THE READ AND VERIFY, FORWARD OR REVERSE.
4132      ;      INPUTS:
4133      ;      OUTPUTS:
4134      ;      REGISTERS:      R2
4135      ;      CALLS:          CMDAC, FIRSTU, VFISU, NEXTU, CKHAE.
4136
4137 016254      VFEXC:: LET CMDPKT := CMDWRD CLR.BY #BRF.C ;COMMAND PACKET = READ REV OR FWD.
016254      013737      003426      002314      MOV      CMDWRD,CMDPKT
016262      042737      004000      002314      BIC      #BRF.C,CMDPKT
4138 016270      IFB SWBFLG NE #0 THEN ;IF BYTES ARE TO BE SWAPPED:
016270      105737      003524      TSTB     SWBFLG
016274      001403      BEQ      50300$
4139 016276      LET CMDPKT := CMDPKT SET.BY #SWB.C ;SET SWAB BIT IN CMD PACKET.
016276      052737      010000      002314      BIS      #SWB.C,CMDPKT
4140 016304      ENDIF
016304
4141 016304      LET CMDSAV := CMDPKT ;SAVE COMMAND PACKET 1ST WORD.
016304      013737      002314      003430      MOV      CMDPKT,CMDSAV
4142 016312      MOV      DATARD,CMDPKT+CP.ADL ;SAVE BUFFER START ADDRESS.
016312      013737      003416      002316      LET NCNT := #0 ;CLEAR NUMBER OF OPERATIONS.
4143 016320      005037      003420      CLR      NCNT
4144 016324      WHILE NCNT LT NCNT1 DO ;WHILE THERE ARE RECORDS REMAINING:
016324      023737      003420      003422      50301$:
016332      002101      CMP      NCNT,NCNT1
016334      004737      007600      BGE      50302$
4145 016334      JSR PC,CMDAC ;STORE CMD ASCII IN ERROR MSG.
4146 016340      TSTB STREAM ;CHECK IF WE ARE STREAMING
4147 016344      001006      BNE 1$ ;BRANCH OVER DEVTBL CHECK. THIS ENABLES
4148 ;US TO TEST ONE DRIVE AT A TIME.
4149 016346      JSR PC,FIRSTU ;SET UP FOR FIRST UNIT.
4150 016352      WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE DEVICES REMAINING:
016352      026527      002536      177777      50303$:
016360      001445      CMP      DEVTBL(R5),#END
4151 016362      1$: IF #MOD.CO SETIN CMDWRD THEN ;IF CMD IS REVERSE THEN:
016362      032737      000400      003426      BEQ      50304$
016370      001421      BIT      #MOD.CO,CMDWRD
4152 016372      IF #X0.BOT NOTSETIN EOTFLG(R5) THEN ;IF NOT AT BOT
016372      032765      000002      003506      BEQ      50305$
016400      001014      BIT      #X0.BOT,EOTFLG(R5)
4153 016402      IF #X0.EOT SETIN EOTFLG(R5) THEN ;BUT IF AT EOT
016402      032765      000001      003506      BNE      50306$
016410      001406      BIT      #X0.EOT,EOTFLG(R5)
4154 016412      IFB ALLEOT NE #0 THEN ;AND ALL OTHERS AT EOT
016412      105737      003531      TSTB     ALLEOT
016416      001402      BEQ      50310$
4155 016420      JSR PC,VFISU ;THEN READ VERIFY
4156 016424      ENDIF ;IF NOT ALL AT EOT, FREEZE UNIT(S) AT E
016424      50310$:
4157 016424      ELSE ;IF NOT AT BOT AND
016424      000402      BR      50311$
016426      50307$:
4158 016426      004737      016540      JSR PC,VFISU ;NOT AT EOT, READ VFY
4159 016432      ENDIF
016432      50311$:
4160 016432      ENDIF
016432      50306$:

```

```

4161 016432          ELSE          ;ELSE IF CMD IS NOT REVERSE:
      016432 000412          BR      50312$
      016434          50305$:
4162 016434          IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN
      016434 032765 000001 003506          BIT      #X0.EOT,EOTFLG(R5)
      016442 001404          BEQ      50313$
      016444 032737 000001 003426          BIT      #CMD.CO,CMDWRD
      016452 001002          BNE      50314$
      016454          50313$:
4163          ;IF NOT AT EOT OR NOT A MOTION CMD THEN:
4164 016454 004737 016540          JSR PC,VFISU          ;ISSUE CMD, CHECK STATUS AND DATA.
4165 016460          ENDIF
      016460          50314$:
4166 016460          ENDIF          50312$:
      016460          ;CHECK FOR TEST OF ON UNIT AT A TIME.
4167 016460 105737 003532          TSTB STREAM          ;BRANCH, IF STREAMING TESTS.
4168 016464 001003          BNE 2$          ;GO FIND THE NEXT UNIT.
4169 016466 004737 017346          JSR PC,NEXTU
4170 016472          ENDDO
      016472 000727          BR      50303$
      016474          50304$:
4171 016474 004737 017706          2$: JSR PC,CKHAE          ;CHECK FOR HALT AFTER EACH CMD.
4172 016500          IF DEVTBL(R5) EQ #NINUSE THEN ;IF DRIVES BEEN DROPPED EXIT
      016500 026527 002536 177774          CMP      DEVTBL(R5),#NINUSE
      016506 001005          BNE      50315$
4173 016510          LET NCNT := NCNT1 - #1
      016510 013737 003422 003420          MOV      NCNT1,NCNT
      016516 005337 003420          DEC      NCNT
4174 016522          ENDIF
      016522          50315$:
4175 016522          LET NCNT := NCNT + #1          ;UPDATE THE RECORD COUNT.
      016522 005237 003420          INC      NCNT
4176 016526          LET PCMDWD := CMDWRD          ;SAVE PREVIOUS COMMAND WORD.
      016526 013737 003426 003432          MOV      CMDWRD,PCMDWD
4177          ENDDO
4178 016534          ENDDO
      016534 000673          BR      50301$
      016536          50302$:
4179 016536 000207          RTS PC          ;RETURN.

```

```

4181      ; SUBROUTINE TO ISSUE COMMAND, AWAIT INTERRUPT,
4182      ; CHECK STATUS, CHECK DATA.
4183      ; INPUTS:
4184      ; OUTPUTS:
4185      ; REGISTERS:      R2
4186      ; CALLS:          EXECUTE, GOWAIT, CKDATA.
4187
4188 016540
4189 016540      VFISU::
016540      LET R2 := DATARD * #8.      ;INIT READ BUFFER POINTER.
016544      MOV      DATARD,R2
016544      ADD      #8.,R2
4190 016550      WHILE R2 NE DATARD DO      ;UNTIL 8 BYTES HAVE BEEN SET.
016550      50316$:
016550      CMP      R2,DATARD
016554      BEQ      50317$
4191 016556      LET -(R2) := #-1      ;INIT READ BUFFER.
016556      MOV      #-1,-(R2)
4192 016562      ENDDO
016562      BR      50316$
016564      50317$:
4193 016564      JSR PC,EXECUTE      ;GO EXECUTE THE COMMAND.
4194 016570      IFB DROPEQ EQ #0 THEN      ;IF UNIT HAS NOT BEEN DROPPED THEN:
016570      TSTB      DROPEQ
016574      BNE      50320$
4195 016576      JSR PC,GOWAIT      ;GO WAIT FOR DONE BIT.
4196 016602      ENDIF
016602
4197 016602      IFB DROPEQ EQ #0 THEN      ;IF UNIT HAS NOT BEEN DROPPED THEN:
016602      50320$:
016602      TSTB      DROPEQ
016606      BNE      50321$
4198 016610      IF #X0.BOT NOTSETIN EOTFLG(R5) THEN      ;WHEN NOT REVERSED INTO BOT, THEN
016610      BIT      #X0.BOT,EOTFLG(R5)
016616      BNE      50322$
4199 016620      IF L#TEST NE #3 THEN
016620      50323$:
016620      CMP      L#TEST,#3
016626      BEQ      50323$
4200 016630      JSR PC,CKDATA      ;GO VERIFY DATA.
4201 016634      ELSE
016634      BR      50324$
016636      50323$:
4202 016636      INC CMPDAT      ;ONE MORE XFER BEFORE A COMPARISON
4203 016642      IF CMPDAT GT DATRAT THEN
016642      50324$:
016642      CMP      CMPDAT,DATRAT
016650      BLE      50325$
4204 016652      JSR PC,CKDATA
4205 016656      LET CMPDAT := #0
016656      CLR      CMPDAT
4206 016662      ENDIF
016662      50325$:
4207 016662      ENDIF
016662      50324$:
4208 016662      ENDIF
016662      50322$:
4209 016662      ENDIF
016662      50321$:
4210 016662      RTS PC
000207
4211

```

```

4213      ; SUBROUTINE TO COMPARE DATA BETWEEN READ AND WRITE BUFFERS
4214      ; AND PRINT ERROR MESSAGE ON MISCOMPARE.
4215      ; INPUTS:
4216      ; OUTPUTS:
4217      ; REGISTERS:      R2, R3, R4.
4218      ; CALLS:          GCMDB
4219
4220      CKDATA:: LET R3 := BRFCNT - MSGPKT*MS.RFC ;COMPUTE REC LENGTH READ
          MOV      BRFCNT,R3
          SUB      MSGPKT*MS.RFC,R3
4221      IF R3 EQ #0 THEN                ;WHEN NO DATA RECEIVED
          TST      R3
          BNE      50326$
          ERRHRD 17,WTVERM,DTAERM        ;PRINT ERROR AND EXIT
          TRAP    C$ERHRD
          .WORD   17
          .WORD   WTVERM
          .WORD   DTAERM
4222      PRINTB #DTAER4                ;COMPARE ROUTINE
          MOV      #DTAER4,-(SP)
          MOV      #1,-(SP)
          MOV      SP,R0
          TRAP    C$PNTB
          ADD      #4,SP
4223      ELSE
          BR      50327$
4224      IF R3 HI BRFCNT THEN          ;WHEN REC READ IS LONGER
          CMP      R3,BRFCNT
          BLOS    50330$
          ERRHRD 17,WTVERM,DTAERM        ;THAN EXPECTED. PRINT
          TRAP    C$ERHRD
          .WORD   17
          .WORD   WTVERM
          .WORD   DTAERM
4225      PRINTB #DTAERS,CMDPKT*CP.CNT  ;AN ERROR MESSAGE
          MOV      CMDPKT*CP.CNT,-(SP)
          MOV      #DTAERS,-(SP)
          MOV      #2,-(SP)
          MOV      SP,R0
          TRAP    C$PNTB
          ADD      #6,SP
4226      ELSE                          ;AND EXIT ROUTINE
          BR      50331$
          50330$:
          LET CKDCNT := R3 - #1          ;SAVE VERIFICATION LENGTH - 1.
          MOV      R3,CKDCNT
          DEC      CKDCNT
          CLR CKDFF                      ;CLEAR # OF BYTES IN ERROR COUNTER.
          CLR R2                          ;INIT BYTE COUNTER
          LET R3 := DATAWT              ;GET WRITE BUFFER ADDRESS.
          MOV      DATAWT,R3
          LET R4 := DATARD              ;GET READ BUFFER ADDRESS.
          MOV      DATARD,R4
          IFB T1SWB NE #0 THEN          ;WHEN RUNNING TEST1-SUB 12.
          TSTB   T1SWB
          BEQ    50332$

```



```

4235 017032 000313 SWAB (R3) ;SWAP FIRST WORD OF WRT BFR
4236 017034 ENDIF ;WHICH CONTAINS THE RECORD COUNT
017034 50332$:
4237 017034 REPEAT ;REPEAT UNTIL ALL DATA IS COMPARED:
017034 50333$:
4238 017034 IF R2 EQ CKDCNT THEN ;IF THIS IS THE LAST BYTE THEN:
017034 020237 017274 CMP R2,CKDCNT
017040 001011 BNE 50334$
4239 017042 IFB SWBFLG NE #0 THEN ;IF BYTE SWAPPING IS ENABLED THEN:
017042 105737 003524 TSTB SWBFLG
017046 001406 BEQ 50335$
4240 017050 IF #BIT00 NOTSETIN CKDCNT THEN ;IF RECORD LENGTH IS ODD THEN:
017050 032737 000001 017274 BIT #BIT00,CKDCNT
017056 001002 BNE 50336$
4241 017060 TSTB (R3). ;LAST BYTE WILL BE IN
4242 017062 TSTB (R4). ;THE UPPER BYTE.
4243 017064 ENDIF 50336$:
017064 50335$:
4244 017064 ENDIF 50334$:
017064 50334$:
4246 017064 121314 CMPB (R3),(R4) ;ARE THEY EQUAL.
4247 017066 001452 BEQ 3$ ;BR IF SO.
4248 017070 005737 017276 TST CKDFF ;1 ST TIME THRU?
4249 017074 001010 BNE 2$ ;BR IF NOT.
4250 017076 005265 003300 INC VFYCNTR(R5) ;INC THE VERIFY ERROR COUNTER.
4251 017102 005265 003310 INC HRDCNTR(R5) ;INC THE HARD ERROR COUNT.
4252 017106 ERRHRD 17,WTVERM,DTAERM ;REPORT WRITE/VERIFY ERROR.
017106 104456 TRAP C$ERHRD
017110 000021 .WORD 17
017112 004246 .WORD WTVERM
017114 005460 .WORD DTAERM
4253 017116 2$: LET CKDFF := CKDFF + #1 ;INCREMENT # OF BYTES IN ERROR.
017116 005237 017276 INC CKDFF
4254 017122 111437 003444 MOVB (R4),TIME1 ;SAVE WAS DATA FOR TYP0UT.
4255 017126 042737 177400 003444 BIC #177400,TIME1 ;CLEAR GARBAGE.
4256 017134 111337 003446 MOVB (R3),TIME2 ;SAVE SHOULD BE DATA FOR TYP0UT.
4257 017140 042737 177400 003446 BIC #177400,TIME2 ;CLEAR GARBAGE.
4258 017146 IF CKDFF LT #11. THEN ;IF ERROR BYTE COUNT IS LESS THAN 11:
017146 023727 017276 000013 CMP CKDFF,#11.
017154 002017 BGE 50337$
4259 017156 PRINTX #DTAER2,R2,<B,TIME1>,<B,TIME2> ;PRINT EXP + ACT DATA.
017156 005046 CLR -(SP)
017160 153716 003446 BISB TIME2,(SP)
017164 005046 CLR -(SP)
017166 153716 003444 BISB TIME1,(SP)
017172 010246 MOV R2,-(SP)
017174 012746 004761 MOV #DTAER2,-(SP)
017200 012746 000004 MOV #4,-(SP)
017204 010600 MOV SP,R0
017206 104415 TRAP C$PNTX
017210 062706 000012 ADD #12,SP
4260 017214 ENDIF
017214 50337$:
4261 017214 105723 3$: TSTB (R3). ;UPDATE WRITE BUFFER ADDRESS.
4262 017216 105724 TSTB (R4). ;UPDATE READ BUFFER ADDRESS.

```



```

4276      ; SUBROUTINE TO FIND THE FIRST DEVICE IN THE TEST SEQUENCE.
4277      ; INPUTS:
4278      ; OUTPUTS:
4279      ; REGISTERS:
4280      ; CALLS:
4281
4282 017300 FIRSTU::LET DROPED :B= #0      ;CLR UNIT DROPPED FLAG
017300 105037 003527      ;CLR UNIT DROPPED FLAG
4283 017304      LET R5 := #0      ;CLR DEVICE POINTER.
017304 005005      ;CLR DEVICE POINTER.
4284 017306      WHILE DEVTBL(R5) EQ #NINUSE DO ;WHILE DEVICES ARE NOT IN USE:
017306      ;WHILE DEVICES ARE NOT IN USE:
017306 026527 002536 177774      50341$:
017314 001003      CMP DEVTBL(R5),#NINUSE
4285 017316      LET R5 := R5 + #2      ;POINT TO NEXT DEVICE.
017316 062705 000002      BNE 50342$
4286 017322      ENDDO      ADD #2,R5
017322 000771      BR 50341$
017324      50342$:
4287 017324      IF DEVTBL(R5) EQ #END THEN ;IF ALL UNITS HAVE BEEN DROPPED THEN:
017324 026527 002536 177777      CMP DEVTBL(R5),#END
017332 001001      BNE 50343$
4288 017334      DOCLN      ;DO CLEAN CODE AND TERMINATE PASS.
017334 104444      TRAP C$DCLN
4289 017336      ENDIF
017336
4290 017336      LET L$LUN := DEVTBL(R5) ;SET UNIT # IN "HEADER" FOR ERROR REPORT
017336 016537 002536 002074      MOV DEVTBL(R5),L$LUN
4291 017344      RTS PC ;RETURN WITH 1ST DEVICE IN R5.
4292
4293
4294
4295
4296
4297      ; SUBROUTINE TO FIND THE NEXT UNIT IN THE TEST CYCLE.
4298      ; INPUTS:
4299      ; OUTPUTS:
4300      ; REGISTERS:
4301      ; CALLS:
4302
4303 017346 NEXTU:: LET DROPED :B= #0      ;CLR UNIT DROPPED FLAG
017346 105037 003527      ;CLR UNIT DROPPED FLAG
4304 017352      BIC #177770,R5      CLRB DROPED
017352 042705 177770      REPEAT ;REPEAT UNTIL THE NEXT DEVICE IS FOUND.
4305 017356      ;REPEAT UNTIL THE NEXT DEVICE IS FOUND.
017356      LET R5 := R5 + #2      ;UPDATE DEVICE TABLE POINTER.
4306 017356 062705 000002      ;UPDATE DEVICE TABLE POINTER.
017356      UNTIL DEVTBL(R5) NE #NINUSE      ADD #2,R5
4307 017362      UNTIL DEVTBL(R5) NE #NINUSE
017362 026527 002536 177774      CMP DEVTBL(R5),#NINUSE
017370 001772      BEQ 50344$
4308 017372      LET L$LUN := DEVTBL(R5) ;SET UNIT # IN "HEADER" FOR ERROR REPORT
017372 016537 002536 002074      MOV DEVTBL(R5),L$LUN
4309 017400      RTS PC ;RETURN.
000207
4310
4311
4312

```

G10

GLOBAL AREAS MACRO M1200 21-MAR-84 16:45 PAGE 94
 RAMER - READ AND DISPLAY SELECTED RAM

SEQ 0123

```

4314      ; SUBROUTINE TO DROP A DEVICE FROM THE TEST SEQUENCE.
4315      ; INPUTS:
4316      ; OUTPUTS:
4317      ; REGISTERS:
4318      ; CALLS:          MOVMSG, PRXST, LOG
4319
4320 017402      DROPU:: LET R5SAVE := R5
017402 010537 003460
4321 017406      LET FTLCNT(R5) := FTLCNT(R5) + #1 ;INCREMENT THE FATAL ERROR COUNT.
017406 005265 003320
4322 017412      LET R4 := MSGPKT*MS.XS3 CLR.BY #377 ;GET UDIAG ERROR CODE FROM XSTAT3.
017412 013704 002354
017416 042704 000377
4323 017422      LET R3 := MSGPKA(R5) ;ADR OF THIS UNIT'S MSG PACKET.
017422 016503 002506
4324 017426      LET R2 := #0 ;CLR COUNTER.
017426 005002
4325 017430      WHILE R2 NE #MSGCNT DO ;WHILE THERE ARE MORE LOCATIONS:
017430
017430 020227 000016
017434 001405
4326 017436      LET (R3)+ := #-1 ;INIT THE MSG PACKET WITH ALL 1'S
017436 012723 177777
4327 017442      LET R2 := R2 + #2 ;UPDATE COUNTER.
017442 062702 000002
4328 017446      ENDDO
017446 000770
017450
4329 017450      LET @TSDB(R5) := #GSCPK ;INITIATE A GET STATUS COMMAND.
017450 012775 002324 002456
4330 017456      JSR PC,WSSR ;WAIT A WHILE FOR SSR=1
4331 017462      JSR PC,MOVMSG ;MOVE MSG PACKET TO COMMON AREA.
4332 017466      IF R4 EQ #X3.RNY THEN ;IF WE HAVE A CAPSTAN RUNAWAY THEN:
017466 020427 157400
017472 001005
4333 017474      ERRDF 16,RNYM,STAERM ;REPORT CAPSTAN RUNAWAY WITH TACH CNT.
017474 104455
017476 000020
017500 004566
017502 005626
4334 017504      ELSE ;ELSE-IF NOT A RUNAWAY:
017504 000402
017506
4335 017506      JSR PC,PRXST ;PRINT EXTENDED STATUS REGISTERS.
4336 017512      ENDIF
017512
4337 017512      IFB RECLOG NE #0 THEN ;IF THE RECORD HAS BEEN LOGGED THEN:
017512 105737 003471
017516 001404
4338 017520      LET DROPED :B= DROPED + #1 ;SET UNIT DROPPED FLAG.
017520 105237 003527
4339 017524      JSR PC,LOG ;LOG DATA BYTES + RD/WR ERRORS.
4340 017530      ENDIF
017530
4341 017530      DORPT ;PRINT PERFORMANCE REPORT
017530 104424
4342 017532      DROPUA: IF PASCNT(R5) NE #0 THEN

```

H10

GLOBAL AREAS MACRO M1200 21-MAR-84 16:45 PAGE 94-1
 RAMER - READ AND DISPLAY SELECTED RAM

SEQ 0124

	017532	005765	003260			TST	PASCNT(R5)
	017536	001402				BEQ	50352\$
4343	017540			LET PASCNT(R5) := PASCNT(R5) - #1			
	017540	005365	003260			DEC	PASCNT(R5)
4344	017544			ENDIF			
	017544						50352\$:
4345	017544			LET DROPN := DEVTBL(R5)	;SAVE # OF UNIT TO BE DROPPED.		
	017544	016537	002536	017622		MOV	DEVTBL(R5),DROPN
4346	017552			LET R0 := R5 SHIFT -1	;R0=LOGICAL DEVICE NUMBER		
	017552	010500				MOV	R5,R0
	017554	006200				ASR	R0
4347	017556			DODU R0	;DROP THE UNIT: EXEC BGNDU-ENDDU CODE IF IDU = 0		
	017556	104451					TRAP C#DODU
4348	017560			IF DEVTBL(R5) NE #NINUSE THEN	;IF UNIT NOT DROPPED		
	017560	026527	002536	177774		CMP	DEVTBL(R5),#NINUSE
	017566	001410				BEQ	50353\$
4349	017570			IFB IREC EQ #0 THEN	;IF RECOVERY IS ENABLED THEN:		
	017570	105737	002210			TSTB	IREC
	017574	001005				BNE	50354\$
4350	017576	000240		NOP			
4351	017600	000240		NOP			
4352	017602	000240		NOP			
4353	017604			LET STAFLG :B= STAFLG + #1	;SET START FLAG TO ENABLE REWIND.		
	017604	105237	003534			INCB	STAFLG
4354	017610			ENDIF			
	017610						50354\$:
4355	017610			ENDIF			
	017610						50353\$:
4356	017610			DRORTN: LET DROPE :B= DROPE + #1	;SET UNIT DROPPED FLAG.		
	017610	105237	003527			INCB	DROPE
4357	017614			LET R5 := R5SAVE			
	017614	013705	003460			MOV	R5SAVE,R5
4358	017620	000207		RTS PC	;RETURN.		
4359							
4360	017622	000000		DROPN: .WORD 0	;# OF UNIT TO BE DROPPED		

```

4362      ;      SUBROUTINE TO PRINT EXTENDED STATUS REGISTERS.
4363      ;      INPUTS:
4364      ;      OUTPUTS:
4365      ;      REGISTERS:
4366      ;      CALLS:
4367
4368 017624 PRXST:: PRINTX @GETSTM
      017624 012746 005241      MOV      @GETSTM,-(SP)
      017630 012746 000001      MOV      @1,-(SP)
      017634 010600      MOV      SP,R0
      017636 104415      TRAP    C$PNTX
      017640 062706 000004      ADD      @4,SP
4369      PRINTX @STAERS,MSGPKT*MS.XS0,MSGPKT*MS.XS1,MSGPKT*MS.XS2,MSGPKT*MS.XS3
      017644 013746 002354      MOV      MSGPKT*MS.XS3,-(SP)
      017650 013746 002352      MOV      MSGPKT*MS.XS2,-(SP)
      017654 013746 002350      MOV      MSGPKT*MS.XS1,-(SP)
      017660 013746 002346      MOV      MSGPKT*MS.XS0,-(SP)
      017664 012746 006453      MOV      @STAERS,-(SP)
      017670 012746 000005      MOV      @5,-(SP)
      017674 010600      MOV      SP,R0
      017676 104415      TRAP    C$PNTX
      017700 062706 000014      ADD      @14,SP
4370      RTS PC
4371
4372
4373
4374
4375      ;      SUBROUTINE TO HALT AFTER EACH COMMAND.
4376      ;      INPUTS:
4377      ;      OUTPUTS:
4378      ;      REGISTERS:      R3, R4
4379      ;      CALLS:
4380
4381 017706 CKMAE:: IFB HAE NE #0 THEN      ;IF HALT FLAG IS SET:
      017706 105737 002206      TSTB    HAE
      017712 001430      BEQ     50355$
4382      IFB MISCFG EQ #0 THEN      ;
      017714 105737 003537      TSTB    MISCFG
      017720 001023      BNE     50356$
4383      MANUAL      ;IS MANUAL INTERVENTION ALLOWED?
      017722 104450      TRAP    C$MANI
4384      BNCOMPLETE CKHRTN      ;BR IF NOT.
      017724 103023      BCC     CKHRTN
4385      LET R4 := CMDWRD      ;COMMAND WORD.
      017726 013704 003426      MOV     CMDWRD,R4
4386      JSR PC,GCMDA      ;FETCH ADR OF CMD ASCII.
      017732 004737 007654      ;MOVE CMD ASCII
      017736 112337 004124      MOVB   (R3)*,HALTM
4388      LET HALTM*1 :B= (R3)*      MOVB   (R3)*,HALTM*1
      017742 112337 004125      ;INTO MESSAGE.
4389      LET HALTM*2 :B= (R3)      MOVB   (R3),HALTM*2
4390      GMANIL HALTM,TIME1,1,YES      ;HALT - WAIT FOR AN OEPRTOR INPUT.
      017752 104443      TRAP    C$GMAN
      017754 000404      BR     10000$
      017756 003444      .WORD  TIME1
      017760 000130      .WORD  T$CODE

```



```

4397      ;      SUBROUTINE TO CREATE THE SEQUENCE FOR A WRITE TAPE MARK
4398      ;      COMMAND. WILL EXECUTE COMMAND TO UUT.
4399      ;      INPUTS:
4400      ;      OUTPUTS: CMDSEQ
4401      ;      CALLS:  SETUP, CMDAC, EXECUTE, GOWAIT
4402
4403 017776      ;
4404 017776      ;      WRITEM::
4405 017776 012701 003542      LET R1 := #CMDSEQ
4406 020002      ;      MOV      #CMDSEQ,R1
4407 020002 012721 100011      LET (R1)+ := #WTM      ;COMMAND
4408 020006      ;      MOV      #WTM,(R1)+
4409 020006 012721 000001      LET (R1)+ := #1      ;BRF
4410 020012      ;      MOV      #1,(R1)+
4411 020012 012721 000001      LET (R1)+ := #1      ;ITERATIONS
4412 020016      ;      MOV      #1,(R1)+
4413 020016 005721      TST (R1)+      ;PATTERN
4414 020020      LET (R1)+ := #END      ;TERMINATOR
4415 020020 012721 177777      ;      MOV      #END,(R1)+
4416 020024      LET R1 := #CMDSEQ      ;TOP OF BUFFER
4417 020024 012701 003542      ;      MOV      #CMDSEQ,R1
4418 020030 004737 007706      JSR PC, SETUP      ;SET UP THE TABLE
4419 020034 004737 007600      JSR PC, CMDAC      ;LOAD THE ASCII
4420 020040 004737 010654      JSR PC, EXECUTE    ;ISSUE THE WTM COMMAND
4421 020044 004737 011274      JSR PC, GOWAIT    ;WAIT FOR THE COMMAND TO FINISH
4422 020050 000207      RTS PC      ;RETURN TO CALLER
4423      .EVEN
4424      ENDMOD

```



```

4430
4431 .TITLE MISCELLANEOUS SECTIONS
4432 .SBTTL REPORT CODING SECTION
4441
4442 020052 BGNMOD
4443
4444 ;**
4445 ; THE REPORT CODING SECTION CONTAINS THE
4446 ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
4447 ;--
4448
4449 020052 BGNRPT
      020052 L$RPT::
4450
4456 020052 LET RSSAVE := R5 ;SAVE CURRENT DEVICE POINTER.
      020052 010537 003460 MOV R5,RSSAVE
4457 020056 JSR PC,FIRSTU ;FIND THE FIRST UNIT.
      004737 017300 WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
4458 020062 50360$:
      020062 026527 002536 177777 CMP DEVTBL(R5),#END
      020070 001562 BEQ 50361$
4459 020072 PRINTS #RPT1A,DEVTBL(R5),PASCNT(R5),RECCNT(R5)
      020072 016546 003330 MOV RECCNT(R5),-(SP)
      020076 016546 003260 MOV PASCNT(R5),-(SP)
      020102 016546 002536 MOV DEVTBL(R5),-(SP)
      020106 012746 020714 MOV #RPT1A, -(SP)
      020112 012746 000004 MOV #4, -(SP)
      020116 010600 MOV SP,R0
      020120 104416 TRAP C$PNTS
      020122 062706 000012 ADD #12,SP
4460 020126 PRINTS #RPT1B,WRBC+30(R5),WRBC+20(R5),WRBC+10(R5),WRBC(R5)
      020126 016546 002560 MOV WRBC(R5),-(SP)
      020132 016546 002570 MOV WRBC+10(R5),-(SP)
      020136 016546 002600 MOV WRBC+20(R5),-(SP)
      020142 016546 002610 MOV WRBC+30(R5),-(SP)
      020146 012746 020771 MOV #RPT1B, -(SP)
      020152 012746 000005 MOV #5, -(SP)
      020156 010600 MOV SP,R0
      020160 104416 TRAP C$PNTS
      020162 062706 000014 ADD #14,SP
4461 020166 PRINTS #RPT1C,RRBC+30(R5),RRBC+20(R5),RRBC+10(R5),RRBC(R5)
      020166 016546 002620 MOV RRBC(R5),-(SP)
      020172 016546 002630 MOV RRBC+10(R5),-(SP)
      020176 016546 002640 MOV RRBC+20(R5),-(SP)
      020202 016546 002650 MOV RRBC+30(R5),-(SP)
      020206 012746 021042 MOV #RPT1C, -(SP)
      020212 012746 000005 MOV #5, -(SP)
      020216 010600 MOV SP,R0
      020220 104416 TRAP C$PNTS
      020222 062706 000014 ADD #14,SP
4462 020226 PRINTS #RPT1D,RFBC+30(R5),RFBC+20(R5),RFBC+10(R5),RFBC(R5)
      020226 016546 002660 MOV RFBC(R5),-(SP)
      020232 016546 002670 MOV RFBC+10(R5),-(SP)
      020236 016546 002700 MOV RFBC+20(R5),-(SP)
      020242 016546 002710 MOV RFBC+30(R5),-(SP)
      020246 012746 021113 MOV #RPT1D, -(SP)
      020252 012746 000005 MOV #5, -(SP)

```

```

020256 010600
020260 104416
020262 062706 000014
4463 020266 PRINTS #RPT1F,WRREC(R5),RRREC(R5),RFREC(R5)
020266 016546 002760
020272 016546 002740
020276 016546 002720
020302 012746 021217
020306 012746 000004
020312 010600
020314 104416
020316 062706 000012
4464 020322 PRINTS #RPT1G,WRUNR(R5),RRUNR(R5),RFUNR(R5)
020322 016546 002770
020326 016546 002750
020332 016546 002730
020336 012746 021270
020342 012746 000004
020346 010600
020350 104416
020352 062706 000012
4465 020356 IFB BADTSW NE #0 THEN ;
020356 105737 002211
020362 001402
4466 020364 004737 020446 JSR PC,BTRPT ;GO PRINT BAD TAPE SPOTS WHEN ENABLED
4467 020370 ENDIF
020370
4468 020370 PRINTS #RPT1I,SCCNT(R5),HRDCNT(R5),FTLCNT(R5),VFYCNT(R5)
020370 016546 003300
020374 016546 003320
020400 016546 003310
020404 016546 003270
020410 012746 021460
020414 012746 000005
020420 010600
020422 104416
020424 062706 000014
4469 020430 004737 017346 JSR PC,NEXTU ;FIND THE NEXT UNIT.
4470 020434 ENDDO
020434 000612
020436
4471 020436 LET R5 := R5SAVE ;RESTORE CURRENT DEVICE POINTER.
020436 013705 003460
4472 020442 EXIT RPT
020442 000167
020444 001206
4473
4474 ; SUBR TO PRINT BAD TAPES SPOTS DURING THE REPORT PRINTS
4475 ; WRITE RETRIES: CUMULATIVE COUNT
4476 ; BAD TAPE SPOTS: COUNT PER TAPE PASS ONLY, NOT CUMULATIVE.
4477 ; COUNT OF RECOVERABLE WRITE ERRORS EXCLUDES BAD TAPE SPOTS.
4478
4479 020446 BTRPT: PRINTS #RPT1E,WRTYCT(R5) ;PRINT GLOBAL WRITE RETRY COUNT
020446 016546 003250
020452 012746 021341
020456 012746 000002
020462 010600
    
```

```

MOV SP,RO
TRAP C$PNTS
ADD #14,SP
MOV RFREC(R5),-(SP)
MOV RRREC(R5),-(SP)
MOV WRREC(R5),-(SP)
MOV #RPT1F,-(SP)
MOV #4,-(SP)
MOV SP,RO
TRAP C$PNTS
ADD #12,SP
MOV RFUNR(R5),-(SP)
MOV RRUNR(R5),-(SP)
MOV WRUNR(R5),-(SP)
MOV #RPT1G,-(SP)
MOV #4,-(SP)
MOV SP,RO
TRAP C$PNTS
ADD #12,SP
TSTB BADTSW
BEQ 50362$
50362$:
MOV VFYCNT(R5),-(SP)
MOV FTLCNT(R5),-(SP)
MOV HRDCNT(R5),-(SP)
MOV SCCNT(R5),-(SP)
MOV #RPT1I,-(SP)
MOV #5,-(SP)
MOV SP,RO
TRAP C$PNTS
ADD #14,SP
BR 50360$
50361$:
MOV POINTER,
R5SAVE,R5
.WORD J$JMP
.WORD L10010-2-.
    
```

020464	104416						TRAP	C:PNTS
020466	062706	000006					ADD	#6,SP
4480	020472			LET BTPT := BTADDR(R5)	;BTPT IS BOTH THE BAD TAPE SPOT		COUNTER	
	020472	016537	002550				MOV	BTADDR(R5),BTPT
4481	020500		003516	LET R3 := @BTPT SHIFT -1	;AND THE LOGGING INDEX		MOV	@BTPT,R3
	020500	017703	163012				MOV	
	020504	006203					ASR	R3
4482	020506			PRINTS #RPT1J,R3	;PRINT # OF BAD TAPE SPOTS			
	020506	010346					MOV	R3,-(SP)
	020510	012746	021371				MOV	#RPT1J,-(SP)
	020514	012746	000002				MOV	#2,-(SP)
	020520	010600					MOV	SP,RO
	020522	104416					TRAP	C:PNTS
	020524	062706	000006				ADD	#6,SP
4483	020530			IF R3 NE #0 THEN	;PRINT RECORD # IF BAD SPOTS DETECTED			
	020530	005703					TST	R3
	020532	001457					BEQ	50363\$
4484	020534			IF R3 HI #20. THEN				
	020534	020327	000024				CMP	R3,#20.
	020540	101402					BLOS	50364\$
4485	020542			LET R3 := #20.	;20 BAD SPOTS IS THE LIMIT			
	020542	012703	000024				MOV	#20.,R3
4486	020546			ENDIF				
	020546							50364\$:
4487	020546			PRINTS #CRLFSP				
	020546	012746	005300				MOV	#CRLFSP,-(SP)
	020552	012746	000001				MOV	#1,-(SP)
	020556	010600					MOV	SP,RO
	020560	104416					TRAP	C:PNTS
	020562	062706	000004				ADD	#4,SP
4488	020566			LET R4 := BTPT + #2	;FETCH A BAD SPOT ID			
	020566	013704	003516				MOV	BTPT,R4
	020572	062704	000002				ADD	#2,R4
4489	020576			LET R2 := #0	;R2 = PRINT COUNT PER LINE: 10 MAX			
	020576	005002					CLR	R2
4490	020600			REPEAT				
	020600							50365\$:
4491	020600			PRINTS #RPT1K,(R4)	;PRINT A BAD SPOT ID			
	020600	011446					MOV	(R4),-(SP)
	020602	012746	021451				MOV	#RPT1K,-(SP)
	020606	012746	000002				MOV	#2,-(SP)
	020612	010600					MOV	SP,RO
	020614	104416					TRAP	C:PNTS
	020616	062706	000006				ADD	#6,SP
4492	020622			LET R2 := R2 + #1	;COUNT PRINTS			
	020622	005202					INC	R2
4493	020624			LET R4 := R4 + #2	;NEXT			
	020624	062704	000002				ADD	#2,R4
4494	020630			IF R2 EQ #10. THEN				
	020630	020227	000012				CMP	R2,#10.
	020634	001014					BNE	50366\$
4495	020636			PRINTS #CRLFSP	;GO TO NEXT PRINT LINE PAST 10 PRINTS			
	020636	012746	005300				MOV	#CRLFSP,-(SP)
	020642	012746	000001				MOV	#1,-(SP)
	020646	010600					MOV	SP,RO
	020650	104416					TRAP	C:PNTS
	020652	062706	000004				ADD	#4,SP

```

4496 020656          LET R3 := R3 - #10.          ;ADJUST BAD SPOT COUNT
      020656 162703 000012          SUB          #10.,R3
4497 020662          LET R2 := R2 - #10.          ;ADJUST PRINT COUNT
      020662 162702 000012          SUB          #10.,R2
4498 020666          ENDIF
      020666
4499 020666          UNTIL R2 EQ R3              ;LIMIT: # OF BAD SPOTS
      020666 020203          50366#
      020670 001343          CMP          R2,R3
4500 020672          ENDIF
      020672
4501 020672          PRINTS #CRLF
      020672 012746 005275          MOV          #CRLF, -(SP)
      020676 012746 000001          MOV          #1, -(SP)
      020702 010600          MOV          SP,RO
      020704 104416          TRAP         C#PNTS
      020706 062706 000004          ADD          #4,SP
4502 020712          RTS PC
4503
4515
4516 020714          045          116          045 RPT1A: .ASCIZ /#N#N#AUNIT #D1#S3#APASS:#D5#S3#ARECORD:#D5#N/
4517 020771          045          101          102 RPT1B: .ASCIZ /#ABYTES WRITTEN #D3#A,#Z3#A,#Z3#A,#Z3#N/
4518 021042          045          101          102 RPT1C: .ASCIZ /#ABYTES READ REV #D3#A,#Z3#A,#Z3#A,#Z3#N/
4519 021113          045          101          102 RPT1D: .ASCII /#ABYTES READ FWD #D3#A,#Z3#A,#Z3#A,#Z3#N/
4520 021163          045          123          062 .ASCIZ /#S23#AWRT#S4#ARDR#S4#ARDF#N/
4521 021217          045          101          122 RPT1F: .ASCIZ /#ARECOVERABLE ERRORS #D5#S2#D5#S2#D5#N/
4522 021270          045          101          125 RPT1G: .ASCIZ /#AUNRECOVERABLE ERRORS #D5#S2#D5#S2#D5#N/
4523 021341          045          101          127 RPT1E: .ASCIZ /#AWRITE RETRIES#S8#D5#N/
4524 021371          045          116          045 RPT1J: .ASCIZ /#N#D2#A BAD SPOTS THIS PASS PRECEDING RECORD #:/
4525 021451          045          104          065 RPT1K: .ASCIZ /#D5#S1/
4526 021460          045          101          123 RPT1I: .ASCII "#ASPEC COND#S3#AHARD#S3#AFATAL#S3#ACOMPARE#N"
4527 021534          045          123          063 .ASCIZ /#S3#D5#S3#D5#S3#D5#S3#D5#N#N/
4528 021571          045          116          045 TAPCAP: .ASCIZ /#N#ATAPE IN CARTRIDGE MUST BE 600' IN LENGTH.#N#N/
4529
4530
4531 021654          021654          ENDRPT
      021654 104425          L10010:
4532
4533
4534
4535
4536
4537 021656          BGNPROT
      021656
4538 021656          000000          L#PROT::
4539 021660          177777          .WORD 0          ;P-TBL OFFSET OF TSSR, THE TK25 CSR
4540 021662          177777          .WORD -1         ;P-TBL OFFSET OF MASS BUS UNIT #: -1 = NOT A MASS BUS DEV
4541 021664          .WORD -1         ;P-TBL OFFSET OF DRIVE #: -1 = NONE, ONE DRIVE PER UNIBUS AD
      ENDRPT
    
```

```

4543 .SBTTL INITIALIZE SECTION
4544
4545
4546 ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
4547 ; AT THE BEGINNING OF EACH PASS.
4548 ;--
4549
4550 021664 BGNINIT
021664 L$INIT::
4551
4561 021664 INIT10: IF @BIT0!BIT1 SETIN @CMDPKT THEN ;IF CMD PACKET IS NOT ON MODULO 4 BOUNDARY:
021664 032727 000003 002314 BIT @BIT0!BIT1,@CMDPKT
021672 001421 BEQ 50367$
4562 021674 ERRSF 1,CMDPKM ;PRINT ERROR MSG, TRAP C$ERSF
021674 104454 .WORD 1
021676 000001 .WORD CMDPKM
021700 004164 .WORD 0
021702 000000 .WORD 0
4563 021704 DELAY 20. ;GO TO SUPERVISOR, WAIT 2 SECONDS.
021704 012727 000024 MOV @20.,(PC)+
021710 000000 .WORD 0
021712 013727 002116 MOV L$DLY,(PC)+
021716 000000 .WORD 0
021720 005367 177772 DEC -6(PC)
021724 001375 BNE -.4
021726 005367 177756 DEC -22(PC)
021732 001367 BNE -.20
4564 021734 BR INIT10
4565 021736 ENDIF
021736
4566 50367$:
4567 021736 IFB CLRFLG NE #0 THEN ;IF CLR COUNTERS FLAG SET:
021736 105737 002204 TSTB CLRFLG
021742 001413 BEQ 50370$
4568 021744 CLRB CLRFLG ;INIT CLR FLAG.
4569 021750 LET R2 := #0
021750 005002 CLR R2
4570 021752 WHILE R2 NE @CNTLEN DO
021752 020227 000550 50371$:
021756 001405 CMP R2,@CNTLEN
4571 021760 LET WRBC(R2) := #0 ;CLR ALL STATISTICAL COUNTERS.
021760 005062 002560 BEQ 50372$
4572 021764 LET R2 := R2 + #2 CLR WRBC(R2)
021764 062702 000002 ADD #2,R2
4573 021770 ENDDO BR 50371$
021770 000770 50372$:
4574 021772 ENDIF 50370$:
021772
4575
4576 021772 IFB RRANV NE #0 THEN ;IF RESET RANDOM VARIABLE FLAG IS SET THEN:
021772 105737 002205 TSTB RRANV
021776 001406 BEQ 50373$
4577 022000 LET RANB := @RANBC ;RESET RANDOM BASE @.
022000 012737 153624 003440 MOV @RANBC,RANB
4578 022006 LET RANS := @RANSC ;RESET RANDOM SAVE LOCATION.

```

4579	022006	012737	032561	003442	ENDIF	MOV	#RANSC,RANS
4580	022014				READEF #EF.START	50373\$:	
	022014					;READ START COMMAND EVENT FLAG.	
	022014	012700	000040			MOV	#EF.START,RO
	022020	104447				TRAP	C\$REFG
4581	022022				BNCOMPLETE INIT15	;BRANCH IF NOT STARTING.	
	022022	103030				BCC	INIT15
4582	022024				LET STAF LG :B= STAF LG * #1	;SET START COMMAND FLAG.	
	022024	105237	003534			INCB	STAF LG
4583	022030				LET HERE :B= #0	;RESET THE TEST 3 ASCII SEMAPHORE	
	022030	105037	003340			CLRB	HERE
4584	022034				LET R5 := #6		
	022034	012705	000006			MOV	#6,R5
4585	022040				REPEAT	;INITIATE UNIT NUMBER TABLE	
	022040					50374\$:	
4586	022040				LET DEVTBL(R5) := #NINUSE	;BY STORING NOT IN USE IN EACH LOCATION.	
	022040	012765	177774	002536		MOV	#NINUSE,DEVTBL(R5)
4587	022046				LET R5 := R5 - #2		
	022046	162705	000002			SUB	#2,R5
4588	022052				UNTIL R5 EQ #0		
	022052	005705				TST	R5
	022054	001371				BNE	50374\$
4589	022056				LET R5 := L\$UNIT SHIFT 1		
	022056	013705	002012			MOV	L\$UNIT,R5
	022062	006305				ASL	R5
4590	022064				REPEAT	;STORE ALL UNIT	
	022064					50375\$:	
4591	022064				LET R5 := R5 - #2	;NUMBERS IN DEVTBL.	
	022064	162705	000002			SUB	#2,R5
4592	022070				LET DEVTBL(R5) := R5 SHIFT -1		
	022070	010565	002536			MOV	R5,DEVTBL(R5)
	022074	006265	002536			ASR	DEVTBL(R5)
4593	022100				UNTIL R5 EQ #0		
	022100	005705				TST	R5
	022102	001370				BNE	50375\$
4594							
4595	022104				INIT15: READEF #EF.PWR	;HAS THERE BE A POWER FAILURE?	
	022104	012700	000034			MOV	#EF.PWR,RO
	022110	104447				TRAP	C\$REFG
4596	022112				BNCOMPLETE INIT16	;BRANCH IF NOT.	
	022112	103004				BCC	INIT16
4597	022114				LET STAF LG :B= STAF LG * #1	;IF SO - SET THE START FLAG.	
	022114	105237	003534			INCB	STAF LG
4598	022120				LET PWRFLG :B= PWRFLG * #1	;IF SO - SET THE POWER FAIL FLAG.	
	022120	105237	003535			INCB	PWRFLG
4599							
4600	022124				INIT16: RFLAGS OPFLAG	;READ AND STORE FLAGS SET BY OPERATOR	
	022124	104421				TRAP	C\$RFLA
	022126	010037	003540			MOV	RO,OPFLAG
4601	022132				LET R3 := #0	;CLEAR EVENT FLAG	
	022132	005003				CLR	R3
4602	022134				IFB PWRFLG EQ #0 THEN	;IF POWER FAIL HAS NOT OCCURRED THEN:	
	022134	105737	003535			TSTB	PWRFLG
	022140	001020				BNE	50376\$
4603	022142				READEF #EF.NEW	;UPDATE PASS COUNT WHEN	
	022142	012700	000035			MOV	#EF.NEW,RO

```

4604 022146 104447          IFCOND C3 THEN          ;SUPERVISOR IS IN NEW PASS TRAP C$REFG
022150 103014          ;AND DIAG WAS NEITHER STARTED BCC 50377$
4605 022152 105737 003534  IFB STAF LG EQ #0 THEN ;TSTB STAF LG
022156 001010          ;NOR BNE 50400$
4606 022160 012700 000037  READEF #EF.RES          ;MOV #EF.RES,RO
022164 104447          ;TRAP C$REFG
4607 022166 103402          IFCOND CC THEN          ;RESTARTED
022170 005103          LET R3 := COMP R3      ;DO IT BCS 50401$
4609 022172 000401          ELSE          ;COM R3
022174          ;BR 50402$
4610 022174 005203          LET R3 := R3 + #1      ;SET 1ST PASS IF NEW PASS AND
022176          ;RESTARTING INC R3
4611 022176          ELSE          ;50402$:
4612 022176 000401          ;BR 50403$
022200          ;50400$:
4613 022200 005203          LET R3 := R3 + #1      ;SET 1ST PASS IF NEW PASS AND
022202          ;STARTING INC R3
4614 022202          ;50403$:
4615 022202          ;DO NOT UPDATE IT ON CONTINUE 50377$:
4616 022202          ;OR ON POWER FAIL 50376$:
022202          ;INIT DEVICE POINTER.
4617 022202 004737 017300  JSR PC,FIRSTU          ;INIT DEVICE COUNTER.
4618 022206 005002          LET R2 := #0          ;CLR R2
4619 022210          WHILE DEVTBL(R5) NE #END DO
022210          ;50404$:
022210 026527 002536 177777  CMP DEVTBL(R5),#END
022216 001450          ;BEQ 50405$
4620 022220          LET R2 := R2 + #1          ;INC R2
4621 022222          LET R0 := R5 SHIFT -1          ;MOV R5,RO
022222 010500          ;ASR RO
022224 006200          ;GET HARDWARE P TABLE FROM SUPER.
4622 022226          GPHARD R0,R0          ;TRAP C$GPHRD
4623 022230          IFCOND C5 THEN          ;BCC 50406$
022230 103036          LET TSSR(R5) := (R0)          ;SAVE TSSR ADDRESS.
4624 022232 011065 002466          LET TSDB(R5) := (R0), - #2 ;SAVE TSDB ADDRESS.
022236 012065 002456          ;MOV (R0),TSSR(R5)
022242 162765 000002 002456  ;MOV (R0),TSDB(R5)
4626 022250          LET TSVCT(R5) := (R0)          ;SUB #2,TSDB(R5)
022250 011065 002476          ;SAVE INTERRUPT VECTOR ADDRESS.
4627 022254          SETVEC TSVCT(R5),TS4INT(R5),#INTPRI ;SET UP INTERUPT PROCESSING CONDITIONS.
022254 012746 000340          ;MOV #INTPRI,-(SP)
022260 016546 002516          ;MOV TS4INT(R5),-(SP)

```

022264	016546	002476							
022270	012746	000003							MOV TSVCT(R5), -(SP)
022274	104437								MOV #3, -(SP)
4628 022276	062706	000010							TRAP C#SVEC
022302			IF R3 NE #0 THEN						ADD #10, SP
022302	005703								;ACTUAL PASSCOUNT UPDATE PER R3
022304	001410								TST R3
4629 022306			IF R3 LT #0 THEN						BEQ 50407\$
022306	005703								TST R3
022310	002003								BGE 50410\$
4630 022312			LET PASCNT(R5) := PASCNT(R5) * #1						INC PASCNT(R5)
022312	005265	003260							BR 50411\$
4631 022316			ELSE						50410\$:
022316	000403								MOV #1, PASCNT(R5)
022320			LET PASCNT(R5) := #1						50411\$:
4632 022320			ENDIF						50407\$:
022320	012765	000001	ENDIF						50406\$:
4633 022326			ENDIF						;CLEAR RECORD COUNT
022326			LET RECCNT(R5) := #0						CLR RECCNT(R5)
4634 022326			JSR PC, NEXTU						;DO IT FOR ALL DEVICES.
4635 022326			ENDDO						BR 50404\$
022326									50405\$:
4636 022326									
4637 022332	005065	003330	IF R2 EQ #0 THEN						;IF THERE ARE NO UNITS:
4638 022336	004737	017346							TST R2
022336	000724								BNE 50412\$
4639 022340			PRINTF #AUDRPM						;PRINT ALL UNITS DROPPED,
4640 022340	005702								MOV #AUDRPM, -(SP)
022340	001026								MOV #1, -(SP)
4641 022344									MOV SP, R0
022344	012746	004727							TRAP C#PNTF
022350	012746	000001							ADD #4, SP
022354	010600								;GO TO SUPERVISOR, WAIT 2 SECONDS.
022356	104417								MOV #20, (PC)
4642 022360	062706	000004	DELAY 20.						.WORD 0
022364	012727	000024							MOV L#DLY, (PC)
022370	000000								.WORD 0
022372	013727	002116							DEC -6(PC)
022376	000000								BNE -4
022400	005367	177772							DEC -22(PC)
022404	001375								BNE -20
022406	005367	177756							;GO TO SUPERVISOR, CHECK TTY.
022412	001367		BREAK						TRAP C#BRK
4643 022414			DOCLN						;DO CLEAN CODE * ABORT PASS.
022414	104422								TRAP C#DCLN
4644 022416			ENDIF						
022416	104444								50412\$:
4645 022420									;LOWER CPU PRIORITY TO 0
022420									MOV #PRI00, R0
4646 022420			SETPRI #PRI00						
4647 022420	012700	000000							


```

4648 022424 104441
022426 105737 002210
022432 001145
022434 032737 000020 003540
022442 001141
4649 022444 004737 017300
4650 022450
022450 026527 002536 177777
022456 001533
4651 022460
4652 022460 012737 000001 003444
022466 000402
022470
022470 005237 003444
022474
022474 023727 003444 000025
022502 003106
4653 022504
022504 012775 002324 002456
4654 022512
022512 012727 000372
022516 000000
022520 013727 002116
022524 000000
022526 005367 177772
022532 001375
022534 005367 177756
022540 001367
4655 022542
022542 032775 000200 002466
022550 001420
4656 022552
022552 032775 000100 002466
022560 001001
4657 022562
022562 000456
4658 022564
022564
4659 022564
022564 016546 002536
022570 012746 005211
022574 012746 000002
022600 010600
022602 104417
022604 062706 000006
4660 022610
022610
4661 022610
022610 000412
022612
4662 022612
022612 016546 002536
022616 012746 023544
022622 012746 000002

```

```

IFB IREC EQ #0 AND #ADR NOTSETIN OPFLAG THEN
    JSR PC,FIRSTU ;AND AUTO-DROP NOT CALLED, THEN SET UP FOR FIRST UNI
    WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
        BEGIN COUNTER ;START 3.5 MINUTE COUNTER
        INCR TIME1 FROM #1 TO #25 BY #1
        LET @TSDB(R5) := #GSCP# ;AND GET UNITS STATUS
        DELAY 250. ;WAIT 25 MSEC.
        IF #TS.SSR SETIN @TSSR(R5) THEN
            IF #TS.OFL NOTSETIN @TSSR(R5) THEN
                LEAVE COUNTER ;EXIT COUNTER WHEN UNIT ON LINE
            ELSE
                PRINTF #OFLIN#,DEVTBL(R5) ;PRINT UNIT OFF LINE EVERY 10 SEC
            ENDIF
        ELSE
            PRINTF #NRDYM#,DEVTBL(R5)

```

```

TRAP C$SPRI
;IF ERROR RECOVERY IS ENABLED
TSTB IREC
BNE 50413$
BIT #ADR,OPFLAG
BNE 50413$
50414$:
CMP DEVTBL(R5),#END
BEQ 50415$
MOV #1,TIME1
BR 50417$
50420$:
INC TIME1
50417$:
CMP TIME1,#25
BGT 50421$
MOV #GSCP#,@TSDB(R5)
MOV #250.,(PC)+
.WORD 0
MOV L$DLY,(PC)+
.WORD 0
DEC -6(PC)
BNE -4
DEC -22(PC)
BNE -20
BIT #TS.SSR,@TSSR(R5)
BEQ 50422$
BIT #TS.OFL,@TSSR(R5)
BNE 50423$
BR 50416$
50423$:
MOV DEVTBL(R5),-(SP)
MOV #OFLIN#,-(SP)
MOV #2, -(SP)
MOV SP,RO
TRAP C$PNTF
ADD #6,SP
50424$:
BR 50425$
50422$:
MOV DEVTBL(R5),-(SP)
MOV #NRDYM, -(SP)
MOV #2, -(SP)

```

022626	010600						MOV	SP,RO
022630	104417						TRAP	C\$PNTF
022632	062706	000006					ADD	#6,SP
4663	022636				ENDIF			
022636								
4664	022636				INCR TIME2 FROM #1 TO #13 BY #1	50425\$:		
022636	012737	000001	003446				MOV	#1,TIME2
022644	000402						BR	50426\$
022646							50427\$:	
022646	005237	003446					INC	TIME2
022652							50426\$:	
022652	023727	003446	000013				CMP	TIME2,#13
022660	003016						BGT	50430\$
4665	022662				DELAY 200. ;WAIT FOR UNIT TO BE SET ON-LINE			
022662	012727	000310					MOV	#200.,(PC)+
022666	000000						.WORD	0
022670	013727	002116					MOV	L\$DLY,(PC)+
022674	000000						.WORD	0
022676	005367	177772					DEC	-6(PC)
022702	001375						BNE	.-4
022704	005367	177756					DEC	-22(PC)
022710	001367						BNE	.-20
4666	022712				BREAK ;ALLOW TERMINAL INTERRUPT			
022712	104422						TRAP	C\$BRK
4667	022714				ENDINC			
022714	000754						BR	50427\$
022716							50430\$:	
4668	022716				ENDINC			
022716	000664						BR	50420\$
022720							50421\$:	
4669	022720				END COUNTER			
022720							50416\$:	
4670	022720				IF TIME1 GT #25 THEN ;IF OFF LINE FOR 3.5 MINUTES			
022720	023727	003444	000025				CMP	TIME1,#25
022726	003404						BLE	50431\$
4671	022730	004737	011726		JSR PC,MOVMSG ;GET MESSAGE PACKET			
4672	022734	004737	012762		JSR PC,TCC1 ;PRINT ERROR AND DROP OFF LINE UNIT			
4673	022740				ENDIF			
022740							50431\$:	
4674								
4675	022740	004737	017346		JSR PC,NEXTU ;REPEAT UNTIL ON LINE OR TIMED OUT.			
4676	022744				ENDDO ;SET UP FOR NEXT UNIT.			
022744	000641						BR	50414\$
022746							50415\$:	
4677	022746				ENDIF			
022746							50413\$:	
4678	022746				IFB PWRFLG EQ #0 THEN			
022746	105737	003535					TSTB	PWRFLG
022752	001026						BNE	50432\$
4679	022754				MEMORY DATAWT ;REQUEST MEMORY FROM SUPER FOR RD/WR BUFFERS.			
022754	104431						TRAP	C\$MEM
022756	010037	003414					MOV	RO,DATAWT
4680	022762				LET DATARD := DATAWT * #DATCNT ;SET RD BFR AD			
022762	013737	003414	003416				MOV	DATAWT,DATARD
022770	062737	010000	003416				ADD	#DATCNT,DATARD
4681	022776				IF #DATAWT LT #DATCNT THEN ;WHEN NOT ENOUGH FREE MEMO AVAILABLE			
022776	027727	160412	010000				CMP	#DATAWT,#DATCNT

```

4682 023004 002011
023006 PRINTF @MEMOM ;WARN OPERATOR BGE 50433$
023006 012746 023054 MOV @MEMOM,-(SP)
023012 012746 000001 MOV @1,-(SP)
023016 010600 MOV SP,RO
023020 104417 TRAP C$PNTF
023022 062706 000004 ADD @4,SP
4683 023026 DOCLN ;AND ABORT PASS TRAP C$DCLN
023026 104444 ;DIAG MUST BE RE-LOADED IN A CPU WITH LARGER MEMO
4684 023030 ENDIF 50433$:
023030 ENDIF 50432$:
4685 023030
023030
4686
4687 023030 LET CHGFLG :B= #0 ;CLR CHANGE CMD SEQ TBL FLAG.
023030 105037 002217 CLR CLRB CHGFLG
4688 023034 LET R3 := #ENDFLG MOV @ENDFLG,R3
023034 012703 003534 ;CLEAR ALL FLAGS.
4689 023040 004737 011656 JSR PC,CLRERR ;CLEAR THE POWER FAIL FLAG.
4690 023044 LET PWRFLG :B= #0 CLR CLRB PWRFLG
023044 105037 003535
4691
4692 023050 EXIT INIT TRAP C$EXIT
023050 104432 .WORD L10012-.
023052 000104
4704 023054 045 101 106 MEMOM: .ASCII /#AFREE MEMO TOO SMALL FOR RD-WR BFRS#N/
023057 122 105 105
023062 040 115 105
023065 115 117 040
023070 124 117 117
023073 040 123 115
023076 101 114 114
023101 040 106 117
023104 122 040 122
023107 104 055 127
023112 122 040 102
023115 106 122 123
023120 045 116
4705 023122 045 101 122 .ASCIZ /#ARE-LOAD IN LARGER MEMO#N/
023125 105 055 114
023130 117 101 104
023133 040 111 116
023136 040 114 101
023141 122 107 105
023144 122 040 115
023147 105 115 117
023152 045 116 000
4706 .EVEN
4707
4708 023156 ENDINIT
023156 L10012:
023156 104411 TRAP C$INIT

```

```

4710      .SBTTL  AUTO DROP SECTION
4711
4712      ;**
4713      ;SECTION EXECUTED AFTER THE INIT CODE WHEN "ADR" FLAG IS SET BY OPERATOR
4714      ;SECTION CHECKS FOR A VALID INTERFACE LOCATION.  DROPS UNIT IF NO RESPONSE
4715      ;FROM INTERFACE
4716      ;--
4717
4718      023160      BGNAUTO
4719      023160      L$AUTO::
4720      023160      004737  017300      JSR PC,FIRSTU      ;FIND FIRST UNIT
4721      023164      WHILE DEVTBL(R5) NE #END DO      ;
4722      023164      026527  002536  177777      50434$:
4723      023172      001525      CMP      DEVTBL(R5),#END
4724      023174      LET TRAPD4 :B= #0      ;
4725      023174      105037  003536      BEQ      50435$
4726      023200      SETVEC #4,#TRAP4,#PRI07      ;SET VECTOR 4
4727      023200      012746  000340      CLRB      TRAPD4
4728      023204      012746  023574      MOV      #PRI07,-(SP)
4729      023210      012746  000004      MOV      #TRAP4,-(SP)
4730      023214      012746  000003      MOV      #4,-(SP)
4731      023220      104437      MOV      #3,-(SP)
4732      023222      062706  000010      TRAP    C$SVEC
4733      023226      LET R2 := #TSSR(R5)      ;ADDRESS TK25 INTERFACE
4734      023226      017502  002466      ADD     #10,SP
4735      023232      CLRVEC #4      ;CLEAR VECTOR AT 4
4736      023232      012700  000004      MOV     #4,R0
4737      023236      104436      TRAP   C$CVEC
4738      023240      IFB TRAPD4 NE #0 THEN
4739      023240      105737  003536      TSTB   TRAPD4
4740      023244      001423      BEQ    50436$
4741      023246      LET FTLCNT(R5) := FTLCNT(R5) + #1
4742      023246      005265  003320      INC    FTLCNT(R5)
4743      023252      PRINTF #AUTODM,TSSR(R5)      ;PRINT ERROR
4744      023252      016546  002466      MOV    TSSR(R5),-(SP)
4745      023256      012746  023450      MOV    #AUTODM,-(SP)
4746      023262      012746  000002      MOV    #2,-(SP)
4747      023266      010600      MOV    SP,R0
4748      023270      104417      TRAP  C$PNTF
4749      023272      062706  000006      ADD    #6,SP
4750      023276      LET DROPN := DEVTBL(R5)      ;SAVE # OF UNIT TO BE DROPPED.
4751      023276      016537  002536  017622      MOV    DEVTBL(R5),DROPN
4752      023304      LET R0 := R5 SHIFT -1      ;R0=LOGICAL DEVICE NUMBER
4753      023304      010500      MOV    R5,R0
4754      023306      006200      ASR   R0
4755      023310      DODU R0      ;DROP THE UNIT: EXEC BGNDU-ENDDU CODE IF IDU = 0
4756      023310      104451      TRAP  C$DODU
4757      023312      ELSE
4758      023312      000452      BR    50437$
4759      023314      50436$:
4760      023314      LET #TSDB(R5) := #GSCPCK      ;SEND GET STATUS COMMAND
4761      023314      012775  002324  002456      MOV    #GSCPCK,#TSDB(R5)
4762      023322      004737  011672      JSR PC,WSSR      ;WAIT
4763      023326      IF #TS.SSR SETIN #TSSR(R5) THEN
4764      023326      032775  000200  002466      BIT    #TS.SSR,#TSSR(R5)
    
```

```

4736 023334 001423
023336 032775 000100 002466
023344 001416
4737 023346
023346 005265 003320
4738 023352
023352 016546 002536
023356 012746 005211
023362 012746 000002
023366 010600
023370 104417
023372 062706 000006
4739 023376 004737 017532
023402
023402
4741 023402
023402 000416
023404
4742 023404
023404 005265 003320
4743 023410
023410 016546 002536
023414 012746 023544
023420 012746 000002
023424 010600
023426 104417
023430 062706 000006
4744 023434 004737 017532
023440
023440
4746 023440
023440
4747 023440 004737 017346
4748 023444
023444 000647
023446
4749
4750 023446
023446
023446 104461
4751
4752 023450 045 101 102
023453 125 123 040
023456 124 122 101
023461 120 040 101
023464 124 040 045
023467 117 066 045
023472 116
4753 023473 045 101 111
023476 116 124 105
023501 122 106 101
023504 103 105 040
023507 102 101 104
023512 040 117 122
023515 040 116 117
023520 124 040 123
    
```

```

BEQ 50440$
BIT #TS.OFL,@TSSR(R5)
BEQ 50441$
INC FTLCNT(R5)
MOV DEVTBL(R5),-(SP)
MOV #OFLINM, -(SP)
MOV #2, -(SP)
MOV SP,RO
TRAP C$PNTF
ADD #6,SP

50441$:
BR 50442$
50440$:
INC FTLCNT(R5)
MOV DEVTBL(R5),-(SP)
MOV #NRDYM, -(SP)
MOV #2, -(SP)
MOV SP,RO
TRAP C$PNTF
ADD #6,SP

50442$:
50437$:
BR 50434$
50435$:
TRAP C$AUTO
    
```

L10013:

AUTODM: .ASCII /#ABUS TRAP AT #06#N/

.ASCIZ /#AINTERFACE BAD OR NOT SET TO ABOVE AD#N/

	023523	105	124	040	
	023526	124	117	040	
	023531	101	102	117	
	023534	126	105	040	
	023537	101	104	045	
	023542	116	000		
4754	023544	045	101	125	NRDYM: .ASCIZ /#AUNIT #D1#A NOT RDY#N/
	023547	116	111	124	
	023552	040	045	104	
	023555	061	045	101	
	023560	040	116	117	
	023563	124	040	122	
	023566	104	131	045	
	023571	116	000		

```

4755          .EVEN
4756
4757          ; DEVICE BUS TRAP HANDLER
4758          ; OUTPUT: TRAPD4 BYTE 1: TRAPED AT 4
4759          ;                               0: NO TRAP
4760
4761 023574    TRAP4:: LET TRAPD4 :B= TRAPD4 + #1
023574 105237 003536
4762 023600    RTI
4763
4764
4765

```

INCB TRAPD4

```

4767      .SBTTL  CLEANUP CODING SECTION
4768
4769      ;**
4770      ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
4771      ; AT THE END OF EACH PASS.
4772      ;--
4773
4774 023602      BGNCLN
023602      L$CLEAN::
4775
4782
4783 023602 004737 017300      JSR  PC,FIRSTU      ;FIND FIRST UNIT.
4784 023606      WHILE DEVTBL(R5) NE #END DO
                                50443$:
                                CMP  DEVTBL(R5),#END
                                BEQ  50444$
4785 023614 001410 026527 002536 177777      JSR PC,WSSR      ;WAIT FOR UNIT READY OR TIMEOUT.
4786 023616 004737 011672      CLRVEC          TSVCT(R5)      ;RELEASE INTERRUPT VECTORS FOR ALL DEV.
                                MOV  TSVCT(R5),RO
                                TRAP C$CVEC
4787 023622 016500 002476
4788 023626 104436
4789 023630 004737 017346      JSR  PC,NEXTU      ;FIND NEXT UNIT.
4790 023634      ENDDO
                                BR   50443$
                                50444$:
4791 023636      EXIT  CLN
                                TRAP C$EXIT
4792 023636 104432
4793 023640 000002
                                .WORD  L10014-.
4802      .EVEN
4803
4804 023642      ENDCLN
023642      L10014:
023642 104412      TRAP  C$CLEAN
    
```

```

4806          .SBTTL  DROP UNIT SECTION
4807
4808          ;**
4809          ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
4810          ; TO NO LONGER BE TESTED.  THAT CODE SHALL BE EXECUTED WHEN D0DU
4811          ;MACRO IS CALLED WHILE IDU FLAG IS NOT SET BY OPERATOR
4812          ;--
4813
4814 023644      BGNDU
4815 023644      L$DU::
4821 023644      LET R5 := R0 SHIFT 1          ;R5 = LOGICAL DEVICE NUMBER X 2.
         023644 010005          MOV      R0,R5
         023646 006305          ASL      R5
4822 023650      LET DEVTBL(R5) := #NINUSE      ;SET NOT IN USE FLAG FOR THE DEVICE.
         023650 012765 177774 002536          MOV      #NINUSE,DEVTBL(R5)
4823 023656      PRINTF #DROPPM,DROPN          ;PRINT DROP DEVICE MESSAGE
         023656 013746 017622          MOV      DROPN,-(SP)
         023662 012746 004700          MOV      #DROPPM,-(SP)
         023666 012746 000002          MOV      #2,-(SP)
         023672 010600          MOV      SP,R0
         023674 104417          TRAP    C$PNTF
         023676 062706 000006          ADD      #6,SP
4824 023702      EXIT  DU
         023702 000167          .WORD   J$JMP
         023704 000000          .WORD   L10015-2-.
4836          .EVEN
4837
4838 023706      ENDDU
         023706      L10015:
         023706 104453          TRAP    C$DU
    
```



```

4841          .SBTTL  ADD UNIT SECTION
4842
4843          ;**
4844          ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
4845          ; TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING.  IF
4846          ; "EF.AUNIT" IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
4847          ;--
4848
4849 023710          BGN AU
4850          L$AU::
4851
4852          LET R5 := R0 SHIFT 1          ;R5 = LOGICAL DEVICE NUMBER X 2.
4853          MOV R0,R5
4854          ASL R5
4855
4856          LET DEVTBL(R5) := R0          ;STORE UNIT # IN DEVICE TABLE.
4857          MOV R0,DEVTBL(R5)
4858          GPHARD R0,R0          ;GET HARDWARE P TABLE FROM SUPER.
4859          TRAP C$GPHRD
4860          LET TSSR(R5) := (R0)          ;SAVE TSSR ADDRESS.
4861          MOV (R0),TSSR(R5)
4862          LET TSDB(R5) := (R0) - #2          ;SAVE TSDB ADDRESS.
4863          MOV (R0),TSDB(R5)
4864          SUB #2,TSDB(R5)
4865          LET TSVCT(R5) := (R0)          ;SAVE INTERRUPT VECTOR ADDRESS.
4866          MOV (R0),TSVCT(R5)
4867          SETVEC TSVCT(R5),TS4INT(R5),#INTPRI          ;SET UP INTERRUPT PROCESSING CONDITIONS.
4868          MOV #INTPRI,-(SP)
4869          MOV TS4INT(R5),-(SP)
4870          MOV TSVCT(R5),-(SP)
4871          MOV #3,-(SP)
4872          TRAP C$SVEC
4873          ADD #10,SP
4874
4875          LET INTFLG(R5) := #0          ;CLEAR INTERRUPT FLAGS.
4876          CLR INTFLG(R5)
4877
4878          EXIT AU
4879
4880          .EVEN
4881          ENDAU
4882          L10016:
4883          TRAP C$AU
4884          ENDMOD

```

```

4887
4898 .TITLE HARDWARE TESTS
4899
4900 .SBTTL TEST 1: BASIC FUNCTIONS.
4901
4902 ;**
4903 ; TEST TO EXECUTE ALL TK25 FUNCTIONS.
4904 ;**
4905
4906 024004          BGNMOD
4907
4908 024004          BGNTST
4909 024004          T1::
4910 024004 105037 003521      LET RANDOM :B= #0          ;CLR THE RANDOM OPERATIONS FLAG.
4911 024010 105037 003520      LET EXPBOT :B= #0          ;CLR EXPECT BOT FLAG.
4912 024014          BGNSUB          ;SUBTEST 1 - SET CHAR, DRIVE INIT, GET STATUS.
4913 024014 104402          T1.1:          TRAP C#BSUB
4914 024016          LET R2 := #BFSEQO          ;ADR OF CMD SEQ.
4915 024016 012702 024642          JSR PC,BFSEQ          ;SET UP CMD SEQ.
4916 024022 004737 024616          JSR PC,EXALL          ;EXECUTE CMD SEQ ON ALL DEVICES.
4917 024026 004737 006744          JSR PC,FIRSTU          ;FIND THE FIRST UNIT.
4918 024032 004737 017300          WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
4919 024036          50445$:
4920 024036 026527 002536 177777      CMP DEVTBL(R5),#END
4921 024044 001434          BEQ 50446$
4922 024046          LET R2 := MSGPKA(R5)          ;GET MSG PACKET ADR.
4923 024052 016502 002506          LET R2 := R2 + #12          ;GET XSTAT2 ADR.
4924 024056 062702 000012          LET TS4CL(R5) := (R2) CLR.BY #177400 ;STORE CODE LEVEL FROM DTR BYTE.
4925 024062 011265 002526          IF PASCNT(R5) EQ #1 THEN ;IF THIS IS PASS 1 THEN:
4926 024070 026527 003260 000001      CMP PASCNT(R5),#1
4927 024076 001014          BNE 50447$
4928 024100          PRINTF #CODELM,DEVTBL(R5),TS4CL(R5) ;PRINT THE TK25 MICROCODE LEVEL.
4929 024100 016546 002526          MOV TS4CL(R5),-(SP)
4930 024104 016546 002536          MOV DEVTBL(R5),-(SP)
4931 024110 012746 004054          MOV #CODELM, -(SP)
4932 024114 012746 000003          MOV #3, -(SP)
4933 024120 010600          MOV SP,R0
4934 024122 104417          TRAP C#PNTF
4935 024124 062706 000010          ADD #10,SP
4936 024130          ENDF
4937 024130          JSR PC,NEXTU          ;FIND NEXT UNIT.
4938 024134          ENDDO          BR 50445$
4939 024136          ENDSUB          50446$:

```


4961	024246			LET R2 := #BFSEQ5	;ADR OF CMD SEQ.	
	024246	012702	025134		MOV	#BFSEQ5,R2
4962	024252	004737	024616	JSR PC,BFSEQ	;SET UP CMD SEQ.	
4963	024256	004737	006744	JSR PC,EXALL	;EXECUTE CMD SEQ ON ALL DEVICES.	
4964	024262			ENDSUB		
	024262			L10025:		
	024262	104403				TRAP C#ESUB
4965				BGNSUB	;SUBTEST 7 - WRITE RETRY.	
4966	024264			T1.7:		
	024264					TRAP C#BSUB
	024264	104402				
4967				LET R2 := #BFSEQ6	;ADR OF CMD SEQ.	
4968	024266	012702	025206		MOV	#BFSEQ6,R2
4969	024272	004737	024616	JSR PC,BFSEQ	;SET UP CMD SEQ.	
4970	024276	004737	006744	JSR PC,EXALL	;EXECUTE CMD SEQ ON ALL DEVICES.	
4971	024302			ENDSUB		
	024302			L10026:		
	024302	104403				TRAP C#ESUB
4972				BGNSUB	;SUBTEST 8 - READ REV RETRY.	
4973	024304			T1.8:		
	024304					TRAP C#BSUB
	024304	104402				
4974				LET R2 := #BFSEQ7	;ADR OF CMD SEQ.	
4975	024306	012702	025240		MOV	#BFSEQ7,R2
4976	024312	004737	024616	JSR PC,BFSEQ	;SET UP CMD SEQ.	
4977	024316	004737	006744	JSR PC,EXALL	;EXECUTE CMD SEQ ON ALL DEVICES.	
4978	024322			ENDSUB		
	024322			L10027:		
	024322	104403				TRAP C#ESUB
4979				BGNSUB	;SUBTEST 9 - READ FWD RETRY.	
4980	024324			T1.9:		
	024324					TRAP C#BSUB
	024324	104402				
4981				LET R2 := #BFSEQ8	;ADR OF CMD SEQ.	
4982	024326	012702	025272		MOV	#BFSEQ8,R2
4983	024332	004737	024616	JSR PC,BFSEQ	;SET UP CMD SEQ.	
4984	024336	004737	006744	JSR PC,EXALL	;EXECUTE CMD SEQ ON ALL DEVICES.	
4985	024342			ENDSUB		
	024342			L10030:		
	024342	104403				TRAP C#ESUB
4986				BGNSUB	;SUBTEST 10- CLEAN.	
4987	024344			T1.10:		
	024344					TRAP C#BSUB
	024344	104402				
4988				LET R2 := #BFSEQ9	;ADR OF CMD SEQ.	
4989	024346	012702	025324		MOV	#BFSEQ9,R2
4990	024352	004737	024616	JSR PC,BFSEQ	;SET UP CMD SEQ.	
4991	024356	004737	006744	JSR PC,EXALL	;EXECUTE CMD SEQ ON ALL DEVICES.	
4992	024362			ENDSUB		
	024362			L10031:		
	024362	104403				TRAP C#ESUB
4993				BGNSUB	;SUBTEST 11 - WTV SWAPPED DATA BYTES.	
4994	024364					

```

024364          T1.11:
024364 104402          TRAP C#BSUB
4995
4996 024366          LET R2 := #BFSE10          ;ADR OF CMD SEQ.
024366 012702 025346          ;MOV #BFSE10,R2
4997 024372 004737 024616          JSR PC,BFSEQ          ;SET UP CMD SEQ.
4998 024376 004737 006744          JSR PC,EXALL          ;WRITE/VERIFY RECORDS 1 AND 2.
4999 024402          LET SWBFLG :B= #1          ;ENABLE BYTE SWAPPING.
024402 112737 000001 003524          MOVB #1,SWBFLG
5000 024410 004737 006744          JSR PC,EXALL          ;WRITE/VERIFY RECORDS 3 AND 4.
5001 024414          LET SWBFLG :B= #0          ;DISABLE BYTE SWAPPING.
024414 105037 003524          CLRB SWBFLG
5002 024420          ENDSUB
024420          L10032:
024420 104403          TRAP C#ESUB
5003
5004 024422          LET R2 := DATAWT * #10.          ;INIT WRITE BUFFER POINTER.
024422 013702 003414          ;MOV DATAWT,R2
024426 062702 000012          ;ADD #10.,R2
5005 024432          WHILE R2 NE DATAWT DO ;UNTIL 10 BYTES HAVE BEEN SWAPPED.
024432          ;50450$:
024432 020237 003414          ;CMP R2,DATAWT
024436 001402          ;BEQ 50451$
5006 024440 000342          ;SWAB -(R2)          ;SWAP DATA BYTES IN WRITE BUFFER.
5007 024442          ENDDO
024442 000773          ;BR 50450$
024444          ;50451$:
5008 024444          LET T1SWB :B= T1SWB * #1          ;SET T1 SWAP BYTES FLAG FOR "CKDATA" SUBR
024444 105237 003530          ;INCB T1SWB
5009
5010 024450          BGNSUB          ;SUBTEST 12 - READ SWAPPED DATA BYTES.
024450          T1.12:
024450 104402          TRAP C#BSUB
5011
5012 024452          LET CMDWRD := #RDR          ;CMD IS READ REV.
024452 012737 104401 003426          ;MOV #RDR,CMDWRD
5013 024460 004737 016254          JSR PC,VFEXC          ;VERIFY ODD LENGTH SWAP (RECORD 4).
5014 024464          LET CNDPKT*CP.CNT := #12          ;CHANGE BYTE COUNT TO 10.
024464 012737 000012 002322          ;MOV #12,CNDPKT*CP.CNT
5015 024472 004737 016254          JSR PC,VFEXC          ;VERIFY EVEN LENGTH SWAP (RECORD 3).
5016 024476          LET SWBFLG :B= #1          ;ENABLE BYTE SWAPPING.
024476 112737 000001 003524          MOVB #1,SWBFLG
5017 024504          LET CNDPKT*CP.CNT := #11          ;CHANGE BYTE COUNT TO 9.
024504 012737 000011 002322          ;MOV #11,CNDPKT*CP.CNT
5018 024512 004737 016254          JSR PC,VFEXC          ;VERIFY ODD LENGTH SWAP (RECORD 2).
5019 024516          LET CNDPKT*CP.CNT := #12          ;CHANGE BYTE COUNT TO 10.
024516 012737 000012 002322          ;MOV #12,CNDPKT*CP.CNT
5020 024524 004737 016254          JSR PC,VFEXC          ;VERIFY EVEN LENGTH SWAP (RECORD 1).
5021 024530          LET CMDWRD := #RDF          ;CMD IS READ FWD.
024530 012737 104001 003426          ;MOV #RDF,CMDWRD
5022 024536 004737 016254          JSR PC,VFEXC          ;VERIFY EVEN LENGTH SWAP (RECORD 1).
5023 024542          LET CNDPKT*CP.CNT := #11          ;CHANGE BYTE COUNT TO 9.
024542 012737 000011 002322          ;MOV #11,CNDPKT*CP.CNT
5024 024550 004737 016254          JSR PC,VFEXC          ;VERIFY ODD LENGTH SWAP (RECORD 2).
5025 024554          LET SWBFLG :B= #0          ;DISABLE BYTE SWAPPING.
024554 105037 003524          CLRB SWBFLG
5026 024560          LET CNDPKT*CP.CNT := #12          ;CHANGE BYTE COUNT TO 10.

```

5027	024560	012737	000012	002322					
5028	024566	004737	016254		JSR	PC,VFEXC			
	024572				LET	CMDPKT.CP.CNT := #11			
5029	024600	012737	000011	002322					
5030	024600	004737	016254		JSR	PC,VFEXC			
5031	024604				ENDSUB				
	024604				L10033:				
	024604	104403							
5032									
5033	024606				LET	T1SWB :B= #0			
	024606	105037	003530						
5034									
5035									
5036	024612				EXIT	TST			
	024612	104432							
	024614	000554							

							MOV	#12,CMDPKT.CP.CNT	
									;VERIFY EVEN LENGTH SWAP (RECORD 3).
									;CHANGE BYTE COUNT TO 9.
							MOV	#11,CMDPKT.CP.CNT	
									;VERIFY ODD LENGTH SWAP (RECORD 4).
							TRAP	C#ESUB	
									;CLEAR T1 SWAP BYTES FLAG
							CLRB	T1SWB	
							TRAP	C#EXIT	
							.WORD	L10017-	

```

5038      ;      SUBROUTINE TO MOVE A COMMAND SEQUENCE TO THE SEQUENCE TABLE.
5039      ;      INPUTS:          R2 = FWA OF COMMAND SEQUENCE.
5040      ;      OUTPUTS:
5041      ;      REGISTERS:
5042      ;      CALLS:
5043
5044 024616      BFSEQ: LET R1 := #CMDSEQ          ;INIT SEQ TABLE ADDRESS.
                    024616 012701 003542      MOV      #CMDSEQ,R1
5045 024622      WHILE (R2) NE #END DO          ;WHILE THERE ARE MORE COMMANDS:
                    024622                    50452$:
                    024622 021227 177777      CMP      (R2),#END
                    024626 001402      BEQ      50453$
5046 024630      LET (R1)+ := (R2)+          ;MOVE COMMANDS TO SEQ TABLE.
                    024630 012221      MOV      (R2)+,(R1)+
5047 024632      ENDDO
                    024632 000773      BR      50452$
                    024634                    50453$:
5048 024634      LET (R1) := #END          ;STORE END OF SEQUENCE CODE.
                    024634 012711 177777      MOV      #END,(R1)
5049 024640      RTS      PC.          ;RETURN.
5050
5051
5052
5053      ;      BASIC FUNCTION COMMAND SEQUENCE
5054
5055 024642 140004      BFSEQ0: .WORD      SCH          ;SET CHAR. 200.          (1)
5056 024644 000200      200
5057 024646 000001      1
5058 024650 000000      0
5059 024652 100013      DRI          ;DRIVE INIT.          (2)
5060 024654 000001      1
5061 024656 000001      1
5062 024660 000000      0
5063 024662 140004      SCH          ;SET CHAR. 20          (3)
5064 024664 000020      20
5065 024666 000001      1
5066 024670 000000      0
5067 024672 100017      GES          ;GET STATUS.          (4)
5068 024674 000001      1
5069 024676 000001      1
5070 024700 000000      0
5071 024702 140004      SCH          ;SET CHAR. 40.          (5)
5072 024704 000040      40
5073 024706 000001      1
5074 024710 000000      0
5075 024712 177777      .WORD      END
5076
5077 024714 102010      BFSEQ1:          RWD          ;REWIND TWICE.          (6)
5078 024716 000001      1
5079 024720 000002      2
5080 024722 000000      0
5081 024724 177777      .WORD      END
5082
5083 024726 104105      BFSEQ2:          WTV          ;WRITE/VERIFY PAT 1.  (7)
5084 024730 010000      DATCNT
5085 024732 000001      1
5086 024734 000001      1
    
```

5087	024736	104105		WTV		;WTV PAT 2.	(8)
5088	024740	010000		DATCNT			
5089	024742	000001		1			
5090	024744	000002		2			
5091	024746	104105		WTV		;WTV PAT 3.	(9)
5092	024750	010000		DATCNT			
5093	024752	000001		1			
5094	024754	000003		3			
5095	024756	104105		WTV		;WTV PAT 4.	(10)
5096	024760	010000		DATCNT			
5097	024762	000001		1			
5098	024764	000004		4			
5099	024766	104105		WTV		;WTV PAT 5.	(11)
5100	024770	010000		DATCNT			
5101	024772	000001		1			
5102	024774	000005		5			
5103	024776	104105		WTV		;WTV PAT 6.	(12)
5104	025000	010000		DATCNT			
5105	025002	000001		1			
5106	025004	000006		6			
5107	025006	104105		WTV		;WTV PAT 0.	(13)
5108	025010	010000		DATCNT			
5109	025012	000001		1			
5110	025014	000000		0			
5111	025016	177777	.WORD	END			
5112							
5113	025020	100011	BFSEQ3:	WTM		;WRITE TAPE MARK.	(14)
5114	025022	000001		1			
5115	025024	000001		1			
5116	025026	000000		0			
5117	025030	104005		WRT		;WRITE 10 RECORDS.	(15)
5118	025032	010000		DATCNT			
5119	025034	000010		10			
5120	025036	000001		1			
5121	025040	100411		ERS		;ERASE 10 TIMES.	(16)
5122	025042	000001		1			
5123	025044	000010		10			
5124	025046	000000		0			
5125	025050	100011		WTM		;WRITE TAPE MARK.	(17)
5126	025052	000001		1			
5127	025054	000001		1			
5128	025056	000000		0			
5129	025060	101011		WTR		;WTM RETRY	(18)
5130	025062	000001		1			
5131	025064	000001		1			
5132	025066	000000		0			
5133	025070	177777	.WORD	END			
5134							
5135	025072	105410	BFSEQ4:	SFR		;SPACE 2 FILES REV.	(19)
5136	025074	000002		2			
5137	025076	000001		1			
5138	025100	000000		0			
5139	025102	105010		SFF		;SPACE 2 FILES FWD.	(20)
5140	025104	000002		2			
5141	025106	000001		1			
5142	025110	000000		0			
5143	025112	105410		SFR		;SPACE 2 FILES REV.	(21)

5144	025114	000001		1		
5145	025116	000002		2		
5146	025120	000000		0		
5147	025122	105010		SFF	;SPACE 2 FILES FWD.	(22)
5148	025124	000001		1		
5149	025126	000002		2		
5150	025130	000000		0		
5151	025132	177777	.WORD	END		
5152						
5153	025134	102010	BFSEQ5:	RWD	;REWIND.	(23)
5154	025136	000001		1		
5155	025140	000001		1		
5156	025142	000000		0		
5157	025144	104010		SRF	;SPACE 7 RECORDS FWD.	(24)
5158	025146	000007		7		
5159	025150	000001		1		
5160	025152	000000		0		
5161	025154	104410		SRR	;SPACE 7 RECORDS REV.	(25)
5162	025156	000007		7		
5163	025160	000001		1		
5164	025162	000000		0		
5165	025164	104010		SRF	;SPACE 7 RECORDS FWD.	(26)
5166	025166	000001		1		
5167	025170	000007		7		
5168	025172	000000		0		
5169	025174	104410		SRR	;SPACE 7 RECORDS REV.	(27)
5170	025176	000001		1		
5171	025200	000007		7		
5172	025202	000000		0		
5173	025204	177777	.WORD	END		
5174						
5175	025206	102010	BFSEQ6:	RWD	;REWIND.	(28)
5176	025210	000001		1		
5177	025212	000001		1		
5178	025214	000000		0		
5179	025216	104005		WRT	;WRITE.	(29)
5180	025220	010000		DATCNT		
5181	025222	000001		1		
5182	025224	000001		1		
5183	025226	105005		WRR	;WRITE RETRY.	(30)
5184	025230	010000		DATCNT		
5185	025232	000001		1		
5186	025234	000001		1		
5187	025236	177777	.WORD	END		
5188						
5189	025240	104401	BFSEQ7:	RDR	;READ REV.	(31)
5190	025242	010000		DATCNT		
5191	025244	000001		1		
5192	025246	000001		1		
5193	025250	105401		RNR	;READ NEXT REV.	(32)
5194	025252	010000		DATCNT		
5195	025254	000001		1		
5196	025256	000001		1		
5197	025260	125401		RNF	;READ NEXT FWD.	(33)
5198	025262	010000		DATCNT		
5199	025264	000001		1		
5200	025266	000001		1		

```

5201 025270 177777          .WORD  END
5202
5203 025272 104001          BFSEQ8:  RDF          ;READ FWD.          (34)
5204 025274 010000          DATCNT
5205 025276 000001          1
5206 025300 000001          1
5207 025302 105001          RPF          ;READ PREVIOUS FWD. (35)
5208 025304 010000          DATCNT
5209 025306 000001          1
5210 025310 000001          1
5211 025312 125001          RPR          ;READ PREVIOUS REV. (36)
5212 025314 010000          DATCNT
5213 025316 000001          1
5214 025320 000001          1
5215 025322 177777          .WORD  END
5216
5217 025324 101012          BFSEQ9: .WORD  CLN          ;CLEAN.          (37)
5218 025326 000001          1
5219 025330 000001          1
5220 025332 000000          0
5221 025334 102010          RWD          ;REWIND          (38)
5222 025336 000001          1
5223 025340 000001          1
5224 025342 000000          0
5225 025344 177777          .WORD  END          ;END OF SEQUENCE.
5226
5227 025346 104105          BFSE10: WTV          ;WRITE/VERIFY EVEN LENGTH. (39)
5228 025350 000012          12
5229 025352 000001          1
5230 025354 000000          0
5231 025356 104105          WTV          ;WRITE/VERIFY ODD LENGTH. (40)
5232 025360 000011          11
5233 025362 000001          1
5234 025364 000000          0
5235 025366 177777          .WORD  END
5236          .EVEN
5237
5238 025370          ENDTST
          025370          L10017:
          025370 104401          TRAP  C$ETST

```

```

5240          .SBTTL TEST 2: DATA RELIABILITY.
5241
5242          ;++
5243          ; TEST TO CHECK THE DATA RELIABILITY OF THE TK25.
5244          ;--
5245 025372    BGNTST
5246 025372    T2::
5247 025372    112737 000001 003521    LET RANDOM :B= #1          ;SET THE RANDOM OPERATIONS FLAG.
5248 025400    025372 112737 000001 003521    LET EXPBOT :B= #0          ;CLEAR EXPECT BOT FLAG.
5249 025404    025400 105037 003520    LET R2 := #DATCNT - #1    ;SET UP THE RECORD LENGTH MASK,
5250 025412    025404 012702 010000    LET LENMSK := COMP R2    ;ALLOW MAXIMUM BUFFER.
5251 025422    025412 010237 003436    JSR PC,SETCH              ;CMD 1 = SET CHARACTERISTIC.
5252 025426    025416 005137 003436    IFB STAFLG NE #0 THEN    ;IF STARTING THEN:
5253 025434    025426 105737 003534    JSR PC,SETRW              ;CMD2=REWIND
5254 025440    025432 001404 006724    LET STAFLG :B= #0        ;CLR START FLAG.
5255 025444    025437 004737 006724    ENDIF                      ;
5256 025444    025444 105037 003534    LET (R1)+ := #WRT         ;CMD3 = WRITE.
5257 025450    025444 012721 104005    LET (R1)+ := #DATCNT      ;SET BRF TO MAX FOR PATTERN GENERATION.
5258 025454    025450 012721 010000    LET R2 := COMP #RNOPSC    ;
5259 025462    025454 012702 177740    LET (R1)+ := R2          ;31 OPERATIONS.
5260 025464    025460 005102 177740    LET (R1)+ := #RANP        ;RANDOM PATTERN.
5261 025470    025464 012721 000007    REPEAT                     ;REPEAT TO EOT:
5262 025470    025470 020127 003632    WHILE R1 LT #SEQEND DO    ;FILL SEQ TBL WITH RANDOM CMDS.
5263 025476    025474 002012 003632    LET RANS := RANS + RANB    ;
5264 025504    025476 063737 003440 003442    LET R2 := RANS CLR.BY #177741 ;R2 = RANDOM # (0 - 36).
5265 025514    025504 013702 003442    JSR PC,#RANCMD(R2)        ;SET UP A RANDOM CMD + BRF.
5266 025520    025510 042702 177741    ENDDO                      ;
5267 025522    025520 000763 177777    LET (R1) := #END          ;STORE END OF SEQUENCE CODE IN TABLE.
5268 025526    025522 012711 177777    JSR PC,EXALL              ;GO EXECUTE ALL CMDS IN SEQUENCE TABLE.
5269 025532    025526 004737 006744    LET R1 := #CMDSEQ         ;INIT CMD SEQ TBL POINTER.

```

```
5270 025532 012701 003542          UNTIL R2 NE #0          ;REPEAT UNTIL EOT IS REACHED
    025536 005702                   ;                          MOV #CMDSEQ,R1
    025540 001753                   ;                          TST R2
5271 025542 105237 003531          LET ALLEOT :B= ALLEOT + #1 ;FLAG ALL UNITS @ EOT   BEQ 50455$
    025546 000240                   ;                          INCB ALLEOT
5272 025546 000240                   NOP
5273 025550 000240                   NOP
5274 025552 000240                   NOP
5275 025554 004737 027640          JSR PC,T5WEOT          ;WRITE ONE RECORD BEYOND EOT ON ALL UNITS
5276                                     ;SO THAT SHORTER READ STOP DISTANCE
5277                                     ;SHALL POSITION HEAD IN CLEAN IRG GAP
5278                                     ;READ REV THAT EXTRA REC TO RE-POSITION THE TAPE
5279 025560 004737 025712          JSR PC,RANRD          ;SET UP READ REV/FWD CMDS.
5280 025564 012737 177740 003546    LET CMDSEQ+4 := COMP #RNOPSC ;# OF RECORDS FOR READ REV.
    025572 005137 003546          ;                          MOV #RNOPSC,CMDSEQ+4
5281 025576 013737 003546 003556    LET CMDSEQ+14 := CMDSEQ+4 ;# OF RECORDS FOR READ FORWARD.
    025576 013737 003546 003556    ;                          COM CMDSEQ+4
5282 025604 012711 177777          LET (R1) := #END        ;STORE END OF SEQUENCE CODE IN SEQ TABLE.
    025604 012711 177777          ;                          MOV #END,(R1)
5283 025610 004737 006744          JSR PC,EXALL          ;GO EXECUTE READ REV/FWD OF LAST N RECORDS.
5284 025614 105037 003531          LET ALLEOT :B= #0      ;CLEAR ALL UNITS @ EOT FLAG
    025614 105037 003531          ;                          CLRB ALLEOT
5285 025620 112737 000001 003523    LET RPTFLG :B= #1      ;REQUEST PERFORMANCE REPORT DURING REWIND.
    025620 112737 000001 003523    ;                          MOV #1,RPTFLG
5286 025626 012701 003542          LET R1 := #CMDSEQ      ;INIT SEQ TBL POINTER.
    025626 012701 003542          ;                          MOV #CMDSEQ,R1
5287 025632 004737 006724          JSR PC,SETRW          ;STORE REWIND IN SEQ TBL.
5288 025636 012711 177777          LET (R1) := #END        ;STORE END IN SEQ TBL.
    025636 012711 177777          ;                          MOV #END,(R1)
5289 025642 004737 006744          JSR PC,EXALL          ;EXECUTE REWIND CMD ON ALL UNITS
5290                                     TRAP C$EXIT
5291 025646 104432                   .WORD L10034-.
    025646 104432
    025650 000174
5292
```

```

5294 ; ADDRESSES OF SUBROUTINES USED TO SET UP RANDOM OPERATIONS IN
5295 ; THE DATA RELIABILITY TEST.
5296
5297 025652 025766 RANCMD: RANWR ;WRITE.
5298 025654 025766 RANWR ;WRITE.
5299 025656 025766 RANWR ;WRITE.
5300 025660 025766 RANWR ;WRITE.
5301 025662 025766 RANWR ;WRITE.
5302 025664 025766 RANWR ;WRITE.
5303 025666 025766 RANWR ;WRITE.
5304 025670 025766 RANWR ;WRITE.
5305 025672 025712 RANRD ;READ.
5306 025674 025712 RANRD ;READ.
5307 025676 025712 RANRD ;READ.
5308 025700 025712 RANRD ;READ.
5309 025702 025712 RANRD ;READ.
5310 025704 025712 RANRD ;READ.
5311 025706 025712 RANRD ;READ.
5312 025710 025712 RANRD ;READ.
5313
5314
5315
5316
5317
5318 ; SUBROUTINE TO SET UP READ COMMANDS IN SEQUENCE TABLE.
5319 ;
5320 ; INPUTS:
5321 ; OUTPUTS:
5322 ; REGISTERS: R2
5323 ; CALLS:
5324 025712 012721 104410 RANRD: LET (R1)+ := #SRR ;STORE SPACE RECORD REVERSE CMD
5325 025716 063737 003442 003440 LET RANB := RANB + RANS MOV #SRR,(R1)+
5326 025724 013702 003440 LET R2 := RANB CLR.BY #RNOPSC ADD RANS,RANB
5327 025730 042702 177740 LET (R1)+ := R2 ;SET REPOSITION COUNT MOV RANB,R2
5328 025734 010221 LET (R1)+ := #1 ;DO ONCE BIC #RNOPSC,R2
5329 025742 012721 000001 LET (R1)+ := #RANP ;RANDOM PATTERN. MOV R2,(R1)+
5330 025746 012721 104001 LET (R1)+ := #RDF ;STORE READ FWD CMD. MOV #1,(R1)+
5331 025752 012721 010000 LET (R1)+ := #DATCNT ;SET BRF TO MAX TO READ RANDOM LENGTHS. MOV #RANP,(R1)+
5332 025756 010221 LET (R1)+ := R2 ;SET RANDOM # OF OPERATIONS. MOV #RDF,(R1)+
5333 025760 012721 000007 LET (R1)+ := #RANP ;RANDOM PATTERN. MOV #DATCNT,(R1)+
5334 025764 000207 RTS PC MOV R2,(R1)+
    
```

```

5336      ;      SUBROUTINE TO SET UP A WRITE COMMAND IN THE SEQUENCE TABLE.
5337      ;      INPUTS:
5338      ;      OUTPUTS:
5339      ;      REGISTERS:
5340      ;      CALLS:
5341
5342 025766      RANWR: LET (R1) := @WRT      ;STORE WRITE CMD.
          025766 012721 104005      MOV      @WRT,(R1)
5343 025772      JSR PC,RANW      ;STORE BR# , # OF OPERATIONS, PATTERN.
          004737 026012      RTS PC
5344 025776      000207
5345
5346
5347
5348
5349
5350      ;      SUBROUTINE TO SET UP A WRITE/VERIFY COMMAND IN THE SEQUENCE TABLE.
5351      ;      INPUTS:
5352      ;      OUTPUTS:
5353      ;      REGISTERS:
5354      ;      CALLS:
5355
5356 026000      RANWV: LET (R1) := @WTV      ;STORE WRITE/VERIFY CMD.
          026000 012721 104105      MOV      @WTV,(R1)
5357 026004      JSR PC,RANW      ;STORE BR# , # OF OPERATIONS, PATTERN.
          004737 026012      RTS      PC
5358 026010      000207
5359
5360
5361
5362
5363
5364      ;      SUBROUTINE TO STORE BR# , # OF OPERATIONS, PATTERN IN COMMAND
5365      ;      SEQUENCE TABLE FOR WRITE AND WRITE/VERIFY COMMANDS.
5366      ;      INPUTS:
5367      ;      OUTPUTS:
5368      ;      REGISTERS:      R2
5369      ;      CALLS:
5370
5371 026012      RANW: LET (R1) := @DATCNT      ;SET BR# TO MAX FOR PATTERN GENERATION.
          026012 012721 010000      MOV      @DATCNT,(R1)
5372      ;RANDOM BR# WILL BE GENERATED FOR EACH RECORD.
5373 026016      LET RANB := RANB * RANS
          026016 063737 003442 003440      ADD      RANS,RANB
5374 026024      LET R2 := RANB CLR.BY @RNOPSC
          026024 013702 003440      MOV      RANB,R2
          026030 042702 177740      BIC      @RNOPSC,R2
5375 026034      LET (R1) := R2      ;SET RANDOM # OF OPERATIONS.
          026034 010221      MOV      R2,(R1)
5376 026036      LET (R1) := @RANP      ;RANDOM PATTERN.
          026036 012721 000007      MOV      @RANP,(R1)
5377 026042      RTS PC      ;RETURN.
5378
5379      .EVEN
5380
5381 026044      ENDTST
          026044      L10034:
          026044 104401      TRAP      C#ETST
5382

```

```

5384 .SBTTL TEST 3: WRITE AND READ STREAMING TEST.
5385
5386 ;
5387 ;
5388 ; THIS TEST STREAM WRITES 9000 RECORDS OF 4096 BYTES EACH.
5389 ; DATA IS THEN VERIFIED BY PERFORMING A REWIND OPERATION, FOLLOWED
5390 ; BY A READ FORWARD OPERATON FOR THE 9000 RECORDS.
5391 ;
5392 ;
5393 ;--
5394 026046 BGNTST
    026046 T3::
5395
5396 026046 IFB HERE EQ #0 THEN
    026046 105737 003340
    026052 001012
5397 026054 PRINTF #TAPCAP
    026054 012746 021571
    026060 012746 000001
    026064 010600
    026066 104417
    026070 062706 000004
5398 026074 LET HERE :B= HERE * #1
    026074 105237 003340
5399 026100 ENDF
    026100
5400 026100 LET RANDOM :B= #0
    026100 105037 003521
5401 026104 LET EXPBOT :B= #0
    026104 105037 003520
5402 026110 JSR PC, SETCH
    026110 004737 006700
5403 026114 JSR PC, SETRW
    026114 004737 006724
5404 026120 LET STAF LG :B= #0
    026120 105037 003534
5405 026124 LET (R1) := #END
    026124 012711 177777
5406 026130 JSR PC, EXALL
    026130 004737 006744
5407
5408 026134 JSR PC, FIRSTU
    026134 004737 017300
5409
5410 ; *****
5411 ;
5412 ;WRITE AND READ EACH UNIT IN TURN BEFORE GOING ON TO THE NEXT.
5413 ;
5414 ; *****
5415
5416 026140 WHILE DEVTBL(R5) NE #END DO
    026140
    026140 026527 002536 177777
    026146 001544
5417
5418 026150 LET BTPT := BTADDR(R5)
    026150 016537 002550 003516
5419 026156 LET #BTPT := #0
    026156 005077 155334
5420 026162 LET STREAM :B= #255.
    026162 112737 000377 003532
    
```

THIS TEST STREAM WRITES 9000 RECORDS OF 4096 BYTES EACH. DATA IS THEN VERIFIED BY PERFORMING A REWIND OPERATION, FOLLOWED BY A READ FORWARD OPERATON FOR THE 9000 RECORDS.

BGNTST

T3::

IFB HERE EQ #0 THEN

PRINTF #TAPCAP

LET HERE :B= HERE * #1

ENDIF

LET RANDOM :B= #0

LET EXPBOT :B= #0

JSR PC, SETCH

JSR PC, SETRW

LET STAF LG :B= #0

LET (R1) := #END

JSR PC, EXALL

JSR PC, FIRSTU

WRITE AND READ EACH UNIT IN TURN BEFORE GOING ON TO THE NEXT.

WHILE DEVTBL(R5) NE #END DO

026527 002536 177777

LET BTPT := BTADDR(R5)

LET #BTPT := #0

LET STREAM :B= #255.

```

TSTB HERE
BNE 50460$
MOV #TAPCAP, -(SP)
MOV #1, -(SP)
MOV SP, R0
TRAP C$PNTF
ADD #4, SP
    
```

```

INCB HERE
50460$:
;CLEAR THE RANDOM OPERATIONS FLAG.
CLRB RANDOM
;CLEAR THE EXPECT BOT FLAG.
CLRB EXPBOT
;SET CHARACTERISTICS.
;SET REWIND COMMAND IN BUFFER.
;CLEAR THE START FLAG.
CLRB STAF LG
;PLACE END FLAG IN SEQUENCE TABLE.
MOV #END, (R1)
;REWIND ALL UNITS.
    
```

```

;WHILE THERE ARE MORE DEVICES:
50461$:
CMP DEVTBL(R5), #END
BEQ 50462$
    
```

```

;CLEAR BAD SPOT COUNTER
MOV BTADDR(R5), BTPT
;START FROM BOT
CLR #BTPT
;SET FLAG - WE'RE GOING TO STREAM
MOVB #255., STREAM
    
```

5421	026170			LET R1 := #CMDSEQ	; SETUP SEQUENCE TABLE ADDRESS
	026170	012701	003542		MOV #CMDSEQ,R1
5422	026174			LET (R1) := #WRT	; WRITE COMMAND
	026174	012721	104005		MOV #WRT,(R1)
5423	026200			LET (R1) := #DATCNT	; 4096-BYTE RECORD LENGTH.
	026200	012721	010000		MOV #DATCNT,(R1)
5424	026204			LET (R1) := #9000.	; WRITE 9000 RECORDS.
	026204	012721	021450		MOV #9000,(R1)
5425	026210			LET (R1) := #5	; GENERATE AND WRITE PATTERN 5.
	026210	012721	000005		MOV #5,(R1)
5426	026214			LET (R1) := #END	; SET END OF SEQUENCE TABLE.
	026214	012711	177777		MOV #END,(R1)
5427					
5428	026220			LET R1 := #CMDSEQ	; SEQ. TABLE ADDRESS FOR SUBR. 'SETUP'.
	026220	012701	003542		MOV #CMDSEQ,R1
5429	026224	004737	007706	JSR PC, SETUP	; SETUP THE COMMAND TABLE
5430	026230			LET R5SAVE := R5	; SAVE R5, EM?
	026230	010537	003460		MOV R5,R5SAVE
5431					
5432	026234			WHILE NCNT LT NCNT1 DO	; WHILE MORE RECORDS SHOULD BE WRITTEN:
	026234				50463\$:
	026234	023737	003420		CMP NCNT,NCNT1
	026242	002022	003422		BGE 50464\$
5433					
5434	026244	004737	007600	JSR PC, CMDAC	; SAVE ASCII COMMAND FOR ERROR MESSAGE
5435	026250	004737	010654	JSR PC, EXECUTE	; ISSUE COMMAND TO UNIT.
5436	026254	004737	011274	JSR PC, GOWAIT	; GO WAIT FOR DONE TO SET
5437	026260			IF DEVTBL(R5) EQ #NINUSE THEN	
	026260	026527	002536		CMP DEVTBL(R5),#NINUSE
	026266	001005	177774		BNE 50465\$
5438	026270			LET NCNT := NCNT1 - #1	
	026270	013737	003422		MOV NCNT1,NCNT
	026276	005337	003420		DEC NCNT
5439	026302			ENDIF	
	026302				50465\$:
5440	026302			LET NCNT := NCNT + #1	; UPDATE THE RECORD COUNT
	026302	005237	003420		INC NCNT
5441	026306			ENDDO	; END OF RECORD 'DO' LOOP
	026306	000752			BR 50463\$
	026310				50464\$:
5442	026310			LET R1 := #CMDSEQ	; RESET R1 TO TOP OF TABLE
	026310	012701	003542		MOV #CMDSEQ,R1
5443	026314	004737	006724	JSR PC, SETRW	; ISSUE REWIND
5444	026320			LET (R1) := #END	; PLACE END FLAG IN SEQUENCE TABLE
	026320	012711	177777		MOV #END,(R1)
5445	026324			LET -(SP) := L\$LUN	; SAVE THE CURRENT LUN
	026324	013746	002074		MOV L\$LUN,-(SP)
5446	026330	004737	006744	JSR PC, EXALL	; DO REWIND, NOW
5447	026334			LET R5 := R5SAVE	; RESTORE R5
	026334	013705	003460		MOV R5SAVE,R5
5448	026340			LET L\$LUN := (SP)	; RESTORE THE CURRENT LUN
	026340	012637	002074		MOV (SP),L\$LUN
5449	026344			IF DEVTBL(R5) NE #NINUSE THEN	
	026344	026527	002536		CMP DEVTBL(R5),#NINUSE
	026352	001431	177774		BEQ 50466\$
5450	026354			LET R1 := #CMDSEQ	; TOP OF COMMAND TABLE
	026354	012701	003542		MOV #CMDSEQ,R1

5451	026360			LET (R1) := @RDF	;READ FORWARD COMMAND		
	026360	012721	104001		MOV	@RDF,(R1)	
5452	026364			LET (R1) := @DATCNT	;4096 BYTE RECORDS		
	026364	012721	010000		MOV	@DATCNT,(R1)	
5453	026370			LET (R1) := @9000.	;9000 ITERATIONS		
	026370	012721	021450		MOV	@9000.,(R1)	
5454	026374			LET (R1) := @5	;READ PATTERN NUMBER 5		
	026374	012721	000005		MOV	@5,(R1)	
5455	026400			LET (R1) := @END	;TABLE TERMINATOR		
	026400	012721	177777		MOV	@END,(R1)	
5456	026404			LET R1 := @CMDSEQ	;TOP OF TABLE, AGAIN!		
	026404	012701	003542		MOV	@CMDSEQ,R1	
5457	026410	004737	007706	JSR PC, SETUP	;SET UP THE COMMAND TABLE		
5458	026414			LET VFYFLG :B= @1	;ALLOW THE DATA VERIFY		
	026414	112737	000001	003522	MOVB	@1,VFYFLG	
5459	026422			LET R5SAVE := R5	;SAVE R5		
	026422	010537	003460		MOV	R5,R5SAVE	
5460	026426	004737	016142	JSR PC, VFYDAT	;GO OFF AND CHECK READ OPERATIONS		
5461	026432			LET R5 := R5SAVE	;RESTORE R5		
	026432	013705	003460		MOV	R5SAVE,R5	
5462	026436			ENDIF			
	026436					50466\$:	
5463	026436			LET NCNT := @0	;CLEAR RECORD COUNT		
	026436	005037	003420		CLR	NCNT	
5464	026442			LET VFYFLG :B= @0	;CLEAR VERIFY FLAG		
	026442	105037	003522		CLRB	VFYFLG	
5465	026446			LET EXPBOT :B= @0	;CLEAR EXPECT BOT FLAG.		
	026446	105037	003520		CLRB	EXPBOT	
5466	026452	004737	017346	JSR PC, NEXTU	;GET NEXT UNIT TO TEST (UUT).		
5467							
5468	026456			ENDDO	;END OF UUT LOOP		
	026456	000630			BR	50461\$	
	026460					50462\$:	
5469							
5470	026460			LET STREAM :B= @0	;CLEAR STREAMING FLAG FOR OTHER TESTS.		
	026460	105037	003532		CLRB	STREAM	
5471	026464			LET ALLEOT :B= @0	;RESET THE UNITS @ EOT STATUS		
	026464	105037	003531		CLRB	ALLEOT	
5472	026470			LET RPTFLG :B= @1	;REQUEST A REPORT		
	026470	112737	000001	003523	MOVB	@1,RPTFLG	
5473	026476			LET R1 := @CMDSEQ	;TOP OF TABLE		
	026476	012701	003542		MOV	@CMDSEQ,R1	
5474	026502	004737	006724	JSR PC, SETRW	;STORE THE REWIND COMMAND		
5475	026506			LET (R1) := @END	;TERMINATOR		
	026506	012711	177777		MOV	@END,(R1)	
5476	026512	004737	006744	JSR PC, EXALL	;REWIND AND REPORT STATUS FOR ALL UNITS		
5477							
5478	026516			EXIT TST	;EXIT TEST		
	026516	104432				TRAP	C\$EXIT
	026520	000002				.WORD	L10035-
5479				.EVEN	;JUST IN CASE.		
5480							
5481	026522			ENDTST			
	026522						
	026522	104401		L10035:		TRAP	C\$ETST

```

5483
5484 .SBTTL TEST 4: WRITE COMPATABILITY/WRITE UTILITY.
5485
5486
5487 ;**
5488 ; TEST TO WRITE RECORDS FROM BOT TO EOT.
5489 ;--
5490 026524          BGNTST
5491 026524          T4::
5492 026524          LET RANDOM :B= #1          ;SET THE RANDOM OPERATIONS FLAG.
5493 026532 112737 000001 003521          MOVB #1,RANDOM
5494 026532 105037 003520          LET EXPBOT :B= #0          ;CLEAR EXPECT BOT FLAG.
5495 026536 012702 010000          LET R2 := #DATCNT - #1          ;SET UP THE RECORD LENGTH MASK.
5496 026542 005302          DEC R2
5497 026544          LET LENMSK := COMP R2          ;ALLOW MAXIMUM BUFFER.
5498 026544 010237 003436          MOV R2,LENMSK
5499 026544 005137 003436          COM LENMSK
5500 026554 004737 006700          JSR PC,SETCH          ;CMD 1 = SET CHARACTERISTIC.
5501 026560 004737 006724          JSR PC,SETRW          ;CMD2=REWIND
5502 026564 105037 003534          LET STAFLG :B= #0          ;CLEAR START FLAG
5503 026570          REPEAT          ;REPEAT TO EOT.
5504 026570          WHILE R1 LT #SEQEND DO          ;WHILE THERE IS MORE ROOM IN SEQ TABLE:
5505 026570          ;
5506 026570 020127 003632          ;
5507 026574 002003          ;
5508 026576 004737 025766          JSR PC,RANWR          ;STORE A WRITE CMD IN SEQUENCE TABLE.
5509 026602          ENDDO
5510 026602 000772          ;
5511 026604          BR 50470$
5512 026604          LET (R1) := #END          ;STORE END OF SEQUENCE CODE IN TABLE.
5513 026604 012711 177777          MOV #END,(R1)
5514 026610 004737 006744          JSR PC,EXALL          ;EXECUTE ALL CMDs IN SEQ TBL ON UNITS.
5515 026614          LET R1 := #CMDSEQ          ;INIT SEQ TBL POINTER.
5516 026620 012701 003542          UNTIL R2 NE #0          ;REPEAT UNTIL EOT IS REACHED
5517 026620 005702          ;
5518 026622 001762          ;
5519 026624          LET ALLEOT :B= ALLEOT + #1          ;SET ALL UNITS @ EOT FLAG
5520 026624 105237 003531          INCB ALLEOT
5521 026630          NOP
5522 026632          NOP
5523 026634          NOP
5524 026636 004737 027640          JSR PC,TSWEOT          ;WRITE ONE RECORD BEYOND EOT ON ALL UNITS
5525          ;SO THAT SHORTER READ STOP DISTANCE
5526          ;SHALL POSITION HEAD IN CLEAN IRG GAP
5527          ;READ REV THAT EXTRA REC TO RE-POSITION TAPE
5528          ;CLEAR ALL UNITS @ EOT FLAG
5529          CLR B ALLEOT
5530          ;STORE REWIND IN SEQ TBL.
5531          LET (R1) := #END          ;STORE END IN SEQ TBL.
5532          MOV #END,(R1)
5533          JSR PC,EXALL          ;EXECUTE REWIND CMD ON ALL UNITS

```

5519							
5520	026662		EXIT	TST			
	026662	104432				TRAP	C#EXIT
	026664	000002				.WORD	L10036-
5521							
5522			.EVEN				
5523							
5524	026666		ENDTST				
	026666		L10036:				
	026666	104401				TRAP	C#ETST
5525							

```

5527
5528
5529
5530
5531
5532
5533
5534 026670
026670
5535
5536 026670 112737 000001 003521
026670
5537 026676 112737 000001 003520
026676
5538 026704 004737 006700
5539 026710 004737 006724
5540 026714
026714 105037 003534
5541 026720
026720 012721 104001
5542 026724
026724 012721 010000
5543 026730
026730 012721 077777
5544 026734
026734 012721 000007
5545 026740
026740 012711 177777
5546 026744 004737 006744
5547
5548
5549
5550
5551
5552
5553
5554
5555 026750
026750 105037 003531
5556 026754
026754 012701 003542
5557 026760 004737 006724
5558 026764
026764 012721 177777
5559 026770 004737 006744
5560
5561 026774
026774 104432
026776 000002
5562
5563
5564
5565 027000
027000
027000 104401

.SBTTL TEST 5: READ COMPATABILITY/READ UTILITY.
; **
; TEST TO READ ENTIRE TAPE FORWARD AND REVERSE.
; --
BGNTST
T5::
LET RANDOM :B= #1 ;SET THE RANDOM OPERATIONS FLAG.
MOV #1,RANDOM
LET EXPBOT :B= #1 ;SET EXPECT BOT FLAG.
MOV #1,EXPBOT
JSR PC,SETCH ;CMD 1 = SET CHARACTERISTIC.
JSR PC,SETRW ;CMD2=REWIND.
LET STAF LG :B= #0 ;CLEAR START FLAG
CLRB STAF LG
LET (R1)+ := #RDF ;CMD3 = READ FORWARD.
MOV #RDF,(R1)+
LET (R1)+ := #DATCNT ;SET LENGTH TO MAX FOR UNKNOWN LENGTHS.
MOV #DATCNT,(R1)+
LET (R1)+ := #77777 ;SET RECORD COUNT TO MAX FOR WHOLE TAPE.
MOV #77777,(R1)+
LET (R1)+ := #RANP ;PATTERN = RANDOM.
MOV #RANP,(R1)+
LET (R1) := #END ;STORE END OF SEQUENCE CODE IN TABLE.
MOV #END,(R1)
JSR PC,EXALL ;EXECUTE ALL CMDS IN SEQ TBL ON ALL UNITS.
LET ALLEOT :B= ALLEOT + #1 ;FLAG TO ALLOW ALL UNITS AT EOT TO READ REV
;
; LET R1 := #CMDSEQ ;INIT CMD SEQ TBL POINTER.
;
; LET (R1)+ := #RDR ;CMD1 = READ REVERSE.
;
; LET (R1)+ := #DATCNT ;SET LENGTH TO MAX FOR UNKNOWN LENGTHS.
;
; LET (R1)+ := #77777 ;RECORD COUNT = MAX FOR WHOLE TAPE.
;
; LET (R1)+ := #RANP ;PATTERN = RANDOM.
;
; LET (R1) := #END ;STORE END OF SEQUENCE CODE IN TABLE.
;
; JSR PC,EXALL ;GO EXECUTE READ REV. OF ENTIRE TAPE.
;
; LET ALLEOT :B= #0 ;CLEAR ALL UNITS @ EOT FLAG
CLRB ALLEOT
LET R1 := #CMDSEQ ;TOP OF COMMAND TABLE
MOV #CMDSEQ,R1
JSR PC,SETRW ;ISSUE A REWIND COMMAND (TWR)
LET (R1)+ := #END ;TERMINATOR
MOV #END,(R1)+
JSR PC,EXALL ;ISSUE THE REWIND
EXIT TST
TRAP C$EXIT
.WORD L10037-.
.EVEN
ENDTST
L10037:
TRAP C$ETST
    
```

```

5567
5568 .SBTTL TEST 6: EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.
5569
5570
5571 ;++
5572 ; TEST TO EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.
5573 ;--
5574 027002          BGNTST
5575 027002          T6::
5576 027002 105037 003521          LET RANDOM :B= #0          ;CLEAR RANDOM MODE FLAG.
5577 027006 112737 000001 003520          LET EXPBOT :B= #1          ;CLEAR RANDOM MODE FLAG.
5578 027014 113737 002213 003526          LET IRE :B= PIRE          ;SET EXPECT BOT FLAG.
5579 027022 004737 006700          JSR PC,SETCH          ;MOVE INHIBIT RFC ERROR REPORT FLAG.
5580 027026 013737 002220 003544          LET CMDSEQ+2 := CHAR          ;MOVE CHAR CODE FROM P TBL TO SEQ TBL.
5581 027034 012702 002222          LET R2 := #CMDD          ;R2 POINTS TO CMD2 IN SOFT P TABLE.
5582 027040 004737 027616          JSR PC,PTCMDS          ;MOVE CMD 2 FROM P TBL TO SEQ TBL.
5583 027044 004737 027616          JSR PC,PTCMDS          ;MOVE CMD 3 FROM P TBL TO SEQ TBL.
5584 027050 004737 027616          JSR PC,PTCMDS          ;MOVE CMD 4 FROM P TBL TO SEQ TBL.
5585 027054 004737 027616          JSR PC,PTCMDS          ;MOVE CMD 5 FROM P TBL TO SEQ TBL.
5586 027060 004737 027616          JSR PC,PTCMDS          ;MOVE CMD 6 FROM P TBL TO SEQ TBL.
5587 027064 004737 027616          JSR PC,PTCMDS          ;MOVE CMD 7 FROM P TBL TO SEQ TBL.
5588 027070 004737 027616          JSR PC,PTCMDS          ;MOVE END CMD FROM P TBL TO SEQ TBL.
5589 027074 005037 003450          LET JLOOP := #0          ;CLEAR JMP CMD LOOP COUNT.
5590 027100 105037 003534          LET STAF LG :B= #0          ;CLEAR START FLAG
5591 027104 012701 003542          LET R1 := #CMDSEQ          ;INIT SEQUENCE TABLE POINTER.
5592
5593          ;BRJMP=0          ;ENABLE JMP SUBSTITUTION FOR BR, IF NECESSARY.
5594
5595 027110          3$: WHILE (R1) NE #END DO          ;WHILE THERE ARE CMDS LEFT IN SEQUENCE TBL:
5596 027110 021127 177777          ;50472$:
5597 027114 001002          CMP (R1),#END
5598 027116 000137 027560          BNE .+6
5599 027130 062701 000002          JMP 50473$
5600 027134 012137 003452          ;IS THIS A JUMP CMD?
5601 027140 022137 003450          ;BR IF NOT.
5602 027144 001003          ;POINT TO BRF.
5603 027146 062701 000002          ADD #2,R1
5604 027152 000756          ;SAVE BRF (LOCATION).
5605 027154 005237 003450          ;HAS LOOP COUNT BE SATISFIED?
5606 027160 012701 003542          ;IF NOT, JMP AGAIN.
5607 027164 005337 003452          ;IF SO, ADJUST SEQ POUNTER
5608          ;AND GO TO NEXT COMMAND.
5609          ;UPDATE THE LOOP COUNT.
5610          ;INIT CMD SEQ TABLE POINTER.
5611          ;DECR LOCATION COUNTER.

```

5607	027170	001747		BEQ 3\$;IF THIS IS THE RIGHT LOCATION TO JMP TO, GO SET UP.
5608	027172			LET R1 := R1 + #10					;IF NOT, UPDATE SEQ POINTER TO NEXT CMD.
	027172	062701	000010						ADD #10,R1
5609	027176	000772		BR 2\$;DO IT AGAIN.
5610	027200	022711	000020	6\$: CMP #DLY.C,(R1)					;DELAY?
5611	027204	001026		BNE 4\$;BR IF NOT.
5612	027206			LET R1 := R1 + #4					;R1 = LOCATION OF N COUNT.
	027206	062701	000004						ADD #4,R1
5613	027212			LET TIME2 := (R1)					;SAVE N COUNT.
	027212	011137	003446						MOV (R1),TIME2
5614	027216			7\$: DELAY 10.					;GO TO SUPER-WAIT 1 MSEC.
	027216	012727	000012						MOV #10.,(PC)+
	027222	000000							.WORD 0
	027224	013727	002116						MOV L\$DLY,(PC)+
	027230	000000							.WORD 0
	027232	005367	177772						DEC -6(PC)
	027236	001375							BNE .-4
	027240	005367	177756						DEC -22(PC)
	027244	001367							BNE .-20
5615	027246	005337	003446	DEC TIME2					
5616	027252	001361		BNE 7\$					
5617	027254			LET R1 := R1 + #4					;POINT TO NEXT CMD.
	027254	062701	000004						ADD #4,R1
5618	027260	000713		BR 3\$;GO CHECK NEXT CMD.
5619	027262	004737	007706	4\$: JSR PC,SETUP					;GO SETUP THE COMMAND BLOCK.
5620	027266			WHILE NCNT LT NCNT1 DO					;WHILE THERE ARE RECORDS REMAINING:
	027266								50474\$:
	027266	023737	003420						CMP NCNT,NCNT1
	027274	002402							BLT .+6
	027276	000137	027550						JMP 50475\$
5621	027302	004737	007600	JSR PC,CMDAC					;STORE CMD ASCII IN ERROR MSG.
5622	027306	004737	007240	JSR PC,EXSUB					;ISSUE CMD TO ALL,AWAIT INTS,CHECK STATUS.
5623	027312			IF CMDWRD EQ #GES THEN					;IF CMD IS GET STATUS THEN:
	027312	023727	003426						100017 CMP CMDWRD,#GES
	027320	001402							BEQ .+6
	027322	000137	027332						JMP 50476\$
5624	027326	004737	017624	JSR PC,PRXST					;PRINT EXTENDED STATUS REGISTERS.
5625	027332			ENDIF					
	027332								50476\$:
5626	027332	004737	017706	JSR PC,CKHAE					;CHECK HALT AFTER EACH CMD FLAG.
5627	027336			LET R2 := #1					;SET ALL UNITS AT BOT/EOT.
	027336	012702	000001						MOV #1,R2
5628	027342	004737	017300	JSR PC,FIRSTU					;FIND FIRST UNIT.
5629	027346			WHILE DEVTBL(R5) NE #END DO					;WHILE THERE ARE MORE UNITS:
	027346								50477\$:
	027346	026527	002536						CMP DEVTBL(R5),#END
	027354	001002							BNE .+6
	027356	000137	027454						JMP 50500\$
5630	027362			IF #MOD.CO SETIN CMDWRD THEN					;IF CMD IS REVERSE THEN:
	027362	032737	000400						BIT #MOD.CO,CMDWRD
	027370	001002	003426						BNE .+6
	027372	000137	027420						JMP 50501\$
5631	027376			IF #X0.BOT NOTSETIN EOTFLG(R5) THEN					;IF NOT AT BOT THEN:
	027376	032765	000002						BIT #X0.BOT,EOTFLG(R5)
	027404	001402	003506						BEQ .+6
	027406	000137	027414						JMP 50502\$
5632	027412			LET R2 := #C					;CLEAR EOT/BOT FLAG.

```

5633 027412 005002                                CLR    R2
027414                                ENDIF
5634 027414                                ELSE
027414 000137 027446                                ;ELSE IF CMD IS NOT REVERSE:
027420                                JMP    50503$
5635 027420                                IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN
027420 032765 000001 003506                                BIT    #X0.EOT,EOTFLG(R5)
027426 001406                                BEQ    50504$
027430 032737 000001 003426                                BIT    #CMD.CO,CMDWRD
027436 001402                                BEQ    .+6
027440 000137 027446                                JMP    50505$
027444                                50504$:
5636                                ;IF NOT AT EOT OR NOT A MOTION CMD THEN:
5637 027444                                LET R2 := #0
027444 005002                                ;CLEAR EOT/BOT FLAG.
5638 027446                                CLR    R2
027446                                50505$:
5639 027446                                ENDIF
027446                                50503$:
5640 027446 004737 017346                                JSR PC,NEXTU                                ;FIND NEXT UNIT
5641 027452                                ENDDO
027452 000735                                ;
027454                                BR    50477$
5642 027454                                IF R2 EQ #1 THEN                                50500$:
027454 020227 000001                                ;IF ALL UNIT ARE AT EOT/BOT THEN:
027460 001402                                CMP    R2,#1
027462 000137 027530                                BEQ    .+6
5643 027466                                LET NCNT1 := NCNT + #1                                ;FORCE TERMINATION OF COMMAND.
027466 013737 003420 003422                                MOV    NCNT,NCNT1
027474 005237 003422                                INC    NCNT1
5644 027500                                LET ALLEOT :B= ALLEOT + #1 ;FLAG ALL UNITS AT EOT/BOT TO ALLOW VERIFY OF DATA
027500 105237 003531                                INCB  ALLEOT
5645 027504                                IF CMDLG EQ #2 THEN                                ;WHEN WRITING IS CURRENT COMMAND
027504 023727 003434 000002                                CMP    CMDLG,#2
027512 001402                                BEQ    .+6
027514 000137 027524                                JMP    50507$
5646 027520 004737 027640                                JSR PC,T5WEOT                                ;GO WRITE/READ REV ONE RECORD BEYOND EOT
5647 027524                                ENDDO
027524                                50507$:
5648 027524                                ELSE
027524 000137 027534                                JMP    50510$
027530                                50506$:
5649 027530                                LET ALLEOT :B= #0 ;WHEN NOT ALL @EOT, CLEAR FLAG
027530 105037 003531                                CLR    ALLEOT
5650 027534                                ENDIF
027534                                50510$:
5651 027534                                LET NCNT := NCNT + #1                                ;UPDATE RECORD COUNT.
027534 005237 003420                                INC    NCNT
5652 027540                                LET PCMDWD := CMDWRD                                ;SAVE PREVIOUS COMMAND WORD.
027540 013737 003426 003432                                MOV    CMDWRD,PCMDWD
5653 027546                                ENDDO
027546 000647                                BR    50474$
027550                                50475$:
5654 027550 004737 016142                                JSR PC,VFYDAT                                ;IF LAST CMD WAS A WRITE VERIFY, THEN GO
5655                                ;VERIFY THE LAST N RECORDS OF DATA.
5656

```

HARDWARE TESTS MACRO M1200 21-MAR-84 16:45 PAGE 112-3
 TEST 6: EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.

SEQ 0167

```

5657 027554          ENDDO
      027554 000137 027110
      027560
5658
5659          177777
5660 027560          $BRJMP=-1 ;TURN OFF JMP SUBSTITUTION (SPMACJ CONTROL, ONLY).
      027560 105037 003531          LET ALLEOT :B= #0 ;NOT AT EOT
5661 027564          LET RPTFLG :B= #1 ;REQUEST A REPORT          CLRB ALLEOT
      027564 112737 000001 003523          LET R1 := #CMDSEQ          ;SET UP THE COMMAND TABLE ADDRESS          MOVB #1,RPTFLG
5662 027572          LET R1 := #CMDSEQ          ;STORE REWIND COMMAND          MOV #CMDSEQ,R1
      027572 012701 003542          JSR PC,SETRW          ;TERMINATE THE TABLE
5663 027576 004737 006724          LET (R1) := #END          ;ISSUE THE REWINDS          MOV #END,(R1)
5664 027602          JSR PC,EXALL
      027602 012711 177777
5665 027606 004737 006744
5666
5667
5668 027612          EXIT TST
      027612 104432
      027614 000130
5669

```

TRAP C\$EXIT
 .WORD L10040-


```

5671
5672 ; SUBROUTINE TO MOVE A COMMAND FROM THE SOFTWARE P TABLE TO
5673 ; THE COMMAND SEQUENCE TABLE.
5674 ; INPUTS: R2 = POINTER TO SOFT "P" TABLE
5675 ; OUTPUTS:
5676 ; REGISTERS: R3.
5677 ; CALLS:
5678
5679 PTCMDS: LET R3 := (R2)+ - #1 SHIFT +1 ;R3 = COMMAND TABLE INDEX.
          027616 012203 MOV (R2)+,R3
          027616 005303 DEC R3
          027622 006303 ASL R3
5680 027624 LET (R1)+ := CMTDBL(R3) ;MOVE COMMAND WORD.
          027624 016321 003644 MOV CMTDBL(R3),(R1)+
5681 027630 LET (R1)+ := (R2)+ ;MOVE # OF BYTES.
          027630 012221 MOV (R2)+,(R1)+
5682 027632 LET (R1)+ := (R2)+ ;MOVE # OF OPERATIONS.
          027632 012221 MOV (R2)+,(R1)+
5683 027634 LET (R1)+ := (R2)+ ;MOVE PATTERN CODE.
          027634 012221 MOV (R2)+,(R1)+
5684 027636 000207 RTS PC
5685
5686 ; SUBROUTINE TO WRITE THEN READ REVERSE ONE RECORD BEYOND EOT
5687 ; INPUTS:
5688 ; OUTPUTS:
5689 ; REGISTERS:
5690 ; CALLS: CMDAC,EXSUB,CKHAE
5691
5692 027640 000240 TSWEOT: NOP
5693 027642 000240 NOP
5694 027644 004737 007240 JSR PC,EXSUB ;WRITE ONE RECORD BEYOND EOT
5695 027650 004737 017706 JSR PC,CKHAE ;SO THAT READ SHORTER STOP DISTANCE
5696 ;SHALL POSITION HEAD IN CLEAN IRG GAP
5697 027654 LET PCMDWD := CMDWRD ;REPOSITION TAPE
          027654 013737 003426 003432 MOV CMDWRD,PCMDWD
5698 027662 LET CMDWRD := #RDR ;BEFORE EXTRA RECORD
          027662 012737 104401 003426 MOV #RDR,CMDWRD
5699 027670 LET CMDLG := #4 ;BY READING REVERSE
          027670 012737 000004 003434 MOV #4,CMDLG
5700 027676 LET CMDPKT := CMDWRD CLR.BY #BRF.C
          027676 013737 003426 002314 MOV CMDWRD,CMDPKT
          027704 042737 004000 002314 BIC #BRF.C,CMDPKT
5701 027712 LET CMDSAV := CMDPKT ;THAT RECORD TO ALLOW
          027712 013737 002314 003430 MOV CMDPKT,CMDSAV
5702 027720 LET CMDPKT+CP.ADL := DATARD ;NEXT COMMAND IN THE
          027720 013737 003416 002316 MOV DATARD,CMDPKT+CP.ADL
5703 027726 JSR PC,CMDAC ;TABLE TO BE EXECUTED
5704 027732 JSR PC,EXSUB
5705 027736 JSR PC,CKHAE
5706 027742 000207 RTS PC
5707 .EVEN
5708 027744 ENDTST
          027744 L10040:
          027744 104401 TRAP C$ETST
5709 027746 ENDMOD

```

```

5712          .TITLE PARAMETER CODING
5723
5724          .SBTTL  HARDWARE PARAMETER CODING SECTION
5733
5734 027746          BGNMOD
5735
5736          ;**
5737          ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
5738          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
5739          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
5740          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
5741          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
5742          ; WITH THE OPERATOR.
5743          ;--
5744
5745 027746          BGNHRD
5746 027746 000024
5747 027750          L$HARD::
5748
5749          GPRMA  TS4ADR,0,0,160002,177564,YES
5750
5751          GPRMD  TS4VCT,2,0,777,60,776,YES
5752
5753          EXIT HRD
5754
5755 027772          .NLIST BEX
5756 027774          .ASCIZ /TSSR ADDRESS/
5757 030011          .ASCIZ /VECTOR/
5758          .LIST BEX
5759          .EVEN
5760
5761          ENDHRD
5762
5763          L10041:
5764
5765          .WORD L10041-L$HARD/2
5766
5767          .WORD T$CODE
5768          .WORD TS4ADR
5769          .WORD T$LLOLIM
5770          .WORD T$HILIM
5771
5772          .WORD T$CODE
5773          .WORD TS4VCT
5774          .WORD 777
5775          .WORD T$LLOLIM
5776          .WORD T$HILIM
5777
5778          .WORD T$CODE
5779
5780          .EVEN
5781
5782          030020
5783
5784          L10041:
5785
5786          .EVEN
    
```

```

5772
5773
5774
5775
5776
5777
5778
5779
5780
5781
5782
5783 030020          .SBTTL  SOFTWARE PARAMETER CODING SECTION
      030020 000531
      030022
5784
5791
5792 030022          ;**
      030022 000130          ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
      030024 030600          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
      030026 000001          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
5793 030030          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
      030030 000130          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
      030032 030617          ; WITH THE OPERATOR.
      030034 000400          ; --
5794 030036          BGNSFT
      030036 001130          L$SOFT::
      030040 030646          .WORD L10042-L$SOFT/2
      030042 000001
5795 030044          GPRML  CLRM,0,1,YES
      030044 001130          .WORD  T$CODE
      030046 030672          .WORD  CLRM
      030050 000400          .WORD  1
5796 030052          GPRML  RRVN,0,400,YES
      030052 002130          .WORD  T$CODE
      030054 030714          .WORD  RRVN
      030056 000001          .WORD  400
5797 030060          GPRML  HAEM,2,1,YES
      030060 004024          .WORD  T$CODE
      030062 002130          .WORD  HAEM
      030064 030735          .WORD  1
      030066 000400          GPRML  RCVERM,2,400,YES
      030070          .WORD  T$CODE
      030070 003130          .WORD  RCVERM
      030072 030762          .WORD  400
      030074 000001          GPRML  IRECM,4,1,YES
      030076          .WORD  T$CODE
      030076 003130          .WORD  IRECM
      030100 031005          .WORD  1
      030102 000400          XFERT  NEXTSP
      030104          .WORD  T$CODE
      030104 004130          GPRML  BADTM,4,400,YES
      030106 031040          .WORD  T$CODE
      030110 000001          .WORD  BADTM
      030112          .WORD  400
5800 030070          NEXTSP: GPRML  DINTM,6,1,YES
      030070 003130          .WORD  T$CODE
      030072 030762          .WORD  DINTM
      030074 000001          .WORD  1
5801 030076          GPRML  IREM,6,400,YES
      030076 003130          .WORD  T$CODE
      030100 031005          .WORD  IREM
      030102 000400          .WORD  400
5802 030104          GPRML  RAMM,10,1,YES
      030104 004130          .WORD  T$CODE
      030106 031040          .WORD  RAMM
      030110 000001          .WORD  1
5803 030112          GPRML  EWPT,12,1,YES
      030112 005130          .WORD  T$CODE

```

	030114	031064			.WORD	EWPT
	030116	000001			.WORD	1
5804	030120		GPRML	CHGM,12,400,YES		
	030120	005130			.WORD	T%CODE
	030122	031122			.WORD	CHGM
	030124	000400			.WORD	400
5805	030126		XFERF	ENDSP1		
	030126	127044			.WORD	T%CODE
5806	030130		G'RMD	CHARM,14,0,377,0,177,YES		
	030130	006032			.WORD	T%CODE
	030132	031141			.WORD	CHARM
	030134	000377			.WORD	377
	030136	000000			.WORD	T%LOLIM
	030140	000777			.WORD	T%HILIM
5807	030142		GPRMD	CMD2M,16,D,37,1,33,YES		
	030142	007052			.WORD	T%CODE
	030144	031166			.WORD	CMD2M
	030146	000037			.WORD	37
	030150	000001			.WORD	T%LOLIM
	030152	000033			.WORD	T%HILIM
5808	030154		GPRMD	BPCRM,20,D,-1,1,DATCNT,YES		
	030154	010052			.WORD	T%CODE
	030156	031174			.WORD	BPCRM
	030160	177777			.WORD	-1
	030162	000001			.WORD	T%LOLIM
	030164	010000			.WORD	T%HILIM
5809	030166		GPRMD	NUMBM,22,D,-1,1,77777,YES		
	030166	011052			.WORD	T%CODE
	030170	031206			.WORD	NUMBM
	030172	177777			.WORD	-1
	030174	000001			.WORD	T%LOLIM
	030176	077777			.WORD	T%HILIM
5810	030200		GPRMD	PATTM,24,D,17,0,10,YES		
	030200	012052			.WORD	T%CODE
	030202	031226			.WORD	PATTM
	030204	000017			.WORD	17
	030206	000000			.WORD	T%LOLIM
	030210	000010			.WORD	T%HILIM
5811	030212		GPRMD	CMD3M,26,D,37,1,33,YES		
	030212	013052			.WORD	T%CODE
	030214	031240			.WORD	CMD3M
	030216	000037			.WORD	37
	030220	000001			.WORD	T%LOLIM
	030222	000033			.WORD	T%HILIM
5812	030224		GPRMD	BPCRM,30,D,-1,1,DATCNT,YES		
	030224	014052			.WORD	T%CODE
	030226	031174			.WORD	BPCRM
	030230	177777			.WORD	-1
	030232	000001			.WORD	T%LOLIM
	030234	010000			.WORD	T%HILIM
5813	030236		GPRMD	NUMBM,32,D,-1,1,77777,YES		
	030236	015052			.WORD	T%CODE
	030240	031206			.WORD	NUMBM
	030242	177777			.WORD	-1
	030244	000001			.WORD	T%LOLIM
	030246	077777			.WORD	T%HILIM
5814	030250		GPRMD	PATTM,34,D,17,0,10,YES		

	030250	016052			.WORD	T\$CODE
	030252	031226			.WORD	PATM
	030254	000017			.WORD	17
	030256	000000			.WORD	T\$LOLIM
	030260	000010			.WORD	T\$HILIM
5815	030262		GPRMD	CMD4M,36,D,37,1,33,YES		
	030262	017052			.WORD	T\$CODE
	030264	031246			.WORD	CMD4M
	030266	000037			.WORD	37
	030270	000001			.WORD	T\$LOLIM
	030272	000033			.WORD	T\$HILIM
5816	030274		GPRMD	BPCRM,40,D,-1,1,DATCNT,YES		
	030274	020052			.WORD	T\$CODE
	030276	031174			.WORD	BPCRM
	030300	177777			.WORD	-1
	030302	000001			.WORD	T\$LOLIM
	030304	010000			.WORD	T\$HILIM
5817	030306		GPRMD	NUMBM,42,D,-1,1,77777,YES		
	030306	021052			.WORD	T\$CODE
	030310	031206			.WORD	NUMBM
	030312	177777			.WORD	-1
	030314	000001			.WORD	T\$LOLIM
	030316	077777			.WORD	T\$HILIM
5818	030320		GPRMD	PATM,44,D,17,0,10,YES		
	030320	022052			.WORD	T\$CODE
	030322	031226			.WORD	PATM
	030324	000017			.WORD	17
	030326	000000			.WORD	T\$LOLIM
	030330	000010			.WORD	T\$HILIM
5819	030332		GPRMD	CMD5M,46,D,37,1,33,YES		
	030332	023052			.WORD	T\$CODE
	030334	031254			.WORD	CMD5M
	030336	000037			.WORD	37
	030340	000001			.WORD	T\$LOLIM
	030342	000033			.WORD	T\$HILIM
5820	030344		GPRMD	BPCRM,50,D,-1,1,DATCNT,YES		
	030344	024052			.WORD	T\$CODE
	030346	031174			.WORD	BPCRM
	030350	177777			.WORD	-1
	030352	000001			.WORD	T\$LOLIM
	030354	010000			.WORD	T\$HILIM
5821	030356		GPRMD	NUMBM,52,D,-1,1,77777,YES		
	030356	025052			.WORD	T\$CODE
	030360	031206			.WORD	NUMBM
	030362	177777			.WORD	-1
	030364	000001			.WORD	T\$LOLIM
	030366	077777			.WORD	T\$HILIM
5822	030370		GPRMD	PATM,54,D,17,0,10,YES		
	030370	026052			.WORD	T\$CODE
	030372	031226			.WORD	PATM
	030374	000017			.WORD	17
	030376	000000			.WORD	T\$LOLIM
	030400	000010			.WORD	T\$HILIM
5823	030402		XFER	ENDSP2		
	030402	002004			.WORD	T\$CODE
5824	030404		ENDSP1: XFER	ENDSP		
	030404	075004			.WORD	T\$CODE

5825	030406		ENDSP2: GPRMD	CMD6M,56,D,37,1,33,YES			
	030406	027052				.WORD	T\$CODE
	030410	031262				.WORD	CMD6M
	030412	000037				.WORD	37
	030414	000001				.WORD	T\$LOLIM
	030416	000033				.WORD	T\$HILIM
5826	030420		GPRMD	BPCRM,60,D,-1,1,DATCNT,YES			
	030420	030052				.WORD	T\$CODE
	030422	031174				.WORD	BPCRM
	030424	177777				.WORD	-1
	030426	000001				.WORD	T\$LOLIM
	030430	010000				.WORD	T\$HILIM
5827	030432		GPRMD	NUMBM,62,D,-1,1,77777,YES			
	030432	031052				.WORD	T\$CODE
	030434	031206				.WORD	NUMBM
	030436	177777				.WORD	-1
	030440	000001				.WORD	T\$LOLIM
	030442	077777				.WORD	T\$HILIM
5828	030444		GPRMD	PATTM,64,D,17,0,10,YES			
	030444	032052				.WORD	T\$CODE
	030446	031226				.WORD	PATTM
	030450	000017				.WORD	17
	030452	000000				.WORD	T\$LOLIM
	030454	000010				.WORD	T\$HILIM
5829	030456		GPRMD	CMD7M,66,D,37,1,33,YES			
	030456	033052				.WORD	T\$CODE
	030460	031270				.WORD	CMD7M
	030462	000037				.WORD	37
	030464	000001				.WORD	T\$LOLIM
	030466	000033				.WORD	T\$HILIM
5830	030470		GPRMD	BPCRM,70,D,-1,1,DATCNT,YES			
	030470	034052				.WORD	T\$CODE
	030472	031174				.WORD	BPCRM
	030474	177777				.WORD	-1
	030476	000001				.WORD	T\$LOLIM
	030500	010000				.WORD	T\$HILIM
5831	030502		GPRMD	NUMBM,72,D,-1,1,77777,YES			
	030502	035052				.WORD	T\$CODE
	030504	031206				.WORD	NUMBM
	030506	177777				.WORD	-1
	030510	000001				.WORD	T\$LOLIM
	030512	077777				.WORD	T\$HILIM
5832	030514		GPRMD	PATTM,74,D,17,0,10,YES			
	030514	036052				.WORD	T\$CODE
	030516	031226				.WORD	PATTM
	030520	000017				.WORD	17
	030522	000000				.WORD	T\$LOLIM
	030524	000010				.WORD	T\$HILIM
5833	030526		GPRMD	CMD8M,76,D,37,1,33,YES			
	030526	037052				.WORD	T\$CODE
	030530	031276				.WORD	CMD8M
	030532	000037				.WORD	37
	030534	000001				.WORD	T\$LOLIM
	030536	000033				.WORD	T\$HILIM
5834	030540		GPRMD	BPCRM,100,D,-1,1,DATCNT,YES			
	030540	040052				.WORD	T\$CODE
	030542	031174				.WORD	BPCRM

030544 177777
 030546 000001
 030550 010000
 5835 030552
 030552 041052
 030554 031206
 030556 177777
 030560 000001
 030562 077777
 5836 030564
 030564 042052
 030566 031226
 030570 000017
 030572 000000
 030574 000010
 5837 030576
 5838 030576
 030576 120004

GPRMD NUMBM,102,D,-1,1,77777,YES

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

ENDSP:

XFER JMPMSG

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

5840									
5847						.NLIST	BEX		
5848	030600	103	114	105	CLRM:	.ASCIZ	/CLEAR COUNTERS/		
5849	030617	122	105	123	RRVM:	.ASCIZ	/RESET RANDOM VARIABLES/		
5850	030646	110	101	114	HAEM:	.ASCIZ	/HALT AFTER EACH CMD/		
5851	030672	120	122	111	RCVERM:	.ASCIZ	/PRINT SOFT ERRORS/		
5852	030714	111	116	110	IREFM:	.ASCIZ	/INHIBIT RECOVERY/		
5853	030735	102	101	104	BADTM:	.ASCIZ	/BAD TAPE SPOT DETECT/		
5854	030762	104	111	123	DINTM:	.ASCIZ	/DISABLE INTERRUPTS/		
5855	031005	111	116	110	IREFM:	.ASCIZ	/INHIBIT RFC ERROR REPORT/		
5856						.LIST	BEX		
5857						.EVEN			
5858	031036				JMPMSG:				
5859	031036				XFER	JMPMS2			
	031036	100004						.WORD	T\$CODE
5860	031040	103	117	116	RAMM:	.ASCIZ	/CONTROLLER RAM DUMP/		
	031043	124	122	117					
	031046	114	114	105					
	031051	122	040	122					
	031054	101	115	040					
	031057	104	125	115					
	031062	120	000						
5861	031064	105	116	101	EWPT:	.ASCIZ	/ENABLE EARLY WARNING MESSAGES/		
	031067	102	114	105					
	031072	040	105	101					
	031075	122	114	131					
	031100	040	127	101					
	031103	122	116	111					
	031106	116	107	040					
	031111	115	105	123					
	031114	123	101	107					
	031117	105	123	000					
5862	031122	103	110	101	CHGM:	.ASCIZ	/CHANGE CMD SEQ/		
	031125	116	107	105					
	031130	040	103	115					
	031133	104	040	123					
	031136	105	121	000					
5863	031141	103	110	101	CHARM:	.ASCIZ	/CHARACTERISTICS CODE/		
	031144	122	101	103					
	031147	124	105	122					
	031152	111	123	124					
	031155	111	103	123					
	031160	040	103	117					
	031163	104	105	000					
5864	031166	103	115	104	CMD2M:	.ASCIZ	"CMD/2"		
	031171	057	062	000					
5865	031174	102	122	106	BPCRM:	.ASCIZ	/BRF COUNT/		
	031177	040	103	117					
	031202	125	116	124					
	031205	000							
5866	031206	043	040	117	NUMBM:	.ASCIZ	/# OF OPERATIONS/		
	031211	106	040	117					
	031214	120	105	122					
	031217	101	124	111					
	031222	117	116	123					
	031225	000							
5867	031226	120	101	124	PATM:	.ASCIZ	/PATTERN/		


```

031231      124      105      122
031234      116      000
5868
5869
5870 031236      JMPMS2:
5871 031236      EXIT SFT
031236 023004      .WORD      T$CODE
5872
5873
5874 031240      103      115      104  CMD3M: .NLIST BEX
5875 031246      103      115      104  CMD4M: .ASCIZ "CMD/3"
5876 031254      103      115      104  CMD5M: .ASCIZ "CMD/4"
5877 031262      103      115      104  CMD6M: .ASCIZ "CMD/5"
5878 031270      103      115      104  CMD7M: .ASCIZ "CMD/6"
5879 031276      103      115      104  CMD8M: .ASCIZ "CMD/7"
5880
5881 031304      .LIST      BEX
                                ENDSFT
                                .EVEN
031304      L10042:
5882      ;*****
5883      ;*****
5884      ;*****
5885      ;      PATCH AREA
5886
5887 031304      PATCH:: .BLKW  64.
5888      ;*****
5889      ;*****
5890
5891 031504      LASTAD
                                .EVEN
031504 031520      .WORD T$FREE
031506 000004      .WORD T$SIZE
031510
5892 0315      L$LAST::
                                ENDMOD

```

PARAMETER CODING
HARD CODED P-TBL

MACRO M1200 21-MAR-84 16:45 PAGE 118

SEQ 0177

5894
5895
5896
5897
5898
5899
5900 031510
5901 031510
031510 000000
031512 000002
031514
5902 031514 172522
5903 031516 000224
5904 031520
031520
5905 031520
5906
5907 000001

.SBTTL HARD CODED P-TBL
; **
;DIAG IS PRE-PARAMETERIZED PER TBL
; **
BGNSETUP 1
BGNPTAB
L10043:
172522
224
ENDPTAB
L10045:
ENDSETUP
.END

.WORD 0
.WORD L10045-./2-1

PARAMETER CODING
SYMBOL TABLE

MACRO M1200 21-MAR-84 16:45 PAGE 118-1

SEQ 0178

ACK.C = 100000 G	BTMSG2 015021	CP.CMD= 000000 G	C#SVEC= 000037	EXSUB 007240 G
ADR = 000020 G	BTMSG3 015076	CP.CNT= 000006 G	C#TPRI= 000013	E#END = 002100
ALLEOT 003531 G	BTPT 003516 G	CRLF 005275 G	DATAWD 003416 G	E#LOAD= 000035
ASSEMB= 000010	BTRPT 020446	CRLFSP 005300 G	DATCNT= 010000 G	FATSM 004455 G
ATTNM 004417 G	BT0 003000 G	CTCC 003456 G	DATRAT 003412 G	FIRSTU 017300 G
AUDRPM 004727 G	BT1 003052 G	CVC.C = 040000 G	DEVTBL 002536 G	FMT.CO= 000040 G
AUTODM 023450	BT2 003124 G	C#AU = 000052	DFPTBL 002176 G	FMT.C1= 000100 G
BADTM 030735	BT3 003176 G	C#AUTO= 000061	DFTSCH= 000040 G	FTLCNT 003320 G
BADTSW 002211 G	BYTECO= 000403	C#BRK = 000022	DIA = 100006 G	FUNRM 004435 G
BFSEQ 024616	CHAR 002220 G	C#BSEG= 000004	DIABLK= 003414 G	F#AU = 000015
BFSEQ0 024642	CHARM 031141	C#BSUB= 000002	DIACNT= 000020 G	F#AUTO= 000020
BFSEQ1 024714	CHGFLG 002217 G	C#CEFG= 000045	DIAGMC= 000000	F#BGN = 000040
BFSEQ2 024726	CHGM 031122	C#CLCK= 000062	DINT 002212 G	F#CLEA= 000007
BFSEQ3 025020	CHKERR 012160 G	C#CLEA= 000012	DINTM 030762	F#DU = 000016
BFSEQ4 025072	CH.EAI= 000040 G	C#CLOS= 000035	DLY = 000020 G	F#END = 000041
BFSEQ5 025134	CH.ERI= 000020 G	C#CLP1= 000006	DLY.C = 000020 G	F#HARD= 000004
BFSEQ6 025206	CH.ESS= 000200 G	C#CVEC= 000036	DRI = 100013 G	F#HW = 000013
BFSEQ7 025240	CKDATA 016664 G	C#DCLN= 000044	DROPDM 004700 G	F#INIT= 000006
BFSEQ8 025272	CKDCNT 017274	C#DODU= 000051	DROPEP 003527 G	F#JMP = 000050
BFSEQ9 025324	CKDFF 017276	C#DRPT= 000024	DROPN 017622	F#MOD = 000000
BFSE10 025346	CKHAE 017706 G	C#DU = 000053	DROPU 017402 G	F#MSG = 000011
BGNFLG= 003464	CKHRTN 017774	C#EDIT= 000003	DROPUA 017532	F#PROT= 000021
BINC 015722	CLN = 101012 G	C#ERDF= 000055	DRORTN 017610	F#PWR = 000017
BIT0 = 000001 G	CLRERR 011656 G	C#ERHR= 000056	DTAERM 005460 G	F#RPT = 000012
BIT00 = 000001 G	CLRFLG 002204 G	C#ERRO= 000060	DTAER2 004761 G	F#SEG = 000003
BIT01 = 000002 G	CLRM 030600	C#ERSF= 000054	DTAER3 005030 G	F#SOFT= 000005
BIT02 = 000004 G	CMDAC 007600 G	C#ERSO= 000057	DTAER4 005072 G	F#SRV = 000010
BIT03 = 000010 G	CMDASC 003732 G	C#ESCA= 000010	DTAER5 005113 G	F#SUB = 000002
BIT04 = 000020 G	CMDD 002222 G	C#ESEG= 000005	EF.CON= 000036 G	F#SW = 000014
BIT05 = 000040 G	CMDLG 003434 G	C#ESUB= 000003	EF.NEW= 000035 G	F#TEST= 000001
BIT06 = 000100 G	CMDPKM 004164 G	C#ETST= 000001	EF.PWR= 000034 G	GCMDA 007654 G
BIT07 = 000200 G	CMDPKT 002314 G	C#EXIT= 000032	EF.RES= 000037 G	GENPAT 010264 G
BIT08 = 000400 G	CMDSAV 003430 G	C#GETB= 000026	EF.STA= 000040 G	GES = 100017 G
BIT09 = 001000 G	CMDSEQ 003542 G	C#GETW= 000027	EINC 015730	GETSTM 005241 G
BIT1 = 000002 G	CMDSE2 003552 G	C#GMAN= 000043	END = 177777 G	GIT 010642
BIT10 = 002000 G	CMDTBL 003644 G	C#GPHR= 000042	ENDERF= 003476	GOWAIT 011274 G
BIT11 = 004000 G	CMDWRD 003426 G	C#GPLO= 000030	ENDFLG= 003534	GSCPK 002324 G
BIT12 = 010000 G	CMD.CO= 000001 G	C#GPRI= 000040	ENDSP 030576	G#CNT0= 000200
BIT13 = 020000 G	CMD.C1= 000002 G	C#INIT= 000011	ENDSP1 030404	G#DELM= 000372
BIT14 = 040000 G	CMD.C2= 000004 G	C#INLP= 000020	ENDSP2 030406	G#DISP= 000003
BIT15 = 100000 G	CMD.C3= 000010 G	C#MANI= 000050	EOTFLG 003506 G	G#EXCP= 000400
BIT2 = 000004 G	CMD.C4= 000020 G	C#MEM = 000031	ERCVER 002207 G	G#HILI= 000002
BIT3 = 000010 G	CMD2M 031166	C#MSG = 000023	ERLOG 003472 G	G#LOLI= 000001
BIT4 = 000020 G	CMD3M 031240	C#OPEN= 000034	ERRREC 003475 G	G#NO = 000000
BIT5 = 000040 G	CMD4M 031246	C#PNTB= 000014	ERS = 100411 G	G#OFFS= 000400
BIT6 = 000100 G	CMD5M 031254	C#PNTF= 000017	ERSFLG 003533 G	G#OF SI= 000376
BIT7 = 000200 G	CMD6M 031262	C#PNTS= 000016	EVL = 000004 G	G#PRMA= 000001
BIT8 = 000400 G	CMD7M 031270	C#PNTX= 000015	EWMSG 012526	G#PRMD= 000002
BIT9 = 001000 G	CMD8M 031276	C#QIO = 000377	EWPRNT 002216 G	G#PRML= 000000
BOE = 000400 G	CMPDAT 003410 G	C#RDBU= 000007	EWPT 031064	G#RADA= 000140
BORERS 015146 G	CNT8GN= 002560	C#REFG= 000047	EWTRY 012632 G	G#RADB= 000000
BPCRM 031174	CNTEND= 003330	C#RESE= 000033	EXALL 006744 G	G#RADD= 000040
BRCPK 002330 G	CNTLEN= 000550 G	C#REVI= 000003	EXARTN 007236	G#RADL= 000120
BRFCNT 003424 G	CODELM 004054 G	C#RFLA= 000021	EXCRTN 011272	G#RADO= 000020
BRF.C = 004000 G	COUNTE= 050416	C#RPT = 000025	EXCUTE 010654 G	G#XFER= 000004
BTADDR 002550 G	CP.ADH= 000004 G	C#SEFG= 000046	EXPBOT 003520 G	G#YES = 000010
BTMSG1 014734	CP.ADL= 000002 G	C#SPRI= 000041		HAE 002206 G

PARAMETER CODING
SYMBOL TABLE

MACRO M1200 21-MAR-84 16:45 PAGE 118-2

SEQ 0179

HAEM	030646	L\$CLEA	023602 G	L10013	023446	OFLINM	005211 G	RANDOM	003521 G
HALTM	004124 G	L\$CO	002032 G	L10014	023642	ONEFIL	= 000001	RANP	= 000007 G
HELP	= 000000	L\$DEPO	002011 G	L10015	023706	OPFLAG	003540 G	RANRD	025712
HERE	003340	L\$DESC	002140 G	L10016	024002	OPP.C	= 020000 G	RANS	003442 G
HOE	= 100000 G	L\$DESP	002076 G	L10017	025370	O\$APTS	= 000000	RANSC	= 032561 G
HRDCNT	003310 G	L\$DEVP	002060 G	L10020	024136	O\$AU	= 000001	RANW	026012
IBE	= 010000 G	L\$DISP	002124 G	L10021	024162	O\$BGNR	= 000001	RANWR	025766
IDU	= 000040 G	L\$DLY	002116 G	L10022	024202	O\$BGNS	= 000001	RANWV	026000
IER	= 020000 G	L\$DTP	002040 G	L10023	024222	O\$DU	= 000001	RAN1	010644 G
IE.C	= 000200 G	L\$DTYP	002034 G	L10024	024242	O\$ERRT	= 000000	RAN2	010646 G
INIT10	021664	L\$DU	023644 G	L10025	024262	O\$GNSW	= 000001	RAN3	010650 G
INIT15	022104	L\$DUT	002072 G	L10026	024302	O\$POIN	= 000001	RCVERM	030672
INIT16	022124	L\$DVTY	002166 G	L10027	024322	O\$SETU	= 000001	RDF	= 104001 G
INTFLG	003476 G	L\$EF	002052 G	L10030	024342	PASCNT	003260 G	RDR	= 104401 G
INTPRI	= 000340 G	L\$ENVI	002044 G	L10031	024362	PATCH	031304 G	RECCNT	003330 G
IRE	003526 G	L\$ETP	002102 G	L10032	024420	PATERN	003454 G	RECLOG	003471 G
IREC	002210 G	L\$EXP1	002046 G	L10033	024604	PATRO	010350 G	RECRED	006546
IRECM	030714	L\$EXP4	002064 G	L10034	026044	PATR1	010406 G	RECTAP	006602 G
IREM	031005	L\$EXP5	002066 G	L10035	026522	PATR2	010426 G	RECUD	012012 G
ISR	= 000100 G	L\$HARD	027750 G	L10036	026666	PATR3	010436 G	REPEAT	= 050236
IXE	= 004000 G	L\$HIME	002120 G	L10037	027000	PATR4	010462 G	RERM	004632 G
I\$AU	= 000041	L\$HPCP	002016 G	L10040	027744	PATR5	010474 G	RETRY	= 050234
I\$AUTO	= 000041	L\$HPTP	002022 G	L10041	030020	PATR6	010506 G	RETRYC	003464 G
I\$CLN	= 000041	L\$HW	002176 G	L10042	031304	PATR7	010526 G	REWRT	015322
I\$DU	= 000041	L\$ICP	002104 G	L10043	031514	PATR8	010652 G	RFBC	002660 G
I\$HRD	= 000041	L\$INIT	021664 G	L10045	031520	PATTBL	010326	RFCERM	004336 G
I\$INIT	= 000041	L\$LADP	002026 G	MBR	= 100012 G	PATM	031226	RFREC	002760 G
I\$MOD	= 000041	L\$LAST	031510 G	MEMOM	023054	PCMDWD	003432 G	RFUNR	002770 G
I\$MSG	= 000041	L\$LOAD	002100 G	MISCFG	003537 G	PIRE	002213 G	RLEXM	004372 G
I\$PROT	= 000040	L\$LUN	002074 G	MOD.CO	= 000400 G	PNT	= 001000 G	RNF	= 125401 G
I\$PTAB	= 000041	L\$MREV	002050 G	MOD.C1	= 001000 G	PRI	= 002000 G	RNOPSC	= 177740 G
I\$PWR	= 000041	L\$NAME	002000 G	MOD.C2	= 002000 G	PRI00	= 000000 G	RNR	= 105401 G
I\$RPT	= 000041	L\$PRIO	002042 G	MOD.C3	= 004000 G	PRI01	= 000040 G	RNYM	004566 G
I\$SEG	= 000041	L\$PROT	021656 G	MOVMSG	011726 G	PRI02	= 000100 G	RPF	= 105001 G
I\$SETU	= 000041	L\$PRT	002112 G	MSGCNT	= 000016 G	PRI03	= 000140 G	RPR	= 125001 G
I\$SFT	= 000041	L\$REPP	002062 G	MSGPKA	002506 G	PRI04	= 000200 G	RPTCNT	003466 G
I\$SRV	= 000041	L\$REV	002010 G	MSGPKT	002340 G	PRI05	= 000240 G	RPTFLG	003523 G
I\$SUB	= 000041	L\$RPT	020052 G	MSGPK0	002356 G	PRI06	= 000300 G	RPT1A	020714
I\$TST	= 000041	L\$SOFT	030022 G	MSGPK1	002374 G	PRI07	= 000340 G	RPT1B	020771
JLOC	003452 G	L\$SPC	002056 G	MSGPK2	002412 G	PRXST	017624 G	RPT1C	021042
JLOOP	003450 G	L\$SPCP	002020 G	MSGPK3	002430 G	PTCMDS	027616	RPT1D	021113
JMP	= 000040 G	L\$SPTP	002024 G	MS.RFC	= 000004 G	PWRFLG	003535 G	RPT1E	021341
JMPMSG	031036	L\$STA	002030 G	MS.XS0	= 000006 G	RAMDAT	003346 G	RPT1F	021217
JMPMS2	031236	L\$SW	002204 G	MS.XS1	= 000010 G	RAMDUM	014050 G	RPT1G	021270
JMP.C	= 000040 G	L\$TEST	002114 G	MS.XS2	= 000012 G	RAMER	015736 G	RPT1I	021460
J\$JMP	= 000167	L\$TIML	002014 G	MS.XS3	= 000014 G	RAMFHR	005306	RPT1J	021371
LENMSK	003436 G	L\$UNIT	002012 G	NCMD.C	= 177740 G	RAMHLD	003342	RPT1K	021451
LOE	= 040000 G	L10000	002202	NCNT	003420 G	RAMIOP	005365	RRANV	002205 G
LOG	015436 G	L10001	002312	NCNT1	003422 G	RAMLIN	005450	RRBC	002620 G
LOOPCT	003525 G	L10002	005624	NEXTSP	030070	RAMP	031040	RRECL	= 000020 G
LOT	= 000010 G	L10003	006550	NEXTU	017346 G	RAMPD	005436	RRREC	002740 G
L\$ACP	002110 G	L10004	006556	NINUSE	= 177774 G	RAMR5H	003344	RRUNR	002750 G
L\$APT	002036 G	L10005	006564	NOINTM	004503 G	RAMSIZ	003406 G	RRVM	030617
L\$AU	023710 G	L10006	006572	NRDYM	023544	RAMWRT	002214 G	RS1	= 001233 G
L\$AUT	002070 G	L10007	006600	NSSRM	004353 G	RANB	003440 G	RS2	= 007622 G
L\$AUTO	023160 G	L10010	021654	NUMBM	031206	RANBC	= 153624 G	RS3	= 000000 G
L\$CCP	002106 G	L10012	023156	NURTY1	005155 G	RANCMD	025652	RTLE	014212 G

PARAMETER CODING
SYMBOL TABLE

RTLRTN 014346
 RWCPK 002334 G
 RWD = 102010 G
 RWERR 003473 G
 RSSAVE 003460 G
 SCCNT 003270 G
 SCERM 004312 G
 SCH = 140004 G
 SCHBK 002446 G
 SCHCNT= 000010 G
 SEQEND 003632 G
 SETCH 006700 G
 SETRW 006724 G
 SETUP 007706 G
 SFF = 105010 G
 SFPTBL 002204 G
 SFR = 105410 G
 SRF = 104010 G
 SRR = 104410 G
 STAERM 005626 G
 STAER1 006140
 STAER2 006316
 STAER3 006375
 STAER4 006433
 STAER5 006453
 STAER6 006262
 STAER7 006232
 STAF LG 003534 G
 STREAM 003532 G
 SVCGBL = 000000
 SVCINS = 000001
 SVCSUB = 000000
 SVCTAG = 000000
 SVCTST = 000000
 SWBFLG 003524 G
 SWB.C = 010000 G
 SYMD = 000007
 SYMS = 000007
 S\$LSYM = 010000
 TAPCAP 021571
 TCCRA 012612
 TCC0 012740 G
 TCC1 012762 G
 TCC2 013000 G
 TCC3 013146 G
 TCC4 013170 G
 TCC5 013670 G
 TCC6 013772 G
 TCC7 014026 G
 TC2RTN 013144
 TIME1 003444 G
 TIME2 003446 G
 TOERM 004271 G
 TOOMM 004542 G
 TRAPD4 003536 G
 TRAP4 023574 G
 TSAM 004520 G

TSBA = 002456 G
 TSC.FC = 177717 G
 TSC.TC = 177761 G
 TSDB 002456 G
 TSSR 002466 G
 TSSREG 003462 G
 TSVCT 002476 G
 TS.A16 = 000400 G
 TS.A17 = 001000 G
 TS.NBA = 002000 G
 TS.NXM = 004000 G
 TS.OFL = 000100 G
 TS.RMR = 010000 G
 TS.SC = 100000 G
 TS.SPE = 020000 G
 TS.SSR = 000200 G
 TS.UPE = 040000 G
 TS4ADR 027774
 TS4CL 002526 G
 TS4INT 002516 G
 TS4INO 006552 G
 TS4IN1 006560 G
 TS4IN2 006566 G
 TS4IN3 006574 G
 TS4VCT 030011
 T\$ARGC = 000001
 T\$CODE = 023004
 T\$ERRN = 000001
 T\$EXCP = 000000
 T\$FLAG = 000041
 T\$FREE = 031520
 T\$GMAN = 000000
 T\$HILI = 000010
 T\$LAST = 000001
 T\$LOLI = 000000
 T\$LSYM = 010000
 T\$LTNO = 000006
 T\$NEST = 177777
 T\$NSO = 000000
 T\$NS1 = 000005
 T\$NS2 = 000002
 T\$PCNT = 000000
 T\$PTAB = 010044
 T\$PTHV = 000001
 T\$PTNU = 000001
 T\$SAVL = 177777
 T\$SEGL = 177777
 T\$SIZE = 000004
 T\$SUBN = 000000
 T\$TAGL = 177777
 T\$TAGN = 010046
 T\$TEMP = 000000
 T\$TEST = 000006
 T\$TSTM = 177777
 T\$TSTS = 000001
 T\$TAU = 010016

T\$AUT = 010013
 T\$CLE = 010014
 T\$DAT = 010045
 T\$DU = 010015
 T\$HAR = 010041
 T\$HW = 010000
 T\$INI = 010012
 T\$MSG = 010003
 T\$PC = 000001
 T\$PRO = 010011
 T\$PTA = 010044
 T\$RPT = 010010
 T\$SOF = 010042
 T\$SRV = 010007
 T\$SUB = 010033
 T\$SW = 010001
 T\$TES = 010040
 T1 024004 G
 T1SWB 003530 G
 T1.1 024014
 T1.10 024344
 T1.11 024364
 T1.12 024450
 T1.2 024140
 T1.3 024164
 T1.4 024204
 T1.5 024224
 T1.6 024244
 T1.7 024264
 T1.8 024304
 T1.9 024324
 T2 025372 G
 T3 026046 G
 T4 026524 G
 T5 026670 G
 T5WEOT 027640
 T6 027002 G
 UAM = 000200 G
 UNL = 100412 G
 UNREC 003474 G
 URERM 004654 G
 VFEXC 016254 G
 VFISU 016540 G
 VFYCNT 003300 G
 VFYDAT 016142 G
 VFYFLG 003522 G
 VFY.C = 000100 G
 WLKZRO 010442
 WRBC 002560 G
 WRECL = 000020 G
 WRITEM 017776 G
 WRR = 105005 G
 WRREC 002720 G
 WRT = 104005 G
 WRTY 014350 G
 WRTYCT 003250 G

WRTYER 003470 G
 WRTYFG 003467 G
 WRUNR 002730 G
 WSSR 011672 G
 WTM = 100011 G
 WTR = 101011 G
 WTV = 104105 G
 WTVERM 004246 G
 WTYBRF 014732
 WTYCMD 014726
 WTYWRD 014730
 X\$ALWA = 000000
 X\$FALS = 000040
 X\$OFFS = 000400
 X\$TRUE = 000020
 X0.BOT = 000002 G
 X0.EOT = 000001 G
 X0.LET = 020000 G
 X0.ONL = 000100 G
 X0.RLL = 010000 G
 X0.RLS = 040000 G
 X0.TMK = 100000 G
 X1.EWN = 000010 G
 X2.OPM = 100000 G
 X3.DCK = 000010 G
 X3.RNY = 157400 G
 ZROPAT 010412
 \$BGNLE = 000002
 \$BRJMP = 177777
 \$BSKO = 050234
 \$BSK1 = 050236
 \$BSK2 = 050416
 \$ERFLG = 000400
 \$F\$AND = 000310
 \$F\$BAD = 000401
 \$F\$BLA = 000170
 \$F\$CAS = 000150
 \$F\$DEC = 000220
 \$F\$DO = 000340
 \$F\$FAL = 000405
 \$F\$GOO = 000400
 \$F\$IF = 000110
 \$F\$INC = 000210
 \$F\$LOO = 000200
 \$F\$NAM = 000160
 \$F\$NO = 000403
 \$F\$OR = 000320
 \$F\$RTI = 000350
 \$F\$RTN = 000300
 \$F\$SEL = 000140
 \$F\$THE = 000330
 \$F\$TRU = 000404
 \$F\$UNT = 000130
 \$F\$WHI = 000120
 \$F\$YES = 000402
 \$IFLEV = 177777

\$ISKO = 000001
 \$ISK1 = 000001
 \$ISK2 = 000001
 \$ISK3 = 000001
 \$ISK4 = 000001
 \$ISK5 = 000001
 \$ISK6 = 000001
 \$LOCTA = 177777
 \$LSTIN = 000001
 \$LSTTA = 000001
 \$LVTAG = 050416
 \$NESTL = 177777
 \$NSKO = 000120
 \$NSK1 = 000120
 \$NSK2 = 000110
 \$NSK3 = 000110
 \$NSK4 = 000110
 \$NSK5 = 000110
 \$NSK6 = 000110
 \$SAVE = 000001
 \$SAVLE = 177777
 \$SELLE = 177777
 \$SSKO = 050473
 \$TAGLE = 177777
 \$TAGNU = 050511
 \$TEMP = 000402
 \$TSKO = 050472
 \$TSK1 = 050473
 \$TSK2 = 050474
 \$TSK3 = 050475
 \$TSK4 = 050510
 \$TSK5 = 050507
 \$TSK6 = 050503
 \$TSK7 = 050505
 \$U = 000403
 \$\$ARGC = 000000
 \$\$BYTE = 000403
 \$\$CASE = 000000
 \$\$DST = 000000
 \$\$ELOC = 000402
 \$\$ERFL = 000000
 \$\$FLAG = 000001
 \$\$FROM = 000000
 \$\$INH = 000403
 \$\$LOC = 027155
 \$\$LOCN = 000000
 \$\$REG = 177777
 \$\$RETU = 000000
 \$\$RTN1 = 000000
 \$\$RTN2 = 000000
 \$\$SRC = 000000
 \$\$TGSV = 000000
 \$\$TGS1 = 000000
 \$\$TGS2 = 000000
 \$\$TO = 000000
 \$\$\$TAG = 050000

PARAMETER CODING
SYMBOL TABLE

MACRO M1200 21-MAR-84 16:45 PAGE 118-4

SEQ 0181

. ABS. 031520 000
000000 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 62895 WORDS (246 PAGES)
DYNAMIC MEMORY: 19748 WORDS (75 PAGES)
ELAPSED TIME: 00:38:39
CZTKIA.BIN,CZTKIA/-SP/CR=SVC/ML,SPMACJ/ML,CZTKIA