

TK25

TK25 DATA RELIAB TST
CZTKIB0

AH-T784B-MC
1 OF 1 JUL 1985
COPYRIGHT © 1984-85

digital
MADE IN USA

The main body of the document is a dense grid of approximately 15 columns and 15 rows of small, individual data plots or charts. Each plot appears to be a separate test result, but the text within them is too small and faded to be legible. The plots are arranged in a regular grid pattern across the page.

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

.REM E

IDENTIFICATION

PRODUCT CODE: AC-T783B-MC

PRODUCT NAME: CZTKIBO TK25 DATA RELIABILITY TEST

PRODUCT DATE: 15-MARCH-1985

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, 1985 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

44
45
46
47
48
49
50
51
52
53
54
55
56
57

REVISION HISTORY:

REVISION B (25-OCT-1984) BY TERRENCE W. REILLY
CORRECTED THE FOLLOWING:

- 1) DATA COMPARE ERROR ON A SUBSEQUENT UNIT, AFTER A DIFFERENT UNIT HAS BEEN DROPPED DUE TO A FATAL ERROR.
- 2) WRITE RETRY ALGORITHM HAS BEEN CHANGED TO PERMIT 256 BAD TAPE SPOTS. ONLY THE FIRST 20 WILL BE "MAPPED", BUT THE REMAINDER WILL BE TALLIED.
- 3) A LINE CLOCK SERVICE ROUTINE HAS BEEN ADDED TO ELIMINATE SOME OF THE MACHINE DEPENDENT TIMING LOOPS. (FOR THE 11/73)

USER'S GUIDE

59	
60	
61	
62	
63	
64	
65	1.0 GENERAL INFORMATION
66	1.1 PROGRAM ABSTRACT
67	1.1.1 FUNCTIONAL DESCRIPTION
68	1.1.2 STRUCTURE OF PROGRAM
69	1.1.3 MEMORY MAP
70	1.1.4 DIAGNOSTIC INFORMATION
71	1.1.4.1 SCOPE
72	1.1.4.2 ERROR RECOVERY
73	1.1.4.3 WRITE ERROR RECOVERY
74	1.1.4.3.1 MEDIA/OPERATIONAL SELECTIVE WRITE ERROR
75	RECOVERY ALGORITHM
76	1.1.4.3.2 OPERATIONAL WRITE-ERROR-RECOVERY ALGORITHM
77	1.1.4.4 EARLY WARNING WRITE ERRORS
78	1.1.4.5 DIAGNOSTIC TIMING ADJUSTMENT
79	1.2 SYSTEM REQUIREMENTS
80	1.2.1 HARDWARE REQUIREMENTS
81	1.2.2 SOFTWARE REQUIREMENTS
82	1.3 RELATED DOCUMENTS AND STANDARDS
83	1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
84	1.5 ASSUMPTIONS
85	1.6 DIAGNOSTIC HISTORY
86	2.0 OPERATING INSTRUCTIONS
87	2.1 HARDWARE PARAMETERS
88	2.2 SOFTWARE PARAMETERS
89	2.2.1 CLEAR COUNTERS
90	2.2.2 RESET RANDOM VARIABLES
91	2.2.3 PRINT SOFT ERRORS
92	2.2.4 INHIBIT RECOVERY
93	2.2.5 BAD TAPE SPOT DETECTION
94	2.2.6 DISABLE INTERRUPTS
95	2.2.7 INHIBIT RFC ERROR REPORTS
96	2.2.8 CONTROLLER RAM DUMP
97	2.2.9 ENABLE EARLY WARNING MESSAGES
98	2.2.10 CHANGE CMD SEQUENCE
99	2.2.11 COMMAND LIST FOR USE IN SOFTWARE DIALOGUE.
100	2.2.12 DATA PATTERN LIST FOR USE IN SOFTWARE DIALOGUE
101	2.3 EXAMPLES OF SOFTWARE DIALOGUE
102	2.3.1 BASIC FUNCTION AND DATA RELIABILITY WITH ALL
103	ERROR REPORTING ENABLED
104	2.3.2 TO SET UP A SCOPE LOOP FOR A FAILURE IN BASIC
105	FUNCTIONS.
106	2.3.3 TO SET UP A SCOPE LOOP FOR A FAILURE IN DATA
107	RELIABILITY
108	2.4 EXECUTION TIMES
109	2.4.1 SYSTEM CONFIGURATION
110	2.4.2 TEST EXECUTION TIMES
111	3.0 ERROR INFORMATION
112	3.1 ERROR REPORTING

114		
115		
116		
117	3.1.1	ERROR 1 - COMMAND PACKET ADDRESS NOT ON A
118		MODULO 4 BOUNDARY:
119	3.1.2	ERROR 2 - TK25 NOT READY:
120	3.1.3	ERROR 3 - NO RESPONSE ERROR:
121	3.1.4	ERROR 4 - NO INTERRUPT ERROR:
122	3.1.5	SPECIAL CONDITION ERRORS:
123	3.1.5.1	ERROR 5 - TERMINATION CLASS CODE 0, UNDEFINED
124		SPECIAL CONDITION
125	3.1.5.2	ERROR 6 - TERMINATION CLASS CODE 1, ATTENTION
126		CONDITION
127	3.1.5.3	ERROR 7 - TERMINATION CLASS CODE 2, TAPE
128		STATUS ALERT
129	3.1.5.4	ERROR 8 - TERMINATION CLASS CODE 3, FUNCTION
130		REJECT
131	3.1.5.5	ERROR 9 - TERMINATION CLASS CODE 4,
132		RECOVERABLE ERROR
133	3.1.5.6	ERROR 10 - TERMINATION CLASS CODE 5,
134		RECOVERABLE ERROR
135	3.1.5.7	ERROR 11 - TERMINATION CLASS CODE 6,
136		UNRECOVERABLE ERROR
137	3.1.5.8	ERROR 12 - TERMINATION CLASS CODE 7, FATAL
138		SUBSYSTEM ERROR
139	3.1.6	ERROR 13 - RFC NON-ZERO ERROR:
140	3.1.7	ERROR 14 - RETRY LIMIT EXCEEDED:
141	3.1.8	ERROR 15 - TOO MANY INTERRUPTS:
142	3.1.9	ERROR 16 - CAPSTAN RUNAWAY:
143	3.1.10	ERROR 17 - DATA COMPARE ERROR:
144	3.2	ERROR HALTS
145	4.0	PERFORMANCE REPORT
146	5.0	TEST SUMMARIES
147	5.1	TEST 1 - BASIC FUNCTIONS.
148	5.2	TEST 2 - DATA RELIABILITY.
149	5.3	TEST 3 - WRITE AND READ STREAMING TEST.
150	5.4	TEST 4 - WRITE COMPATABILITY/WRITE UTILITY.
151	5.5	TEST 5 - READ COMPATABILITY/READ UTILITY.
152	5.6	TEST 6 - EXECUTE OPERATOR SELECTED COMMAND
153		SEQUENCE.
154	6.0	DEVICE INFORMATION TABLES
155	6.1	GENERAL
156	6.2	BUS INTERFACE SPECIFICATIONS
157	6.3	BIT DEFINITIONS FOR TK25 REGISTERS
158	6.3.1	TK25 REGISTER SUMMARY
159	6.3.2	TK25 STATUS REGISTER (TSSR)
160	6.3.2.1	TSSR READ ONLY
161	6.3.2.2	TSSR WRITE ONLY
162	6.3.3	EXTENDED STATUS REGISTER 0 (XSTAT0)
163	6.3.4	EXTENDED STATUS REGISTER 1 (XSTAT1)
164	6.3.5	EXTENDED STATUS REGISTER 2 (XSTAT2)
165	6.3.6	EXTENDED STATUS REGISTER 3 (XSTAT3)

167		
168		
169		
170	GLOSSARY	
171	-----	
172		
173	ACT	AUTOMATED COMPUTER TEST SYSTEM
174		
175	APT	AUTOMATED PRODUCT TEST SYSTEM
176		
177	BYTE/RECORD/FILE COUNT	IS STORED IN THE 4TH WORD OF THE COMMAND PACKET
178	BRF	AND IT'S USE BY THE TK25 DEPENDS ON THE TYPE OF
179		COMMAND.
180		
181	CMD	TK25 COMMAND
182		
183	COMMAND PACKET	FOUR WORD PACKET IN THE CPU MEMORY WHICH
184	CMDPKT	CONTAINS ALL INFORMATION NEEDED BY THE TK25 TO
185		EXECUTE A COMMAND.
186		
187	EXTENDED STATUS	FOUR WORDS OF TK25 STATUS WHICH ARE TRANSFERRED
188		AS PART OF THE MESSAGE PACKET AT THE COMPLETION
189		OF A COMMAND.
190		
191	MESSAGE PACKET	SEVEN WORD PACKET IN THE CPU MEMORY INTO WHICH
192		THE TK25 STORES STATUS AT THE COMPLETION OF A
193		COMMAND.
194		
195	PC	PROGRAM COUNTER
196		
197	PSW	PROCESSOR STATUS WORD
198		
199	RESIDUAL FRAME COUNT	THIS COUNT IS PART OF THE MESSAGE PACKET AND
200	RFC	CONTAINS THE NUMBER OF BYTES/RECORDS/FILES
201		REMAINING TO BE PROCESSED AT THE COMPLETION OF A
202		COMMAND.
203		
204	SPECIAL CONDITION	TSS4 BIT15. WHEN SET, INDICATES THAT THE LAST
205	SPEC COND	COMMAND DID NOT COMPLETE WITHOUT INCIDENT.
206		
207	TERMINATION CLASS CODE	THREE BIT CODE IN THE TSSR WHICH INDICATES THE
208	TCC	THE TYPE OF COMMAND TERMINATION.
209		
210	TSBA	TAPE SYSTEM BUS ADDRESS REGISTER.
211		
212	TSDB	TAPE SYSTEM DATA BUFFER REGISTER.
213		
214	TSSR	TAPE SYSTEM STATUS REGISTER.
215		
216	XSTO	EXTENDED STATUS REGISTER 0
217		
218		

220	XST1	EXTENDED STATUS REGISTER 1
221		
222	XST2	EXTENDED STATUS REGISTER 2
223		
224	XST3	EXTENDED STATUS REGISTER 3
225		
226	XXDP+	XXDP+ IS A "CATCH-ALL" NAME FOR A GROUP OF
227		PDP-11 DIAGNOSTIC PACKAGES AVAILABLE ON
228		MULTIMEDIA.
229		

231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

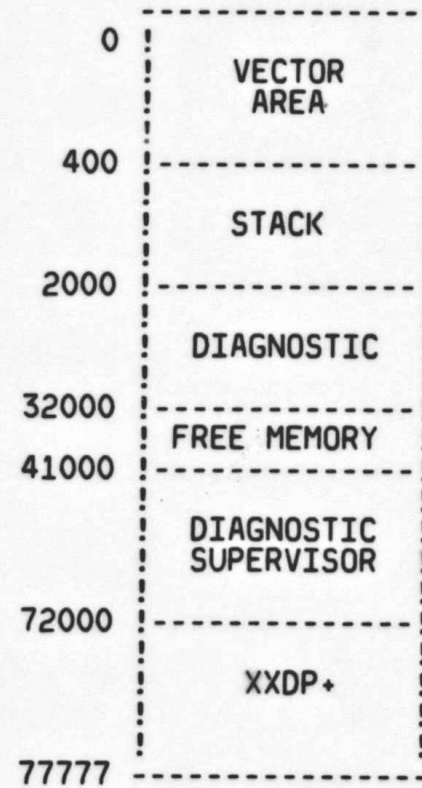
1.1.1 FUNCTIONAL DESCRIPTION -

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.

287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338

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.

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

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

393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435

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.

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
476
477
478
479
480
481
482
483
484
485
486
487
488

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 #:

SPEC COND	HARD	FATAL	COMPARE
6	6		
2	0	0	0

DR>

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
531
532
533
534
535
536
537
538
539
540
541
542
543

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 "L\$DLY". AS THE DIAGNOSTIC IS RUN ON DIFFERENT CPU'S, THESE DELAYS WILL VARY IN LENGTH WITH MEMORY SPEED.

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
587
588
589
590
591
592
593
594
595
596
597
598
599

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:

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
643
644
645
646
647
648
649
650
651
652
653
654
655

- 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.

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

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 FLEXABILITY 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 ?
- CONTROLLER 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.

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
743
744
745
746
747
748
749
750
751
752
753
754
755

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.

757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784

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>:

786			
787			
788			
789			
790		CHARACTERISTICS CODE (0) 40 ?	(0,20,40,200) (OCTAL)
791		CMD/2 (D) 13 ?	(1-27) (DECIMAL)
792		BRF COUNT (D) 1 ?	(1-4K) (DECIMAL)
793		# OF OPERATIONS (D) 1 ?	(1-32K) (DECIMAL)
794		PATTERN (D) 7 ?	(0-8) (DECIMAL)
795		CMD/3 (D) 4 ?	(1-27) (DECIMAL)
796		BRF COUNT (D) 4096 ?	(1-4K) (DECIMAL)
797		# OF OPERATIONS (D) 11719 ?	(1-32K) (DECIMAL)
798		PATTERN (D) 7 ?	(0-8) (DECIMAL)
799		CMD/4 (D) 7 ?	(1-27) (DECIMAL)
800		BRF COUNT (D) 4096 ?	(1-4K) (DECIMAL)
801		# OF OPERATIONS (D) 11719 ?	(1-32K) (DECIMAL)
802		PATTERN (D) 7 ?	(0-8) (DECIMAL)
803		CMD/5 (D) 2 ?	(1-27) (DECIMAL)
804		BRF COUNT (D) 4096 ?	(1-4K) (DECIMAL)
805		# OF OPERATIONS (D) 11719 ?	(1-32K) (DECIMAL)
806		PATTERN (D) 7 ?	(0-8) (DECIMAL)
807		CMD/6 (D) 13. ?	(1-27) (DECIMAL)
808		BRF COUNT (D) 1 ?	(1-4K) (DECIMAL)
809		# OF OPERATIONS (D) 1 ?	(1-32K) (DECIMAL)
810		PATTERN (D) 7 ?	(0-8) (DECIMAL)
811		CMD/7 (D) 27. ?	(1-27) (DECIMAL)
812		BRF COUNT (D) 4096 ?	(1-4K) (DECIMAL)
813		# OF OPERATIONS (D) 11719 ?	(1-32K) (DECIMAL)
814		PATTERN (D) 7 ?	(0-8) (DECIMAL)
815		CMD/8 (D) 27. ?	(1-27) (DECIMAL)
816		BRF COUNT (D) 4096 ?	(1-4K) (DECIMAL)
817		# OF OPERATIONS (D) 11719 ?	(1-32K) (DECIMAL)
818		PATTERN (D) 7 ?	(0-8) (DECIMAL)
819			
820			

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.

821
822
823
824
825
826
827
828
829

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
861
862
863
864
865
866
867
868
869
870
871
872
873

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" MILISECONDS WHERE N IS DEFINED IN THE NUMBER OF OPERATIONS. THIS DELAY IS USED BETWEEN EACH EXECUTABLE COMMAND.
27 =	END	END OF COMMAND SEQUENCE.

875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915

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> |

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
959
960
961
962
963
964
965
966
967
968
969
970
971

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 (O) 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 ? <1Z>
```

973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020

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

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
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063

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.

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
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118

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
CMDPKT 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.

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
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167

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.

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
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218

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.

1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265

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.

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
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321

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.

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
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375

- * 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.

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
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427

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.

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
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475

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.

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
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530

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).

1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559

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.

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
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606

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}	}TNO}	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 }

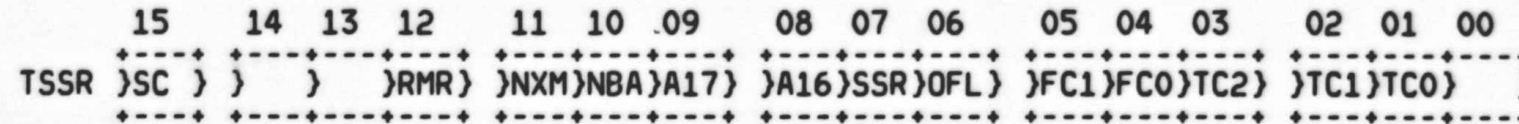
TERMINATION CLASS CODES (TSSR TC0-TC2):

- 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

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
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650

6.3.2 TK25 STATUS REGISTER (TSSR) -

6.3.2.1 TSSR READ ONLY -



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.

1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690

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 TC0, 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

1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726

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.

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
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780

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)TMK) )RLS)LET)RLL) )WLE)NEF)ILC) )ILA)MOT)ONL) )IE )VCK)  ) )WLK)BOT)EOT)

```

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

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
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834

				<ul style="list-style-type: none"> 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	5/1/3		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 A 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.

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
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887

6.3.4 EXTENDED STATUS REGISTER 1 (XSTAT1) -

```

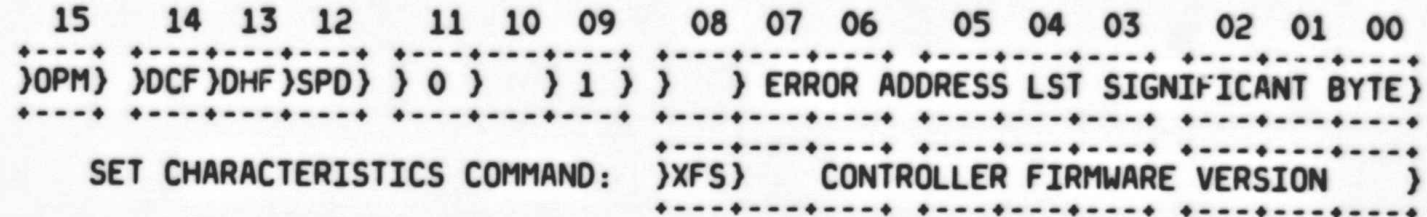
15  14  13  12  11  10  09  08  07  06  05  04  03  02  01  00
)DLT) ) ) )CRS) )NER) ) ) ) )TN3)TN2) )TN1)TNO) EW) ) )UNC) )

```

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	TNO	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.

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
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938

6.3.5 EXTENDED STATUS REGISTER 2 (XSTAT2) -



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.

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
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986

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	S	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	S	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

```

1999      .TITLE PROGRAM HEADER AND TABLES
2000      .SBTTL PROGRAM HEADER
2031
2033 000000      .ENABL ABS,AMA
2034      002000      = 2000
2036 002000      BGNMOD
2037
2038      ;++
2039      ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
2040      ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
2041      ;--
2042
2043 002000      POINTER BGNRPT,BGNSW,BGNSFT,BGNAU,BGNDU,BGNSETUP
2044
2052
2053 002000      HEADER CZTKI,B,0,5000,1,INTPRI
002000      L$NAME::      ;DIAGNOSTIC NAME
002000      103
002001      132      .ASCII /C/
002002      124      .ASCII /Z/
002003      113      .ASCII /T/
002004      111      .ASCII /K/
002005      000      .ASCII /I/
002006      000      .BYTE 0
002007      000      .BYTE 0
002010      L$REV::      ;REVISION LEVEL
002010      102      .ASCII /B/
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$HPCP::      ;POINTER TO H.W. QUES.
002016 030474      .WORD L$HARD
002020      L$SPCP::      ;POINTER TO S.W. QUES.
002020 030546      .WORD L$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$LADP::      ;DIAG. END ADDRESS
002026 032234      .WORD L$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 #	.WORD	0
002050	003			.BYTE	C\$REVISION
002051	003			.BYTE	C\$EDIT
002052		L\$EF::	;DIAG. EVENT FLAGS		
002052	000000			.WORD	0
002054	000000			.WORD	0
002056		L\$SPC::			
002056	000000			.WORD	0
002060		L\$DEVP::	; POINTER TO DEVICE TYPE LIST	.WORD	0
002060	002166			.WORD	L\$DVTYP
002062		L\$REPP::	;PTR. TO REPORT CODE	.WORD	L\$RPT
002062	020440			.WORD	0
002064		L\$EXP4::		.WORD	0
002064	000000			.WORD	0
002066		L\$EXP5::		.WORD	0
002066	000000			.WORD	0
002070		L\$AUT::	;PTR. TO ADD UNIT CODE	.WORD	L\$AU
002070	024424			.WORD	L\$DU
002072		L\$DUT::	;PTR. TO DROP UNIT CODE	.WORD	L\$DU
002072	024360			.WORD	0
002074		L\$LUN::	;LUN FOR EXERCISERS TO FILL	.WORD	0
002074	000000			.WORD	L\$DESC
002076		L\$DESP::	;POINTER TO DIAG. DESCRIPTION	.WORD	L\$DESC
002076	002140			EMT	E\$LOAD
002100		L\$LOAD::	;GENERATE SPECIAL AUTOLOAD EMT	.WORD	0
002100	104035			.WORD	L\$INIT
002102		L\$ETP::	;POINTER TO ERR TBL	.WORD	L\$INIT
002102	000000			.WORD	L\$CLEAN
002104		L\$ICP::	;PTR. TO INIT CODE	.WORD	L\$AUTO
002104	022252			.WORD	L\$PROT
002106		L\$CCP::	;PTR. TO CLEAN-UP CODE	.WORD	0
002106	024302			.WORD	0
002110		L\$ACP::	;PTR. TO AUTO CODE	.WORD	0
002110	023660			.WORD	0
002112		L\$PRT::	;PTR. TO PROTECT TABLE	.WORD	0
002112	022244			.WORD	0
002114		L\$TEST::	;TEST NUMBER	.WORD	0
002114	000000			.WORD	0
002116		L\$DLY::	;DELAY COUNT	.WORD	0
002116	000000			.WORD	0
002120		L\$HIME::	;PTR. TO HIGH MEM	.WORD	0
002120	000000			.WORD	0

2062
2063
2064
2065
2066
2067
2068
2069 002122
002122 000006
002124
002124 024520
002126 026106
002130 026562
002132 027250
002134 027414
002136 027526

.SBTTL DISPATCH TABLE

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

DISPATCH 6 ; SIX TESTS

L\$DISPATCH::

.WORD 6
.WORD T1
.WORD T2
.WORD T3
.WORD T4
.WORD T5
.WORD T6

2070
2077
2078
2079
2080
2081
2082
2083
2084

.SBTTL DESCRIPTIVE TEXT

;++
; 2 LINES OF TEXT PRINTED TO THE OPERATOR TO IDENTIFY THE DIAGNOSTIC AND THE DEVICE UNDER TEST
;--

DESCRIPT <DATA RELIABILITY TEST>

L\$DESC::

002140
002140 104 101 124
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

.ASCIZ /DATA RELIABILITY TEST/

2085 002166
002166
002166 124 113 062
002171 065 000

DEVTYP <TK25>

L\$DVTYP::

.EVEN
.ASCIZ /TK25/
.EVEN

```

2088      .SBTTL  DEFAULT HARDWARE P-TABLE
2089
2090
2091      ;++
2092      ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
2093      ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
2094      ; IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
2095      ;--
2096      002174      BGNHW  DFPTBL
                002174      000002
                002176
                002176
                L$HW::
                DFPTBL::
                .WORD  L10000-L$HW/2
2097
2103
2104      002176      172522
2105      002200      000224
                172522      ;TSSR ADDRESS.
                224        ;VECTOR ADDRESS.
2106
2107      002202
                002202
                ENDHW
                L10000:

```


2179
2180
2181
2190
2191 002312
2192
2193
2194
2195
2196
2197
2198 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
;
; 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.
2199
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221      100000      TS.SC==100000      ;SPECIAL CONDITION BIT.
2222      040000      TS.UPE==40000      ;UNIBUS PARITY ERROR
2223      020000      TS.SPE==20000      ;SERIAL BUS PARITY ERROR.
2224      010000      TS.RMR==10000      ;REGISTER MODIFICATION REFUSED.
2225      004000      TS.NXM==4000      ;NON-EXISTENT MEMORY.
2226      002000      TS.NBA==2000      ;NEED BUFFER ADDRESS.
2227      001000      TS.A17==1000      ;BUS ADDRESS BIT 17.
2228      000400      TS.A16==400      ;BUS ADDRESS BIT 16.
2229      000200      TS.SSR==200      ;UNIT READY BIT.
2230      000100      TS.OFL==100      ;OFF LINE.
2231      177717      TSC.FCC==177717      ;FATAL CLASS CODE MASK.
2232      177761      TSC.TCC==177761      ;TERMINATION CLASS CODE MASK.

```

```

2234      ;THE FOLLOWING ARE BIT DEFINITIONS FOR THE COMMAND WORD
2235
2236      100000      ACK.C==100000      ;ACKNOWLEDGE BIT
2237      040000      CVC.C==40000      ;CLEAR VOLUME CHECK.
2238      020000      OPP.C==20000      ;OPPOSITE BIT
2239      010000      SWB.C==10000      ;SWAP BYTE BIT
2240      004000      MOD.C3==4000      ;MODE BIT 3
2241      004000      BRF.C==4000      ;BYTE/RECORD/FILE COUNT FLAG BIT. NOT USED
2242      ;BY TK25 BUT USED INTERNALLY BY THIS PROGRAM ONLY.
2243      002000      MOD.C2==2000      ;MODE BIT 2
2244      001000      MOD.C1==1000      ;MODE BIT 1
2245      000400      MOD.C0==400      ;MODE BIT 0
2246      000200      IE.C==200      ;INTERRUPT ENABLE
2247      000100      FMT.C1==100      ;FORMAT BIT 1
2248      000100      VFY.C==100      ;WRITE VERIFY FLAG BIT. INTERNAL USE ONLY.
2249      ;NOT USED BY TK25.
2250      000040      FMT.C0==40      ;FORMAT BIT 0.
2251      000040      JMP.C==40      ;JUMP BIT-TO DIRECT THIS PROGRAM TO JUMP TO
2252      ;A CERTAIN LOCATION IN THE COMMAND SEQUENCE
2253      ;TABLE. INTERNAL USE ONLY.
2254      000020      CMD.C4==20      ;COMMAND BIT 4
2255      000020      DLY.C==20      ;INSERT DELAY. INTERNAL USE ONLY.
2256      000010      CMD.C3==10      ;COMMAND BIT 3
2257      000004      CMD.C2==4      ;COMMAND BIT 2
2258      000002      CMD.C1==2      ;COMMAND BIT 1
2259      000001      CMD.C0==1      ;COMMAND BIT 0
2260
2261      ; BIT DEFINITIONS FOR DEVICE CHARACTERISTICS.
2262
2263      000200      CH.ESS==200      ;ENABLE SKIP TAPE MARKS STOP (STOP AT LOGICAL EOT).
2264      000040      CH.EAI==40      ;ENABLE ATTENTION INTERRUPTS.
2265      000020      CH.ERI==20      ;ENABLE MESSAGE BUFFER RELEASE INTERRUPTS.
2266      000040      DFTSCH==CH.EAI      ;DEFAULT CHARACTERISTICS CODE.
2267
2268      ;THE FOLLOWING INDICATES THE RELATIVE POSITIONS OF THE STATUS WORDS
2269      ;IN THE MESSAGE BUFFER.
2270
2271      000004      MS.RFC==4      ;RESIDUAL FRAME COUNT.
2272      000006      MS.XS0==6      ;EXT STATUS REG 0
2273      000010      MS.XS1==10      ;EXT STATUS REG 1
2274      000012      MS.XS2==12      ;EXT STATUS REG 2
2275      000014      MS.XS3==14      ;EXT STATUS REG 3
2276
2277      ;THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0.
2278
2279      100000      XO.TMK==100000      ;TAPE MARK.
2280      040000      XO.RLS==40000      ;RECORD LENGTH SHORT.
2281      020000      XO.LET==20000      ;LOGICAL EOT.
2282      010000      XO.RLL==10000      ;RECORD LENGTH LONG.
2283      000100      XO.ONL==100      ;ON LINE BIT.
2284      000002      XO.BOT==2      ;BOT BIT.
2285      000001      XO.EOT==1      ;EOT BIT.
2286
2287      ;THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 1.
2288
2289      000010      X1.EWN==BIT3
2290

```

```

2291 ;THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2.
2292
2293 100000 X2.OPM==100000 ;OPERATION IN PROGRESS, TAPE MOVING
2294
2295 ;THE FOLLOWING ARE BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3.
2296
2297 000010 X3.DCK==10 ;DENSITY CHECK.
2298 157400 X3.RNY==157400 ;CAPSTAN RUNAWAY UDIAG ERROR CODE.
2299
2300 ;THE FOLLOWING DEFINITIONS SHOW THE RELATIVE POSITIONS OF THE COMMAND
2301 ;PACKET ENTRIES.
2302
2303 000000 CP.CMD==0 ;CMDPKT.0==TK25 COMMAND.
2304 000002 CP.ADL==2 ;CMDPKT.2==BUFFER ADDRESS LOW.
2305 000004 CP.ADH==4 ;CMDPKT.4==BUFFER ADDRESS HIGH.
2306 000006 CP.CNT==6 ;CKDPKT.6==BYTE/FILE/RECORD COUNT
2307
2308 ; MISCELLANEOUS DEFINITIONS.
2309
2310 000340 INTPRI==PRI07 ;PRIORITY TO BE USED IN INTERRUPT STATE.
2311 000010 SCHCNT==10 ;ARBITRARY BYTE LENGTH FOR CHARACTERISTIC
2312 ;BUFFER LENGTH. (EVEN #)
2313 000016 MSGCNT==16 ;MESSAGE BUFFER LENGTH IN BYTES. (EVEN #)
2314 000020 DIACNT==20 ;DIAGNOSTIC COMMAND BUFFER EXTENT.
2315 010000 DATCNT==4096. ;MAXIMUM RECORD LENGTH IN BYTES.
2316 ;THIS COUNT SHOULD BE A MULTIPLE OF 256 TO INSURE
2317 ;PROPER READ/WRITE BUFFER ALLOCATION BY THE SUPER.
2318 177740 RNOPSC==177740 ;RANDOM # OF OPERATIONS MASK.
2319 000007 RANP==7 ;CODE TO SELECT RANDOM PATTERN.
2320 000020 RRECL==16. ;READ RECOVERY ATTEMPT LIMIT.
2321 000020 WRECL==16. ;WRITE RECOVERY ATTEMPT LIMIT.
2322 153624 RANBC==153624 ;CONSTANT USED TO RESET RANDOM # GENERATOR BASE.
2323 032561 RANSC==32561 ;CONSTANT USED TO RESET RANDOM # SAVE LOCATION.
2324 177774 NINUSE==177774 ;NOT IN USE CODE FOR DEVICE STATE TABLE.
2325 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
2326 ;NOT "COMMAND" BITS.
2327
2328 ;THE FOLLOWING DEFINES THE COMMAND WORD FOR EACH TK25 COMMAND.
2329
2330 100013 DRI== ACK.C!CMD.C3!CMD.C1!CMD.CO ;DRIVE INIT.
2331
2332
2333 104001 RDF== ACK.C!BRF.C!CMD.CO ;READ FORWARD
2334
2335
2336 104401 RDR== ACK.C!BRF.C!MOD.CO!CMD.CO ;READ REVERSE
2337
2338
2339 104005 WRT== ACK.C!BRF.C!CMD.CO!CMD.C2 ;WRITE COMMAND
2340
2341
2342 104105 WTV== ACK.C!BRF.C!VFY.C!CMD.CO!CMD.C2 ;WRITE VERIFY
2343
2344
2345 104010 SRF== ACK.C!BRF.C!CMD.C3 ;SPACE RECORD FORWARD
2346
2347

```

2348	104410	SRR==	ACK.C!BRF.C!MOD.CO!CMD.C3	
2349				;SPACE RECORD REVERSE
2350				
2351	105401	RNR==	ACK.C!BRF.C!MOD.C1!MOD.CO!CMD.CO	
2352				;READ REV RETRY1 - REREAD NEXT REVERSE, IE. SPACE FWD, READ REVERSE
2353				
2354	125401	RNF==	ACK.C!BRF.C!OPP.C!MOD.C1!MOD.CO!CMD.CO	
2355				;READ REV RETRY2 - REREAD NEXT FORWARD, IE. READ FORWARD, SPACE REVERSE
2356				
2357	105001	RPF==	ACK.C!BRF.C!MOD.C1!CMD.CO	
2358				;READ FWD RETRY1 - REREAD PREVIOUS FORWARD, IE. SPACE REVERSE, READ FORWARD
2359				
2360	125001	RPR==	ACK.C!BRF.C!OPP.C!MOD.C1!CMD.CO	
2361				;READ FWD RETRY2 - REREAD PREVIOUS REVERSE, IE. READ REVERSE, SPACE FORWARD
2362				
2363	105005	WRR==	ACK.C!MOD.C1!BRF.C!CMD.C2!CMD.CO	
2364				;WRITE RETRY
2365				
2366	102010	RWD==	ACK.C!MOD.C2!CMD.C3	
2367				;REWIND COMMAND
2368				
2369	100012	MBR==	ACK.C!CMD.C3!CMD.C1	
2370				;MESSAGE BUFFER RELEASE
2371				
2372	100011	WTM==	ACK.C!CMD.C3!CMD.CO	
2373				;WRITE TAPE MARK.
2374				
2375	101011	WTR==	ACK.C!MOD.C1!CMD.C3!CMD.CO	
2376				;WRITE TAPE MARK RETRY.
2377				
2378	105010	SFF==	ACK.C!BRF.C!MOD.C1!CMD.C3	
2379				;SPACE FILE FORWARD
2380				
2381	105410	SFR==	ACK.C!BRF.C!MOD.CO!MOD.C1!CMD.C3	
2382				;SPACE FILE REVERSE
2383				
2384	100017	GES==	ACK.C!CMD.CO!CMD.C1!CMD.C2!CMD.C3	
2385				;GET EXTENDED STATUS
2386				
2387	100411	ERS==	ACK.C!MOD.CO!CMD.C3!CMD.CO	
2388				;ERASE 3 INCHES OF TAPE
2389				
2390	100412	UNL==	ACK.C!MOD.CO!CMD.C3!CMD.C1	
2391				;UNLOAD COMMAND
2392				
2393	101012	CLN==	ACK.C!MOD.C1!CMD.C3!CMD.C1	
2394				;ERASE TAPE.
2395				
2396	140004	SCH==	ACK.C!CVC.C!CMD.C2	
2397				;SET DEVICE CHARACTERISTICS.
2398	100006	DIA==	ACK.C!CMD.C2!CMD.C1	
2399				;DIAGNOSTICS.
2400	000040	JMP==	JMP.C	
2401				;JUMP TO "N"TH COMMAND
2402	000020	DLY==	DLY.C	
2403				;DELAY "N" MS.
2404	177777	END==	177777	
				;END OF COMMAND SEQUENCES

```

2406          .SBTTL GLOBAL DATA SECTION
2407
2408          ;**
2409          ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
2410          ; IN MORE THAN ONE TEST.
2411          ;--
2412
2413          ;      COMMAND PACKET.
2414
2415          =          .+3&177774          ;MUST BE ON MOD 4 BOUNDRY.
2416 002314 000000  CNDPKT:: 0          ;1ST WORD IS TK25 COMMAND.
2417 002316 000000          0          ;2ND WORD IS THE BUFFER LOW ADDRESS.
2418 002320 000000          0          ;3RD WORD IS THE BUFFER HIGH ADDRESS.
2419 002322 000000          0          ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.
2420
2421          ;      GET STATUS COMMAND PACKET.
2422
2423          =          .+3&177774          ;MUST BE ON MOD 4 BOUNDRY.
2424 002324 100017  GSCPK:: .WORD  GES
2425
2426          ;      MESSAGE BUFFER RELEASE COMMAND PACKET.
2427
2428          =          .+3&177774          ;MUST BE ON MOD 4 BOUNDRY.
2429 002330 100012  BRCPK:: .WORD  MBR
2430
2431          ;      REWIND COMMAND PACKET (USED IN ERROR RECOVERY ONLY)
2432
2433          =          .+3&177774          ;MUST BE ON A MODULE 4 BOUNDARY.
2434 002334 102010  RWCPK:: .WORD  RWD
2435 002336 000001          .WORD  1
2436
2437          ;      WORK AREA FOR ANALYSIS OF MESSAGE PACKET CONTENTS.
2438
2439          MSGPKT:: .BLKW 7          ;1ST WORD:: MESSAGE TYPE.
2440          ;2ND WORD:: DATA FIELD LENGTH.
2441          ;3RD WORD:: RESIDUAL FRAME COUNT.
2442          ;4TH WORD:: XSTAT0
2443          ;5TH WORD:: XSTAT1
2444          ;6TH WORD:: XSTAT2
2445          ;7TH WORD:: XSTAT3
2446
2447          ;      MESSAGE PACKETS.
2448
2449          MSGPK0:: .BLKW 7          ;MESSAGE PACKET FOR DEVICE #0
2450 002356          MSGPK1:: .BLKW 7          ;MESSAGE PACKET FOR DEVICE #1
2451 002374          MSGPK2:: .BLKW 7          ;MESSAGE PACKET FOR DEVICE #2
2452 002412          MSGPK3:: .BLKW 7          ;MESSAGE PACKET FOR DEVICE #3
2453 002430

```

```

2455      ;      SET CHARACTERISTIC BLOCK.
2456
2457 002446 002356      SCHBK:: MSGPK0      ;1ST WORD:: MSGPKT ADDR LO(SET UP BY EXECUTE ROUTINE).
2458 002450 000000      0      ;2ND WORD:: MSGPKT ADDR HI.
2459 002452 000016      MSGCNT      ;3RD WORD:: MSG BUFFER LENGTH (BYTES)
2460 002454 000040      CH.EAI      ;4TH WORD:: CHARACTERISTICS WORD(SET BY SETUP ROUTINE).
2461
2462      ;      TK25 REGISTER ADDRESSES.
2463
2464 002456      TSDB:: .BLKW 4      ;TK25 DATA BUFFER ADDRESSES.
2465 002466      TSSR:: .BLKW 4      ;TK25 STATUS REGISTER ADDRESSES.
2466 002476      TSVCT:: .BLKW 4      ;TK25 VECTOR ADDRESSES.
2467      002456      TSBA==TSDB      ;DATA BUFFER ADDRESS REGISTER.
2468
2469      ;      ADDRESSES OF MESSAGE PACKETS.
2470
2471 002506 002356      MSGPKA:: MSGPK0      ;DEVICE 0.
2472 002510 002374      MSGPK1      ;DEVICE 1.
2473 002512 002412      MSGPK2      ;DEVICE 2.
2474 002514 002430      MSGPK3      ;DEVICE 3.
2475
2476      ;      ADDRESSES OF INTERRUPT HANDLING ROUTINES.
2477
2478 002516 006564      TS4INT:: TS4INO      ;DEVICE 0.
2479 002520 006606      TS4IN1      ;DEVICE 1.
2480 002522 006630      TS4IN2      ;DEVICE 2.
2481 002524 006652      TS4IN3      ;DEVICE 3.
2482
2483      ;      TK25 CODE LEVELS, WILL BE STORED AFTER SCH CMD IN BASIC FUNCTION TEST
2484
2485 002526 000000      TS4CL:: 0      ;DEVICE 0
2486 002530 000000      0      ;DEVICE 1
2487 002532 000000      0      ;DEVICE 2
2488 002534 000000      0      ;DEVICE 3
2489
2490      ;      UNIT NUMBERS OF ALL DEVICES BEING TESTED(1-4).
2491      ;      WHEN DEVICE IS NOT IN USE, IT,S LOCATION WILL = -3.
2492      ;      R5 WILL ALWAYS CONTAIN THE PRESENT LOGICAL UNIT NUMBER X 2.
2493
2494 002536 177774      DEVTBL:: .WORD NINUSE
2495 002540 177774      .WORD NINUSE
2496 002542 177774      .WORD NINUSE
2497 002544 177774      .WORD NINUSE
2498 002546 177777      .WORD END
2499
2500      ;      BAD TAPE TABLE POINTER: USED BY WRITE RETRY ROUTINE
2501      ;      "WRTY" TO LOG BAD TAPE SPOTS ON UNITS UNDER TEST
2502
2503 002550 003000      BTADDR:: BT0
2504 002552 003052      BT1
2505 002554 003124      BT2
2506 002556 003176      BT3

```

```

2508 ; COUNTER AREA.
2509
2510 002560 CNTBGN=
2511 002560 WRBC:: .BLKW 20 ;BYTES WRITTEN.
2512 002620 RRBC:: .BLKW 20 ;BYTES READ REV.
2513 002660 RFBC:: .BLKW 20 ;BYTES READ FWD.
2514 002720 WRREC:: .BLKW 4 ;RECOVERABLE WRITE ERRORS.
2515 002730 WRUNR:: .BLKW 4 ;UNRECOVERABLE WRITE ERRORS.
2516 002740 RRREC:: .BLKW 4 ;RECOVERABLE READ REV ERRORS.
2517 002750 RRUNR:: .BLKW 4 ;UNRECOVERABLE READ REV ERRORS.
2518 002760 RFREC:: .BLKW 4 ;RECOVERABLE READ FWD ERRORS.
2519 002770 RFUNR:: .BLKW 4 ;UNRECOVERABLE READ FWD ERRORS.
2520 003000 BT0:: .BLKW 21. ;UNIT 0 BAT TAPE SPOTS LOG
2521 003052 BT1:: .BLKW 21. ;UNIT 1 BAT TAPE SPOTS LOG
2522 003124 BT2:: .BLKW 21. ;UNIT 2 BAT TAPE SPOTS LOG
2523 003176 BT3:: .BLKW 21. ;UNIT 3 BAT TAPE SPOTS LOG
2524 003250 WRTYCT:: .BLKW 4 ;WRITE RETRY COUNTER
2525 003260 PASCNT:: .BLKW 4 ;PASS COUNT.
2526 003270 SCCNT:: .BLKW 4 ;SPECIAL CONDITION COUNT.
2527 003300 VFYCNT:: .BLKW 4 ;COUNT OF TK25 DATA COMPARE ERRORS.
2528 003310 HRDCNT:: .BLKW 4 ;COUNT OF HARD ERRORS.
2529 003320 FTLCNT:: .BLKW 4 ;COUNT OF FATAL ERRORS.
2530 003330 CNTEND=
2531 003330 RECCNT:: .BLKW 4 ;END OF STATICTICAL COUNTERS.
2532 ;NUMBER OF RECORDS FROM BOT: CLEARED ON REWIND
2533 000550 CNTLEN==CNTEND-CNTBGN ;AND WHEN RESTARTING OR CONTINUING TEST 2.
2534 ;LENGTH OF STATISTICAL COUNTER AREA.
2535 003340 000
2536 HERE: .BYTE 0 ;TEST 3 ASCII SEMAPHORE
2537 ; THE FOLLOWING ARE THE DEFINITIONS OF VARIABLES
2538 ; USED BY THE PROGRAM.
2539 .EVEN
2540 003342 000000 RAMHLD: .WORD 0 ;RAM ADDR HOLDER 1ST ADDRESS
2541 003344 000000 RAMR5H: .WORD 0 ;HOLDS R5 FOR LATER
2542 003346 RAMDATA:: .BLKW 16. ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
2543 003406 000000 RAMSIZ:: .WORD 0 ;RAM DATA SIZE FOR PRAMPKT ROUTINE
2544 003410 000000 CMPDAT:: .WORD 0 ;COUNTS # OF READS (TEST 3) BEFORE ALLOWING A DATA COMPARE
2545 003412 000003 DATRAT:: .WORD 3 ;CONTROLS THE DATA COMPARE RATIO
2546 003414 000000 DATAW:: .WORD 0 ;WRITE BUFFER ADDRESS.
2547 003416 000000 DATARD:: .WORD 0 ;READ BUFFER ADDRESS.
2548 003420 000000 NCNT:: .WORD 0 ;STORAGE FOR VALUE OF N.
2549 003422 000000 NCNT1:: .WORD 0 ;TEMP STORAGE FOR VALUE OF N.
2550 003424 000000 BRFCNT:: .WORD 0 ;STORAGE FOR BPCR VALUE.
2551 003426 177777 CMDWRD:: .WORD END ;CONTAINS COMMAND WORD BEING EXECUTED PRESENTLY.
2552 003430 177777 CMDSAV:: .WORD END ;SAVE LOCATION FOR CMD WORD DURING ERROR RECOVERY
2553 003432 177777 PCMDWD:: .WORD END ;CONTAINS PREVIOUS COMMAND WORD.
2554 003434 000000 CMDLG:: .WORD 0 ;CURRENT COMMAND LOGGING CODE.
2555 003436 000000 LENMSK:: .WORD 0 ;RANDOM WRITE LENGTH MASK, TO BE SET UP BY TESTS
2556 003440 153624 RANB:: .WORD 153624 ;RANDOM # GENERATOR BASE.
2557 003442 032561 RANS:: .WORD 32561 ;RANDOM # SAVE LOCATION.
2558 003444 000000 TIME1:: .WORD 0 ;TIME COUNT 1.
2559 003446 000000 TIME2:: .WORD 0 ;TIME COUNT 2.
2560 003450 000000 JLOOP:: .WORD 0 ;JMP COMMAND LOOP COUNT.
2561 003452 000000 JLOC:: .WORD 0 ;JMP COMMAND LOCATION COUNT.
2562 003454 000000 PATERN:: .WORD 0 ;PATTERN SELECT CODE.

```



```

2564 003456 000000 CTCC:: .WORD 0 ;CURRENT TERMINATION CLASS CODE.
2565 003460 000000 R5SAVE:: .WORD 0 ;LOCATION FOR SAVING CURRENT DEVICE POINTER.
2566 003462 000000 TSSREG:: .WORD 0 ;CURRENT STATUS REGISTER.
2567 003414 003414 DIABLK=DATAWT ;WRITE BUFFER ALSO USED FOR DIAG CMD.
2568 003464 000000 LCSR:: .WORD 0 ;L CLOCK CSR ADDRESS
2569 003466 000000 LBRVEC:: .WORD 0 ;L CLOCK BR LEVEL
2570 003470 000000 LVECT:: .WORD 0 ;L CLOCK VECTOR ADDRESS
2571 003472 000000 LHERTZ:: .WORD 0 ;L CLOCK FREQUENCY
2572
2573 ; ERROR FLAG AREA, THESE FLAGS ARE CLEARED DURING INITIALIZATION AND
2574 ; AFTER EACH COMMAND IS COMPLETED.
2575
2576 003474 BGNFLG=.
2577 003474 000000 RETRYC:: .WORD 0 ;# OF RECOVERY ATTEMPTS EXECUTED.
2578 003476 000 RPTCNT:: .BYTE 0 ;WRITE REPEAT ON SAME SPOT CNTR: 4 PER WRITE RETRY
2579 003477 000 WRTYFG:: .BYTE 0 ;WRITE RETRY ON SAME SPOT IN PROGRESS FLAG
2580 003500 000 WRTYER:: .BYTE 0 ;WRITE RETRY ON SAME SPOT ERROR FLAG
2581 003501 000 RECLOG:: .BYTE 0 ;RECORD COUNT HAS BEEN UPDATED FOR THIS RECORD.
2582 003502 000 ERLOG:: .BYTE 0 ;DATA BYTES AND ERRORS HAVE BEEN LOGGED FOR THIS RECORD.
2583 003503 000 RWERR:: .BYTE 0 ;READ/WRITE ERROR HAS OCCURED.
2584 003504 000 UNREC:: .BYTE 0 ;UNRECOVERABLE ERROR HAS OCCURED.
2585 003505 000 ERRREC:: .BYTE 0 ;ERROR RECOVERY MODE.
2586 .EVEN
2587 003506 ENDERF=.
2588
2589 ; ADDITIONAL FLAGS, THESE FLAGS ARE CLEARED DURING INITIALIZATION.
2590
2591 003506 INTFLG:: .BLKW 4 ;INTERRUPT OCCURRED FLAGS FOR EACH DEVICE.
2592 003516 EOTFLG:: .BLKW 4 ;EOT/BOT FLAGS FOR EACH DEVICE (XSTATO).
2593 003526 000000 BTPT:: .WORD 0 ;BAD TAPE SPOT POINTER TO BT0-BT3 VIA BTADDR
2594 003530 000000 TC4STO:: .WORD 0 ;TCC4 ROUTINE - TEMPORARY STORAGE
2595 003532 000 EXPBOT:: .BYTE 0 ;BOT IS EXPECTED, DO NOT ABORT ON BOT/FUNC RTI.
2596 003533 000 RANDOM:: .BYTE 0 ;RANDOM EVERYTHING FLAG.
2597 003534 000 VFYFLG:: .BYTE 0 ;SET DURING WRITE/VERIFY COMMAND.
2598 003535 000 RPTFLG:: .BYTE 0 ;PERFORMANCE REPORT HAS BEEN REQUESTED.
2599 003536 000 SWBFLG:: .BYTE 0 ;ENABLES SWAP BYTE FUNCTION WHEN NOT EQUAL TO ZERO.
2600 003537 000 LOOPCT:: .BYTE 0 ;WRITE LOOP CONTROL (TEST 3)
2601 003540 000 IRE:: .BYTE 0 ;INHIBIT RESIDUAL FRAME COUNT ERROR REPORT.
2602 003541 000 DROPED:: .BYTE 0 ;CURRENT UNIT HAS BEEN DROPPED
2603 003542 000 T1SWB:: .BYTE 0 ;TEST1 SWAP BYTES FLAG
2604 003543 000 ALLEOT:: .BYTE 0 ;ALL UNITS @ EOT FLAG
2605 003544 000 STREAM:: .BYTE 0 ;INDICATES TEST ONE UNIT AT A TIME, COMPLETELY.
2606 003545 000 ERSFLG:: .BYTE 0 ;ERASE FLAG: DO ERASE AFTER A SPACE REV TO DELETE
;BADLY WRITTEN RECORD. 1 TO 4 ERASES LEAVING
;A 3 TO 12 INCH GAP MAY RESULT.
2607
2608
2609 .EVEN
2610 003546 ENDFLG=.
2611
2612 ; ADDITIONAL FLAGS, THESE FLAGS ARE CLEARED ONLY AFTER BEING CHECKED.
2613
2614 003546 000 STAF LG:: .BYTE 0 ;START FLAG - SET BY INIT CODE IF STARTING.
2615 003547 000 PWRFLG:: .BYTE 0 ;POWER FAILURE FLAG - SET ONLY DURING INIT.
2616 003550 000 TRAPD4:: .BYTE 0 ;TRAPED AT 4 FLAG
2617 003551 000 MISCFG:: .BYTE 0 ;MISCELLANEOUS FLAG
2618

```

```

2620 ; OPERATOR FLAG SETTINGS PASSED BY DIAG. SUPERVISOR IN A 16 BIT WORD
2621 ; SEE GLOBAL EQUATES SECTION FOR FLAG BIT LIST
2622
2623 003552 000000 OPFLAG:: .WORD 0 ;READ ONLY OPERATOR FLAG WORD
2624 .EVEN
2625
2626 ;THE FOLLOWING IS THE COMMAND SEQUENCE TABLE. THE TABLE
2627 ;HAS DEFAULT VALUES AT PROGRAM LOAD AS SHOWN. THESE VALUES
2628 ;CAN BE UPDATED BY A TEST OR BY OPERATOR INPUT.
2629
2630 003554 140004 CMDSEQ:: .WORD SCH ;SET CHARACTERISTICS.
2631 003556 000040 .WORD CH.EAI
2632 003560 000001 .WORD 1
2633 003562 000000 .WORD 0
2634 003564 102010 CMDSE2:: .WORD RWD ;REWIND.
2635 003566 000001 .WORD 1 ;BYTE COUNT.
2636 003570 000001 .WORD 1 ;ONCE.
2637 003572 000007 .WORD RANP ;PATTERN.
2638 003574 104005 .WORD WRT ;WRITE.
2639 003576 010000 .WORD DATCNT ;MAX BUFFER LENGTH.
2640 003600 035230 .WORD 15000. ;15000 RECORDS.
2641 003602 000007 .WORD RANP ;PATTERN - RANDOM DATA
2642 003604 102010 .WORD RWD ;REWIND
2643 003606 000001 .WORD 1 ;BYTE COUNT
2644 003610 000001 .WORD 1 ;ONE ITERATION
2645 003612 000007 .WORD RANP ;RANDOM DATA
2646 003614 104001 .WORD RDF ;READ FWD.
2647 003616 010000 .WORD DATCNT ;MAX BUFFER LENGTH.
2648 003620 035230 .WORD 15000. ;15000 RECORDS.
2649 003622 000007 .WORD RANP ;RANDOM DATA
2650 003624 102010 .WORD RWD ;REWIND
2651 003626 000001 .WORD 1 ;BYTE COUNT.
2652 003630 000001 .WORD 1 ;ONCE.
2653 003632 000007 .WORD RANP ;PATTERN.
2654 003634 .BLKW 4 ;EXTENSION TO HOLD 1 MORE CMD.
2655 003644 177777 SEQEND:: .WORD END ;SOFT END OF SEQUENCE TABLE.
2656 003646 177777 .WORD END
2657 003650 177777 .WORD END
2658 003652 177777 .WORD END
2659 003654 177777 .WORD END ;HARD END OF SEQUENCE TABLE.

```

```

2661                                     ;THE FOLLOWING IS THE TK25 COMMAND TABLE
2662
2663 003656 100013          CMDTBL:: .WORD DRI          ;DRIVE INIT.
2664 003660 104001          .WORD RDF          ;READ FORWARD.
2665 003662 104401          .WORD RDR          ;READ REVERSE.
2666 003664 104005          .WORD WRT          ;WRITE
2667 003666 104105          .WORD WTV          ;WRITE/VERIFY. (WRITE ALL RECORDS, RDR AND
2668                                     ;CHECK DATA ON ALL RECORDS, RDF AND
2669                                     ;CHECK DATA ON ALL RECORDS.)
2670 003670 104010          .WORD SRF          ;SPACE "N" RECORDS FORWARD.
2671 003672 104410          .WORD SRR          ;SPACE "N" RECORDS REVERSE.
2672 003674 105401          .WORD RNR          ;READ NEXT REVERSE. I.E., SPACE FWD, READ REVERSE.
2673 003676 125401          .WORD RNF          ;READ NEXT FORWARD, I.E., READ FORWARD, SPACE REVERSE.
2674 003700 105001          .WORD RPF          ;READ PREVIOUS FORWARD. I.E., SPACE REVERSE, READ FORWARD
2675 003702 125001          .WORD RPR          ;READ PREVIOUS REVERSE. I.E., READ REVERSE, SPACE FORWARD
2676 003704 105005          .WORD WRR          ;WRITE RETRY.
2677 003706 102010          .WORD RWD          ;REWIND.
2678 003710 100012          .WORD MBR          ;MESSAGE BUFFER RELEASE
2679 003712 100011          .WORD WTM          ;WRITE TAPE MARK
2680 003714 101011          .WORD WTR          ;WRITE TAPE MARK RETRY.
2681 003716 105010          .WORD SFF          ;SPACE "N" FILES FORWARD.
2682 003720 105410          .WORD SFR          ;SPACE "N" FILES REVERSE.
2683 003722 100017          .WORD GES          ;GET EXTENDED STATUS.
2684 003724 100411          .WORD ERS          ;ERASE 3 INCHES OF TAPE.
2685 003726 100412          .WORD UNL          ;REWIND AND UNLOAD.
2686 003730 101012          .WORD CLN          ;CLEAR TAPE.
2687 003732 140004          .WORD SCH          ;SET CHARACTERISTICS.
2688 003734 100006          .WORD DIA          ;DIAGNOSTIC COMMAND.
2689 003736 000040          .WORD JMP          ;JUMP TO THE NTH COMMAND IN THE SEQUENCE.
2690 003740 000020          .WORD DLY          ;DELAY "N" MS.
2691 003742 177777          .WORD END          ;END OF COMMAND TABLE
2692

```

```

2694                                     ; THE FOLLOWING TABLE CONTAINS THE ASCII FOR EACH COMMAND.
2695
2696 003744      104      122      111  CMDASC:: .ASCII /DRI/      ;DRIVE INIT.
2697 003747      122      104      106      .ASCII /RDF/      ;READ FORWARD.
2698 003752      122      104      122      .ASCII /RDR/      ;READ REVERSE.
2699 003755      127      122      124      .ASCII /WRT/      ;WRITE
2700 003760      127      124      126      .ASCII /WTV/      ;WRITE/VERIFY. (WRITE ALL RECORDS, RDR AND CHECK DATA
2701                                     ;ON ALL RECORDS, RDF AND CHECK DATA ON ALL RECORDS.)
2702 003763      123      122      106      .ASCII /SRF/      ;SPACE "N" RECORDS FORWARD.
2703 003766      123      122      122      .ASCII /SRR/      ;SPACE "N" RECORDS REVERSE.
2704 003771      122      116      122      .ASCII /RNR/      ;READ NEXT REVERSE. I.E., SPACE FWD READ REVERSE.
2705 003774      122      116      106      .ASCII /RNF/      ;READ NEXT FORWARD, I.E., READ FORWARD, SPACE REVERSE.
2706 003777      122      120      106      .ASCII /RPF/      ;READ PREVIOUS FORWARD. IE., SPACE REVERSE, READ FORWARD
2707 004002      122      120      122      .ASCII /RPR/      ;READ PREVIOUS REVERSE. IE., READ REVERSE, SPACE FORWARD
2708 004005      127      122      122      .ASCII /WRR/      ;WRITE RETRY.
2709 004010      122      127      104      .ASCII /RWD/      ;REWIND.
2710 004013      115      102      122      .ASCII /MBR/      ;MESSAGE BUFFER RELEASE
2711 004016      127      124      115      .ASCII /WTM/      ;WRITE TAPE MARK
2712 004021      127      124      122      .ASCII /WTR/      ;WRITE TAPE MARK RETRY.
2713 004024      123      106      106      .ASCII /SFF/      ;SPACE "N" FILES FORWARD.
2714 004027      123      106      122      .ASCII /SFR/      ;SPACE "N" FILES REVERSE.
2715 004032      107      105      123      .ASCII /GES/      ;GET EXTENDED STATUS.
2716 004035      105      122      123      .ASCII /ERS/      ;ERASE 3 INCHES OF TAPE.
2717 004040      125      116      114      .ASCII /UNL/      ;REWIND AND UNLOAD.
2718 004043      103      114      116      .ASCII /CLN/      ;CLEAN TAPE.
2719 004046      123      103      110      .ASCII /SCH/      ;SET CHARACTERISTICS. WHERE BRF=200, 40, 20, 0.
2720                                     ;SEE TK25 PROGRAMMING SPECIFICATION FOR DESCRIPTION.
2721 004051      104      111      101      .ASCII /DIA/      ;DIAGNOSTICS. SEE TK25 PROGRAMMING SPECIFICATION
2722                                     ;FOR DESCRIPTION. ODT MUST BE USED TO LOAD DIAGNOSTIC DATA
2723                                     ;INTO THE WRITE BUFFER BEFORE THIS CMD IS ISSUED.
2724 004054      112      115      120      .ASCII /JMP/      ;JUMP TO THE NTH COMMAND IN THE COMMAND
2725                                     ;SEQUENCE TABLE, WHERE N IS DEFINED IN
2726                                     ;THE # OF OPERATIONS.
2727 004057      104      114      131      .ASCII /DLY/      ;DELAY "N" MS, WHERE N IS DEFINED IN
2728                                     ;THE # OF OPERATIONS.
2729 004062      105      116      104      .ASCII /END/      ;END OF COMMAND SEQUENCE.
2730 .EVEN
2731
2732
2733

```

```

2735      .SBTTL GLOBAL TEXT SECTION
2736
2737      ;++
2738      ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
2739      ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
2740      ; MORE THAN ONE TEST.
2741      ;--
2742
2743
2750      ;
2751      ; FORMAT STATEMENTS USED IN PRINT CALLS
2752      ;
2753
2754      .NLIST BEX
2755
2756 004066      045      116      045 CODELM:: .ASCIZ /%N%UNIT %D1%A TK25 CODE LEVEL P%03%N%N/
2757      .EVEN
2758 004136      130      130      130 HALTM:: .ASCIZ /XXX CMD - TYPE <CR> TO CONTINUE/
2759 004176      103      115      104 CMDPKM:: .ASCIZ /CMD PACKET ADR NOT ON MODULO 4 BOUNDARY: RELOAD!/
2760      .EVEN
2761 004260      104      101      124 WTVRM:: .ASCIZ /DATA COMPARE ERROR/
2762 004303      116      117      040 TOERM:: .ASCIZ /NO TK25 RESPONSE/
2763 004324      125      116      104 SCERM:: .ASCIZ /UNDEFINED SPEC COND/
2764 004350      122      106      103 RFCERM:: .ASCIZ /RFC NON ZERO/
2765 004365      124      113      062 NSSRM:: .ASCIZ /TK25 NOT READY/
2766 004404      122      105      124 RLEXM:: .ASCIZ /RETRY LIMIT EXCEEDED/
2767 004431      125      116      111 ATTNM:: .ASCIZ /UNIT OFF LINE/
2768 004447      106      125      116 FUNRM:: .ASCIZ /FUNCTION REJECT/
2769 004467      106      101      124 FATSM:: .ASCIZ /FATAL SUBSYSTEM ERROR/
2770 004515      116      117      040 NOINTM:: .ASCIZ /NO INTERRUPT/
2771 004532      124      101      120 TSAM:: .ASCIZ /TAPE STATUS ALERT/
2772 004554      124      117      117 TOOMM:: .ASCIZ /TOO MANY INTERRUPTS/
2773 004600      103      101      120 RNYM:: .ASCIZ /CAPSTAN RUNAWAY-GET STATUS RESULTS:/
2774 004644      122      105      103 RERM:: .ASCIZ /RECOVERABLE ERROR/
2775 004666      125      116      122 URERM:: .ASCIZ /UNRECOVERABLE ERROR/
2776 004712      045      116      045 DROPDM:: .ASCIZ /%N%ADROPPED UNIT %D1%N/
2777 004741      045      116      045 AUDRPM:: .ASCIZ /%N%AALL UNITS DROPPED%N%N/
2778 004773      045      116      045 DTAER2:: .ASCIZ "%N%ABYTE:%D4%S2%AWAS:%B8%S2%AS/B:%B8%N"
2779 005042      045      104      064 DTAER3:: .ASCIZ "%D4%A BYTES IN ERROR OUT OF %D4%N"
2780 005104      045      101      116 DTAER4:: .ASCIZ /%ANO DATA READ%N/
2781 005125      045      101      122 DTAER5:: .ASCIZ /%ARECORD TOO LONG: >%04%A BYTES%N/
2782 005167      045      101      122 NURTY1:: .ASCIZ /%ARECOVERED ON RETRY #%D2%N/
2783 005223      045      101      125 OFLINM:: .ASCIZ /%AUNIT %D1%A OFF LINE%N/
2784 005253      045      101      107 GETSTM:: .ASCIZ /%AGET STATUS CMD RESULTS:%N/
2785 005307      045      116      000 CRLF:: .ASCIZ /%N/
2786 005312      045      116      045 CRLFSP:: .ASCIZ /%N%S7/
2787 005320      045      116      045 RAMFHR:: .ASCIZ '%N%A ***** CONTROLLER RAM DUMP *****'
2788 005377      045      116      045 RAMIOP:: .ASCIZ '%N%A RAM ADDRESS (OCTAL) = %03%A - %03%N'
2789 005450      045      101      040 RAMPD:: .ASCIZ '%A %03%A '
2790 005462      045      116      045 RAMLIN:: .ASCIZ '%N%N%N'
2791      .LIST BEX
2792      .EVEN
2793

```

```

2795      .SBTTL  GLOBAL ERROR REPORT SECTION
2796
2797      ;**
2798      ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX CALLS
2799      ; THAT ARE USED IN MORE THAN ONE TEST.  IT ALSO INCLUDES THE ASCII MESSAGES
2800      ; THAT ARE USED BY THE PRINTB AND PRINTX CALLS..
2801      ;--
2802
2803
2804      BGNMSG  DTAERM
2805      DTAERM::
2806      PRINTB  #STAER1,DEVTBL(R5),PASCNT(R5),RECCNT(R5)
2807
2808      MOV      RECCNT(R5),-(SP)
2809      MOV      PASCNT(R5),-(SP)
2810      MOV      DEVTBL(R5),-(SP)
2811      MOV      #STAER1, -(SP)
2812      MOV      #4, -(SP)
2813      MOV      SP,R0
2814      TRAP    C$PNTB
2815      ADD     #12,SP
2816
2817      PRINTB  #STAER7
2818
2819      MOV      #STAER7, -(SP)
2820      MOV      #1, -(SP)
2821      MOV      SP,R0
2822      TRAP    C$PNTB
2823      ADD     #4,SP
2824
2825      LET RECD := R2          ;SAVE R2          MOV      R2,RECD
2826      LET TIME1 := R3       ;SAVE R3          MOV      R3,TIME1
2827      LET TIME2 := R4       ;SAVE R4          MOV      R4,TIME2
2828      JSR PC,RECTAP        ;RETRIEVE RECORD READ
2829      LET R2 := RECD        ;RESTORE R2       MOV      RECD,R2
2830      LET RECD := R3       ;SAVE RECORD READ MOV      R3,RECD
2831      LET R3 := TIME1      ;RESTORE R3       MOV      TIME1,R3
2832      LET R4 := TIME2      ;RESTORE R4       MOV      TIME2,R4
2833      PRINTB  #STAER6,RECD  ;PRINT RECORD READ
2834
2835      MOV      RECD, -(SP)
2836      MOV      #STAER6, -(SP)
2837      MOV      #2, -(SP)
2838      MOV      SP,R0
2839      TRAP    C$PNTB
2840      ADD     #6,SP
2841
2842      EXIT    MSG
2843
2844      .WORD   J$JMP
2845      .WORD   L10002-2-.
2846
2847      .EVEN
2848      ENDMSG
2849      L10002:
2850
2851      TRAP    C$MSG
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000

```

2826	005640		BGNMSG STAERM		
	005640		STAERM::		
2827	005640		PRINTB #STAER1,DEVTBL(R5),PASCNT(R5),RECCNT(R5)		
	005640	016546			MOV RECCNT(R5),-(SP)
	005644	016546			MOV PASCNT(R5),-(SP)
	005650	016546			MOV DEVTBL(R5),-(SP)
	005654	012746			MOV #STAER1,-(SP)
	005660	012746			MOV #4,-(SP)
	005664	010600			MOV SP,R0
	005666	104414			TRAP C#PNTB
	005670	062706			ADD #12,SP
2828	005674		PRINTB #STAER7		
	005674	012746			MOV #STAER7,-(SP)
	005700	012746			MOV #1,-(SP)
	005704	010600			MOV SP,R0
	005706	104414			TRAP C#PNTB
	005710	062706			ADD #4,SP
2829	005714		LET R2 := CMDPKT CLR.BY #177740		
	005714	013702			MOV CMDPKT,R2
	005720	042702			BIC #177740,R2
2830	005724		LET R2 := R2 - #1		
	005724	005302			DEC R2
2831	005726		IF R2 EQ #0 THEN		
	005726	005702			TST R2
	005730	001016			BNE 50000\$
2832	005732	004737	JSR PC,RECTAP		
	005732	004737	LET RECD := R3		
2833	005736		PRINTB #STAER6,RECD		
	005736	010337			MOV R3,RECD
2834	005742				
	005742	013746			MOV RECD, -(SP)
	005746	012746			MOV #STAER6, -(SP)
	005752	012746			MOV #2, -(SP)
	005756	010600			MOV SP,R0
	005760	104414			TRAP C#PNTB
	005762	062706			ADD #6,SP
2835	005766		ENDIF		
	005766				50000\$:
2836	005766		PRINTX #STAER2		
	005766	012746			MOV #STAER2, -(SP)
	005772	012746			MOV #1, -(SP)
	005776	010600			MOV SP,R0
	006000	104415			TRAP C#PNTX
	006002	062706			ADD #4,SP
2837	006006		PRINTX #STAER3,CMDPKT,@TSDB(R5),MSGPKT*MS.RFC,TSSREG,CTCC		
	006006	013746			MOV CTCC, -(SP)
	006012	013746			MOV TSSREG, -(SP)
	006016	013746			MOV MSGPKT*MS.RFC, -(SP)
	006022	017546			MOV @TSDB(R5), -(SP)
	006026	013746			MOV CMDPKT, -(SP)
	006032	012746			MOV #STAER3, -(SP)
	006036	012746			MOV #6, -(SP)
	006042	010600			MOV SP,R0
	006044	104415			TRAP C#PNTX
	006046	062706			ADD #16,SP
2838	006052		PRINTX #STAER4,CMDPKT*2,CMDPKT*4,CMDPKT*6		
	006052	013746			MOV CMDPKT*6, -(SP)
	006056	013746			MOV CMDPKT*4, -(SP)


```

2860      .SBTTL GLOBAL SUBROUTINES SECTION
2861      ;**
2862      ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
2863      ; THAT ARE USED IN MORE THAN ONE TEST.
2864      ;--
2865
2866      ;      MODULES TO HANDLE TK25 INTERRUPTS.
2867
2868      006564      BGNSRV TS4IN0      ;DEVICE 0.
2869      006564      TS4IN0::      LET INTFLG := INTFLG * #1      ;SET INTERRUPT OCCURRED FLAG.
2870      006564      005237 003506      INC      INTFLG
2871      006570      ENDSRV #0
2872      006570      L10004:
2873      006570      142766 000340 000002      BICB #340,2(SP)
2874      006576      152766 000000 000002      BISB #0,2(SP)
2875      006604      000002      RTI
2876
2877      006606      BGNSRV TS4IN1      ;DEVICE 1.
2878      006606      TS4IN1::      LET INTFLG*2 := INTFLG*2 * #1      ;SET INTERRUPT OCCURRED FLAG.
2879      006606      005237 003510      INC      INTFLG*2
2880      006612      ENDSRV #0
2881      006612      L10005:
2882      006612      142766 000340 000002      BICB #340,2(SP)
2883      006620      152766 000000 000002      BISB #0,2(SP)
2884      006626      000002      RTI
2885
2886      006630      BGNSRV TS4IN2      ;DEVICE 2.
2887      006630      TS4IN2::      LET INTFLG*4 := INTFLG*4 * #1      ;SET INTERRUPT OCCURRED FLAG.
2888      006630      005237 003512      INC      INTFLG*4
2889      006634      ENDSRV #0
2890      006634      L10006:
2891      006634      142766 000340 000002      BICB #340,2(SP)
2892      006642      152766 000000 000002      BISB #0,2(SP)
2893      006650      000002      RTI
2894
2895      006652      BGNSRV TS4IN3      ;DEVICE 3.
2896      006652      TS4IN3::      LET INTFLG*6 := INTFLG*6 * #1      ;SET INTERRUPT OCCURRED FLAG.
2897      006652      005237 003514      INC      INTFLG*6
2898      006656      ENDSRV #0
2899      006656      L10007:
2900      006656      142766 000340 000002      BICB #340,2(SP)
2901      006664      152766 000000 000002      BISB #0,2(SP)
2902      006672      000002      RTI

```

```

2884
2885 ; SUBROUTINE TO RETRIEVE RECORD COUNT READ FROM TAPE FOR ERROR
2886 ; PRINTS.
2887 ; INPUTS:
2888 ; OUTPUTS: R3 = RECORD COUNT READ
2889 ; REGISTERS: R2, R3, R4
2890 ; CALLS:
2891
2892 006674 RECTAP::IF #MOD.CO SETIN CMDWRD THEN ;READ REV FETCH
006674 032737 000400 003426
006702 001430
2893 006704 LET R2 := MSGPKT*MS.RFC + DATARD ;FIND LAST READ AD.
006704 013702 002344
006710 063702 003416
2894 006714 IF #BIT00 SETIN R2 THEN ;ODD AD., REASSEMBLE
006714 032702 000001
006720 001417
2895 006722 LET R2 := R2 + #1 ;REC COUNT STARTING
006722 005202
2896 006724 LET R3 :B= (R2) CLR.BY #177400 ;WITH UPPER BYTE FETCH
006724 111203
006726 142703 177400
2897 006732 LET R3 := SWAP R3 ;
006732 000303
2898 006734 LET R2 := R2 - #1 ;LOWER BYTE AD.
006734 005302
2899 006736 IFB SWBFLG NE #0 THEN
006736 105737 003536
006742 001401
2900 006744 LET R2 := R2 - #1 ;LOWER BYTE AD.
006744 005302
2901 006746 ENDIF
006746
2902 006746 LET R4 :B= (R2) CLR.BY #177400 ;FETCH LOWER
006746 111204
006750 142704 177400
2903 006754 LET R3 := R3 OR R4 ;MERGE BYTES
006754 050403
2904 006756 ELSE
006756 000401
006760
2905 006760 LET R3 := (R2) ;EVEN AD. FETCH
006760 011203
2906 006762 ,ENDIF
006762
2907 006762 ELSE
006762 000402
006764
2908 006764 LET R3 := @DATARD ;READ FWD FETCH
006764 017703 174426
2909 006770 ENDIF
006770
2910
2911 006770 RTS PC

```

```

2914      ;      SUBROUTINE TO STORE A SET CHARACTERISTIC COMMAND AS
2915      ;      THE FIRST ENTRY IN THE SEQUENCE TABLE.
2916      ;      INPUTS:
2917      ;      OUTPUTS:
2918      ;      REGISTERS:
2919      ;      CALLS:
2920
2921 006772      SETCH:: LET R1 := #CMDSEQ      ;INIT COMMAND SEQUENCE TABLE POINTER.
      006772 012701 003554      MOV      #CMDSEQ,R1
2922 006776      MOV      #SCH,(R1)+      ;THIS CODE SETS UP A SET CHARACTERISTIC
      012721 140004      MOV      #DFTSCH,(R1)+ ;COMMAND AS THE FIRST COMMAND IN THE
2923 007002      MOV      #1,(R1)+      ;SEQUENCE TABLE.
      012721 000040      TST      (R1)+      ;SKIP PATTERN LOCATION.
2924 007006      RTS PC
      012721 000001
2925 007012      ;
      005721
2926 007014      ;
      000207
2927
2928
2929
2930
2931      ;      SUBROUTINE TO STORE A REWIND COMMAND IN THE SEQUENCE TABLE
2932      ;      INPUTS:
2933      ;      OUTPUTS:
2934      ;      REGISTERS:
2935      ;      CALLS:
2936
2937 007016      SETRW:: LET (R1)+ := #RWD      ;CMD = REWIND.
      007016 012721 102010      MOV      #RWD,(R1)+
2938 007022      LET (R1)+ := #1      ;BRF.
      012721 000001      MOV      #1,(R1)+
2939 007026      LET (R1)+ := #1      ;# OF OPERATIONS.
      012721 000001      MOV      #1,(R1)+
2940 007032      TST (R1)+      ;SKIP PATTERN.
      005721      MOV      #1,(R1)+
2941 007034      RTS PC      ;RETURN
      000207

```

```

2943      ; SUBROUTINE TO EXECUTE ALL COMMANDS IN THE SEQUENCE TABLE ON ALL
2944      ; DEVICES.
2945      ; INPUTS:
2946      ; OUTPUTS:          R2 = TERMINATION INDICATOR (0=END OF TABLE,1=EOT)
2947      ; REGISTERS:
2948      ; CALLS:           CMDAC,SETUP,EXSUB,CKHAE,NEXTU,FIRSTU,VFYDAT.
2949
2950 007036 EXALL:: LET R1 := #CMDSEQ          ;INIT SEQUENCE TABLE POINTER.
007036 012701 003554          MOV #CMDSEQ,R1
2951 007042          WHILE (R1) NE #END DO ;WHILE THERE ARE CMDS IN THE SEQUENCE TABLE.
007042          50006$:
007042 021127 177777          CMP (R1),#END
007046 001527          BEQ 50007$
2952 007050          JSR PC,SETUP          ;GO SETUP THE COMMAND BLOCK.
2953 007054          WHILE NCNT LT NCNT1 DO ;WHILE THERE ARE RECORDS REMAINING:
007054          50010$:
007054 023737 003420 003422          CMP NCNT,NCNT1
007062 002116          BGE 50011$
2954 007064          JSR PC,CMDAC          ;STORE CMD ASCII IN ERROR MESSAGE.
2955 007070          IFB RANDOM NE #0 THEN ;IF IN RANDOM MODE:
007070 105737 003533          TSTB RANDOM
007074 001435          BEQ 50012$
2956 007076          IF CMDWRD EQ #WRT THEN ;IF CMD IS A WRITE THEN:
007076 023727 003426 104005          CMP CMDWRD,#WRT
007104 001031          BNE 50013$
2957 007106          IFB VFYFLG EQ #0 THEN ;IF DATA IS NOT TO BE VERIFIED THEN:
007106 105737 003534          TSTB VFYFLG
007112 001026          BNE 50014$
2958 007114          LET RANB := RANB + RANS ;GENERATE
007114 063737 003442 003440          ADD RANS,RANB
2959 007122          LET RANS := RANS + RANB ;RANDOM
007122 063737 003440 003442          ADD RANB,RANS
2960 007130          LET BRFCNT := RANS ;LENGTH
007130 013737 003442 003424          MOV RANS,BRFCNT
2961 007136          LET BRFCNT := BRFCNT CLP.BY LENMSK ;MASK RANDOM LENGTH.
007136 043737 003436 003424          BIC LENMSK,BRFCNT
2962 007144          IF BRFCNT LT #18. THEN ;DO NOT ALLOW BYTE COUNT OF LESS THAN 18.
007144 023727 003424 000022          CMP BRFCNT,#18.
007152 002003          BGE 50015$
2963 007154          LET BRFCNT := #18. ;CHANGE COUNT OF 0-17 TO 18.
007154 012737 000022 003424          MOV #18.,BRFCNT
2964 007162          ENDIF
007162          50015$:
2965 007162          LET CMDPKT+CP.CNT := BRFCNT ;MOVE BRFCNT TO CMD PACKET.
007162 013737 003424 002322          MOV BRFCNT,CMDPKT+CP.CNT
2966 007170          ENDIF
007170          50014$:
2967 007170          ENDIF
007170          50013$:
2968 007170          ENDIF
007170          50012$:
2969 007170 004737 007332          JSR PC,EXSUB          ;ISSUE CMD TO ALL,AWAIT INTS,CHECK STATUS.
2970 007174 004737 020274          JSR PC,CKHAE          ;CHECK HALT AFTER EACH CMD FLAG.
2971 007200          LET R2 := #1          ;SET ALL UNITS AT BOT/EOT.
007200 012702 000001          MOV #1,R2
2972 007204 004737 017666          JSR PC,FIRSTU          ;FIND FIRST UNIT.
2973 007210          WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE UNITS:

```

```

007210
007210 026527 002536 177777
007216 001426
2974 007220
007220 032737 000400 003426
007226 001406
2975 007230
007230 032765 000002 003516
007236 001001
2976 007240
007240 005002
2977 007242
007242
2978 007242
007242 000411
007244
2979 007244
007244 032765 000001 003516
007252 001404
007254 032737 000001 003426
007262 001001
007264
2980
2981 007264
007264 005002
2982 007266
007266
2983 007266
007266
2984 007266 004737 017734
2985 007272
007272 000746
007274
2986 007274
007274 020227 000001
007300 001001
2987 007302 000412
2988 007304
007304
2989 007304
007304 005237 003420
2990 007310
007310 013737 003426 003432
2991 007316
007316 000656
007320
2992 007320 004737 016530
2993
2994 007324
007324 000646
007326
2995 007326
007326 005002
2996 007330 000207
2997
2998
2999

```

```

IF #MOD.CO SETIN CMDWRD THEN ;IF CMD IS REVERSE THEN:
    IF #X0.BOT NOTSETIN EOTFLG(R5) THEN ;IF NOT AT BOT THEN:
        LET R2 := #0
    ENDIF
ELSE
    ;ELSE IF CMD IS NOT REVERSE:
        IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN
            LET R2 := #0
        ENDIF
        JSR PC,NEXTU
    ENDDO
    IF R2 EQ #1 THEN
        BR EXARTN
    ENDIF
    LET NCNT := NCNT + #1
    LET PCMDWD := CMDWRD
    ENDDO
    JSR PC,VFYDAT
    ENDDO
    LET R2 := #0
EXARTN: RTS PC

```

```

50016$:
CMP DEVTBL(R5),#END
BEQ 50017$
BIT #MOD.CO,CMDWRD
BEQ 50020$
BIT #X0.BOT,EOTFLG(R5)
BNE 50021$
;CLEAR EOT/BOT FLAG.
CLR R2
50021$:
;ELSE IF CMD IS NOT REVERSE:
BR 50022$
50020$:
IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN
    BIT #X0.EOT,EOTFLG(R5)
    BEQ 50023$
    BIT #CMD.CO,CMDWRD
    BNE 50024$
50023$:
;IF NOT AT EOT OR NOT A MOTION CMD THEN:
;CLEAR EOT/BOT FLAG.
CLR R2
50024$:
50022$:
BR 50016$
50017$:
;IF ALL UNIT ARE AT EOT/BOT THEN:
CMP R2,#1
BNE 50025$
;RETURN WITH R2 = #1.
50025$:
;UPDATE RECORD COUNT.
INC NCNT
;SAVE PREVIOUS COMMAND WORD.
MOV CMDWRD,PCMDWD
BR 50010$
50011$:
;IF LAST CMD WAS A WRITE VERIFY, THEN GO
;VERIFY THE LAST N RECORDS OF DATA.
BR 50006$
50007$:
;SET NORMAL RETURN INDICATOR.
CLR R2
;RETURN.

```

```

3000 ; SUBROUTINE TO ISSUE COMMAND TO ALL DEVICES, WAIT FOR
3001 ; ALL INTERRUPTS, AND CHECK ALL STATUS.
3002 ; INPUTS:
3003 ; OUTPUTS:
3004 ; REGISTERS:
3005 ; CALLS: EXCUTE,GOWAIT,NEXTU,FIRSTU.
3006
3007 007332 004737 017666 EXSUB:: JSR PC,FIRSTU ;SET UP FOR FIRST UNIT.
3008 007336 WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
007336 026527 002536 177777 50026$:
007344 001465 CMP DEVTBL(R5),#END
3009 007346 IF #MOD.CO SETIN CMDWRD THEN ;IF CMD IS REVERSE THEN:
007346 032737 000400 003426 BEQ 50027$
007354 001421 BIT #MOD.CO,CMDWRD
3010 007356 IF #X0.BOT NOTSETIN EOTFLG(R5) THEN ;IF NOT AT BOT
007356 032765 000002 003516 BEQ 50030$
007364 001014 BIT #X0.BOT,EOTFLG(R5)
3011 007366 IF #X0.EOT SETIN EOTFLG(R5) THEN ;BUT IF AT EOT
007366 032765 000001 003516 BNE 50031$
007374 001406 BIT #X0.EOT,EOTFLG(R5)
3012 007376 IFB ALLEOT NE #0 THEN ;AND ALL OTHERS AT EOT
007376 105737 003543 TSTB ALLEOT
007402 001402 BEQ 50032$
3013 007404 004737 010774 JSR PC,EXCUTE ;THEN EXECUTE REV CMD
3014 007410 ENDIF ;IF NOT ALL AT EOT, FREEZE UNIT(S) AT EOT
007410 ELSE ;IF NOT AT BOT AND
3015 007410 000402 BR 50034$
007412 JSR PC,EXCUTE 50032$:
3016 007412 004737 010774 ENDIF ;NOT AT EOT, EXEC REV CMD
007416 ENDIF 50034$:
3018 007416 ELSE ;ELSE IF CMD IS NOT REVERSE:
007416 000435 BR 50035$
007420 IF CMDLG EQ #2 AND #X0.BOT SETIN EOTFLG(R5) THEN 50030$:
3020 007420 023727 003434 000002 CMP CMDLG,#2
007426 001011 BNE 50036$
007430 032765 000002 003516 BIT #X0.BOT,EOTFLG(R5)
007436 001405 BEQ 50036$
3021 ;CLEAR BAD SPOT COUNTS WHEN WRITING FROM BOT
3022 007440 LET BTPT := BTADDR(R5)
007440 016537 002550 003526 MOV BTADDR(R5),BTPT
3023 007446 LET @BTPT := #0
007446 005077 174054 CLR @BTPT
3024 007452 ENDIF
007452 IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN 50036$:
3025 007452 032765 000001 003516 BIT #X0.EOT,EOTFLG(R5)
007460 001404 BEQ 50037$
007462 032737 000001 003426 BIT #CMD.CO,CMDWRD
007470 001003 BNE 50040$
007472 50037$:
3026 ;IF NOT AT EOT OR NOT A MOTION CMD THEN:

```

```

3027 007472 004737 010774      JSR PC,EXCUTE      ;ISSUE CMD TO TK25.
3028 007476      ELSE
      007476 000405
      007500
3029 007500      IFB ALLEOT NE #0 THEN
      007500 105737 003543
      007504 001402
3030 007506 004737 010774      JSR PC,EXCUTE
3031 007512      ENDIF
      007512
3032 007512      ENDIF
      007512
3033 007512      ENDIF
      007512
3034 007512 004737 017734      JSR      PC,NEXTU      ;FIND NEXT UNIT IN TEST CYCLE.
3035 007516      ENDDO
      007516 000707
      007520
3036 007520      IFB RPTFLG NE #0 THEN      ;IF REPORT HAS BEEN REQUESTED THEN:
      007520 105737 003535
      007524 001403
      LET RPTFLG :B= #0      ;CLR THE FLAG,
      DORPT      ;PRINT THE PERFORMANCE REPORT.
3037 007526      ENDIF      ;
      007526 105037 003535
      007532 104424
3038 007532      JSR PC,FIRSTU      ;SET UP FOR FIRST UNIT.
3039 007534      WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
      007534 004737 017666
      007540
      007540 026527 002536 177777
      007546 001450
3042 007550      IF #MOD.CO SETIN CMDWRD THEN ;IF CMD IS REVERSE THEN:
      007550 032737 000400 003426
      007556 001421
      IF #X0.BOT NOTSETIN EOTFLG(R5) THEN ;IF NOT AT BOT
3043 007560      ENDIF
      007560 032765 000002 003516
      007566 001014
      IF #X0.EOT SETIN EOTFLG(R5) THEN
3044 007570      ENDIF
      007570 032765 000001 003516
      007576 001406
      IFB ALLEOT NE #0 THEN
3045 007600      ENDIF
      007600 105737 003543
      007604 001402
3046 007606 004737 011470      JSR PC,GOWAIT
3047 007612      ENDIF
      007612
3048 007612      ELSE
      007612 000402
      007614
3049 007614 004737 011470      JSR PC,GOWAIT
3050 007620      ENDIF
      007620
3051 007620      ENDIF
      007620
3052 007620      ELSE      ;ELSE IF CMD IS FORWARD:
      007620 000420

```

BR 50041\$
50040\$:TSTB ALLEOT
BEQ 50042\$

50042\$:

50041\$:

50035\$:

BR 50026\$
50027\$:TSTB RPTFLG
BEQ 50043\$CLRB RPTFLG
TRAP C\$DRPT

50043\$:

50044\$:

CMP DEVTBL(R5),#END
BEQ 50045\$BIT #MOD.CO,CMDWRD
BEQ 50046\$BIT #X0.BOT,EOTFLG(R5)
BNE 50047\$;BUT IF AT EOT
BIT #X0.EOT,EOTFLG(R5)
BEQ 50050\$;AND ALL OTHERS AT EOT
TSTB ALLEOT
BEQ 50051\$;THEN WAIT FOR CMD END
;IF NOT ALL AT EOT, DO NOT WAIT
50051\$:;NOT AT BOT, AND NOT AT EOT
BR 50052\$

50050\$:

;WAIT FOR INT,CHECK STATUS.

50052\$:

50047\$:

FORWARD:
BR 50053\$

```

007622
3053 007622 032765 000001 003516 IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN
007622 001404 BIT #X0.EOT,EOTFLG(R5)
007630 001404 BEQ 50054$
007632 032737 000001 003426 BIT #CMD.CO,CMDWRD
007640 001003 BNE 50055$
007642 50054$:
3054 ;IF NOT AT EOT OR NOT A MOTION CMD THEN:
3055 007642 004737 011470 JSR PC,GOWAIT ;WAIT FOR INT,CHECK STATUS.
3056 007646 000405 ELSE
007646 000405 BR 50056$
007650 50055$:
3057 007650 IFB ALLEOT NE #0 THEN TSTB ALLEOT
007650 105737 003543 BEQ 50057$
007654 001402
3058 007656 004737 011470 JSR PC,GOWAIT
3059 007662 ENDF
007662 50057$:
3060 007662 ENDF 50056$:
3061 007662 ENDF
007662 50053$:
3062 007662 004737 017734 JSR PC,NEXTU ;FIND NEXT UNIT IN TEST CYCLE.
3063 007666 ENDDO
007666 000724 BR 50044$
007670 50045$:
3064 007670 000207 RTS PC ;RETURN.

```



```

3066 ; THIS SUBROUTINE STORES THE ASCII FOR THE CURRENT COMMAND AND PREVIOUS
3067 ; COMMAND IN THE STANDARD ERROR MESSAGE. ON ENTRY LOCATION CMDWRD
3068 ; CONTAINS CURRENT CMD AND LOCATION PCMDWD CONTAINS PREVIOUS CMD.
3069 ; INPUTS:
3070 ; OUTPUTS:
3071 ; REGISTERS: R3, R4.
3072 ; CALLS: GCMDA
3073
3074 007672 013704 003426 CMDAC:: LET R4 := CMDWRD ;R4 = CMD BINARY.
007672 004737 007746 ;R4 = CMD BINARY. MOV CMDWRD,R4
3075 007676 004737 007746 JSR PC,GCMDA ;GET CMD ASCII.
3076 007702 112337 006154 MOVB (R3)+,STAER1+2 ;MOVE CMD ASCII
3077 007706 112337 006155 MOVB (R3)+,STAER1+3 ;
3078 007712 111337 006156 MOVB (R3),STAER1+4 ;INTO MSG.
3079 007716 013704 003432 LET R4 := PCMDWD ;R4 = PREVIOUS CMD BINARY.
007716 004737 007746 ;R4 = PREVIOUS CMD BINARY. MOV PCMDWD,R4
3080 007722 004737 007746 JSR PC,GCMDA ;GET CMD ASCII.
3081 007726 000240 NOP ;
3082 007730 112337 006270 LET STAER7+24 :B= (R3)+ ;MOVE CMD ASCII
007730 112337 006270 ; MOVB (R3)+,STAER7+24
3083 007734 112337 006271 LET STAER7+25 :B= (R3)+ ;
007734 112337 006271 ; MOVB (R3)+,STAER7+25
3084 007740 111337 006272 LET STAER7+26 :B= (R3) ;INTO MSG.
007740 000207 006272 ; MOVB (R3),STAER7+26
3085 007744 000207 RTS PC ;RETURN. GO EXECUTE NEXT FUNCTION.
3086
3087
3088
3089 ; SUBROUTINE TO FIND THE ASCII EQUIVILENT OF THE COMMAND IN R4.
3090 ; ADDRESS OF ASCII 1ST WORD IS RETURNED IN R3.
3091 ; INPUTS: R4 = PRESENT COMMAND WORD.
3092 ; OUTPUTS: R3 = ADDRESS OF PRESENT COMMAND ASCII.
3093 ; REGISTERS:
3094 ; CALLS:
3095
3096 007746 005003 GCMDA:: LET R3 := #0 ;INIT CMD TBL POINTER.
007746 005003 ;INIT CMD TBL POINTER. CLR R3
3097 007750 007750 WHILE CMTBL(R3) NE R4 DO ;UNTIL CURRENT CMD IS FOUND:
007750 026304 003656 ;UNTIL CURRENT CMD IS FOUND: 50060$:
007754 001403 ;UNTIL CURRENT CMD IS FOUND: CMP CMTBL(R3),R4
3098 007756 062703 000002 LET R3 := R3 + #2 ;SEARCH CMD TABLE. BEQ 50061$
007756 062703 000002 ;SEARCH CMD TABLE. ADD #2,R3
3099 007762 000772 ENDDO ; BR 50060$
007762 000772 ; BR 50061$:
3100 007764 010304 LET R4 := R3 ;
007764 010304 ; MOV R3,R4
3101 007766 006203 LET R3 := R3 SHIFT -1 ;POINT TO ASCII FOR THAT COMMAND
007766 006203 ;POINT TO ASCII FOR THAT COMMAND ASR R3
3102 007770 060403 ADD R4,R3
3103 007772 062703 003744 ADD #CMDASC,R3
3104 007776 000207 RTS PC ;RETURN.

```

```

3106 ; THIS SUBROUTINE LOADS THE TK25 COMMAND PACKET FROM ONE
3107 ; ENTRY IN THE SEQUENCE TABLE.
3108 ; INPUTS:
3109 ; OUTPUTS:
3110 ; REGISTERS: R2, R3.
3111 ; CALLS: GENPAT.
3112
3113 010000 SETUP:: LET CMDLG := #0 ;CLR CMD LOGGING CODE(DISABLES LOGGING)
      010000 005037 003434 ; CLR CMDLG
3114 010004 012137 002314 MOV (R1)+,CMDPKT ;LOAD THE COMMAND WORD.
3115 010010 011137 002322 MOV (R1),CMDPKT+CP.CNT ;LOAD THE BYTE/RECORD/FILE COUNT.
3116 010014 011137 003424 MOV (R1),BRFCNT ;SAVE BRF FOR THIS COMMAND.
3117 010020 013702 002314 MOV CMDPKT,R2 ;GET CMD.
3118 010024 042702 177740 BIC #NCMD.C,R2 ;CLR ALL BUT CMD BITS.
3119 010030 010203 MOV R2,R3 ;SAVE IT TWICE.
3120 010032 162703 000010 SUB #CMD.C3,R3 ;POSITION COMMAND?
3121 010036 001003 BNE 2$ ;BR IF NOT.
3122 010040 011137 002316 MOV (R1),CMDPKT+2 ;MOVE BPCR IN 2ND PKT WORD FOR POSITION CMD.
3123 010044 000461 BR 3$
3124 010046 2$: IF CMDPKT EQ #WTM THEN ;IF CMD IS A WRITE TAPE MARK THEN:
      010046 023727 002314 100011 CMP CMDPKT,#WTM
      010054 001003 BNE 50062$
3125 010056 LET CMDLG := #2 ;WTM LOGGING CODE IS 2.
      010056 012737 000002 003434 MOV #2,CMDLG
3126 010064 ENDF
      010064 50062$:
3127 010064 010203 MOV R2,R3
3128 010066 162703 000001 SUB #CMD.CO,R3 ;IS IT A READ?
3129 010072 001017 BNE 1$ ;BR IF NOT.
3130 010074 013737 003416 002316 MOV DATARD,CMDPKT+CP.ADL ;IF SO, LOAD THE BUFFER ADDR.
3131 010102 IF #MOD.CO SETIN CMDPKT THEN ;IF CMD IS A READ REV THEN:
      010102 032737 000400 002314 BIT #MOD.CO,CMDPKT
      010110 001404 BEQ 50063$
3132 010112 LET CMDLG := #4 ;LOGGING CODE IS 4.
      010112 012737 000004 003434 MOV #4,CMDLG
3133 010120 ELSE ;ELSE - IF CMD IS A READ FWD:
      010120 000403 BR 50064$
      010122 50063$:
3134 010122 LET CMDLG := #6 ;LOGGING CODE IS 6.
      010122 012737 000006 003434 MOV #6,CMDLG
3135 010130 ENDF
      010130 50064$:
3136 010130 000427 BR 3$ ;CONTINUE.
3137 010132 010203 1$: MOV R2,R3 ;IS IT
3138 010134 162703 000004 SUB #CMD.C2,R3 ;A SET CHARACTERISTICS CMD?
3139 010140 001011 BNE 4$ ;BR IF NOT.
3140 010142 LET CMDPKT+CP.ADL := #SCHBK ;SET UP ADR LO FOR SET CHAR.
      010142 012737 002446 002316 MOV #SCHBK,CMDPKT+CP.ADL
3141 010150 012737 000010 002322 MOV #SCHCNT,CMDPKT+CP.CNT ;SET BUFFER EXTENT
3142 010156 LET SCHBK+6 := (R1) ;STORE CHARACTERISTIC CODE IN SCH BLOCK.
      010156 011137 002454 MOV (R1),SCHBK+6
3143 010162 000412 BR 3$ ;CONTINUE.
3144 010164 010203 4$: MOV R2,R3 ;IS IT
3145 010166 162703 000006 SUB #CMD.C1!CMD.C2,R3 ;A DIAGNOSTIC (DIA) CMD?
3146 010172 001006 BNE 3$ ;BR IF NOT.
3147 010174 012737 000020 002322 MOV #DIACNT,CMDPKT+CP.CNT ;LOAD BUFFER EXTENT.
3148 010202 012737 003414 002316 MOV #DIABLK,CMDPKT+CP.ADL ;LOAD BUFFER ADR LOW.

```

```

3149 010210 005721          3$:  TST      (R1)+          ;POINT TO N (NUMBER OF TIMES TO EXECUTE THIS INSTRUC
3150 010212          LET NCNT1 := (R1)+    ;SAVE NUMBER OF OPERATIONS
      010212 012137 003422          MOV      (R1)+,NCNT1
3151 010216          LET NCNT := #0          ;CLEAR OPERATION COUNTER.
      010216 005037 003420          CLR      NCNT
3152 010222 012137 003454          MOV      (R1)+,PATERN    ;SAVE PATTERN CODE FOR CURRENT CMD.
3153 010226 010203          MOV      R2,R3          ;IS IT
3154 010230 162703 000005          SUB      #CMD.CO!CMD.C2,R3 ;A WRITE?
3155 010234 001010          BNE      5$          ;BR IF NOT.
3156 010236 013737 003414 002316          MOV      DATAW,CMDPKT+CP.ADL ;LOAD WRITE BUFFER LO ORDER.
3157 010244 004737 010356          JSR      PC,GENPAT      ;GO GENERATE THE WRITE PATTERN.
3158 010250          LET CMDLG := #2          ;WRITE LOGGING CODE IS 2.
      010250 012737 000002 003434          MOV      #2,CMDLG
3159 010256          5$:  IF #VFY.C SETIN CMDPKT THEN ;IF DATA VERIFICATION IS REQUIRED:
      010256 032737 000100 002314          BIT      #VFY.C,CMDPKT
      010264 001407          BEQ      50065$
3160 010266          LET VFYFLG :B= #1          ;SET VERIFY FLAG.
      010266 112737 000001 003534          MOV      #1,VFYFLG
3161 010274 042737 000100 002314          BIC      #VFY.C,CMDPKT ;CLEAR VERIFY BIT(NOT USED BY HARDWARE).
3162 010302          ELSE          ;IF DATA VERIFICATION IS NOT REQUIRED:
      010302 000402          BR      50066$
      010304          BR      50065$
3163 010304          LET VFYFLG :B= #0          ;CLR VERIFY FLAG.
      010304 105037 003534          CLR      VFYFLG
3164 010310          ENDIF
      010310          BR      50066$
3165 010310          LET PCMDWD := CMDWRD          ;SAVE PREVIOUS CMD WORD.
      010310 013737 003426 003432          MOV      CMDWRD,PCMDWD
3166 010316          LET CMDWRD := CMDPKT          ;SAVE PRESENT CMD WORD.
      010316 013737 002314 003426          MOV      CMDPKT,CMDWRD
3167 010324          IFB SWBFLG NE #0 THEN          ;IF SWAP BYTES IS ENABLED:
      010324 105737 003536          TST      SWBFLG
      010330 001403          BEQ      50067$
3168 010332          LET CMDPKT := CMDPKT SET.BY #SWB.C ;SET SWAP BIT IN COMMAND.
      010332 052737 010000 002314          BIS      #SWB.C,CMDPKT
3169 010340          ENDIF
      010340          BR      50067$
3170 010340          BIC      #BRF.C,CMDPKT          ;CLR BRF BIT (INTERNAL ONLY).
3171 010346          LET CMDSAV := CMDPKT          ;SAVE 1ST WORD OF COMMAND PACKET.
      010346 013737 002314 003430          MOV      CMDPKT,CMDSAV
3172 010354 000207          RTS      PC          ;RETURN.

```

```

3174 ; THIS SUBROUTINE SETS UP AND CALLS THE APPROPRIATE SUBROUTINE TO GENERATE
3175 ; THE DESIRED PATTERN FOR THE WRITE AND WRITE/VERIFY COMMANDS.
3176 ; INPUTS:
3177 ; OUTPUTS:
3178 ; REGISTERS: R2, R3, R4.
3179 ; CALLS: PATRO - PATR7
3180
3181 010356 GENPAT:: LET R3 := PATERN SHIFT 1 ;SETUP PATTERN ROUTINE POINTER
010356 013703 003454 MOV PATERN,R3
010362 006303 ASL R3
3182 010364 LET R4 := BRFCNT + #1 ;SET LENGTH OF WRITE BFR
010364 013704 003424 MOV BRFCNT,R4
010370 005204 INC R4
3183 010372 LET R4 := R4 CLR.BY #1 ;ROUNDED UP TO NEXT WORD
010372 042704 000001 BIC #1,R4
3184 010376 LET R4 := R4 - #2 ;WITH FIRST WORD RESERVED
010376 162704 000002 SUB #2,R4
3185 010402 LET R2 := DATAWT + #2 ;FOR RECORD COUNT
010402 013702 003414 MOV DATAWT,R2
010406 062702 000002 ADD #2,R2
3186 010412 JSR PC,@PATTBL(R3) ;GO GENERATE THE APPROPRIATE PATTERN.
3187 010416 000207 RTS PC ;RETURN TO SETUP SUBROUTINE.
3188
3189 ;TK25 WRITE PATTERN LOOKUP TABLE. USED TO JSR TO THE
3190 ;CORRECT DATA PATTERN GENERATING ROUTINE.
3191
3192 010420 010442 PATTBL: PATRO ; INCREMENTING PATTERN, 0 - 377
3193 010422 010500 PATR1 ; ALL ONES PATTERN
3194 010424 010520 PATR2 ; ALL ZEROES PATTERN
3195 010426 010530 PATR3 ; '1' BIT SHIFT, RIGHT TO LEFT
3196 010430 010554 PATR4 ; '0' BIT SHIFT, RIGHT TO LEFT
3197 010432 010566 PATR5 ; ALTERNATE '0' & '1' WITH ALT. BYTES COMPL.
3198 010434 010600 PATR6 ; ALTERNATE BYTES OF 000 AND 377
3199 010436 010620 PATR7 ; RANDOM PATTERN.
3200 010440 010744 PATR8 ; DUMMY. NO PATTERN, JUST EXITS.
3201
3202
3203 ;INCREMENTING PATTERN. 0 - 377.
3204
3205 010442 PATRO:: LET R3 := # '0'
010442 012703 000400 MOV #400,R3
3206 010446 162704 000002 1$: LET R4 := R4 - #2 ;DECREMENT WORD COUNT. SUB #2,R4
010446 100411 BMI 2$ ;BR IF DONE.
3207 010452 100411 LET (R2)+ := R3 ;STORE DATA WORD.
3208 010454 010322 MOV R3,(R2)+
010454 010322 LET R3 := R3 + #1002 ;UPDATE PATTERN. ADD #1002,R3
3209 010456 062703 001002 IF R3 EQ #1000 THEN ;IF PATTERN HAS WRAPPED AROUND THEN:
010462 020327 001000 CMP R3,#1000
010466 001002 BNE 50070$
3211 010470 LET R3 := #400 ;INIT THE PATTERN AGAIN. MOV #400,R3
010470 012703 000400
3212 010474 ENDIF
010474
3213 010474 000764 2$: BR 1$ ;DO IT AGAIN.
3214 010476 000207 RTS PC ;RETURN.

```

```
3215
3216
3217
3218 010500 012703 177777
3219 010504 162704 000002
3220 010510 100402
3221 010512 010322
3222 010514 000773
3223
3224 010516 000207

;ALL ONE'S PATTERN.
PATR1:: MOV #1,R3
ZROPAT: LET R4 := R4 - #2

;ALL ONES PATTERN;.
;DECREMENT BYTE COUNT.
SUB #2,R4

;DONE?,BR IF YES.
;IF NOT LOAD NEXT BYTE WITH PATTERN.
;DO IT AGAIN.

1$: RTS PC
;RETURN.
```

```

3226                                     ;ALL ZEROES PATTERN.
3227
3228 010520 005003
3229 010522 004737 010504 PATR2:: CLR R3 ;CLR PATTERN REGISTER.
3230 010526 000207 JSR PC,ZROPAT ;GO GENERATE IT.
RTS PC ;RETURN.
3231
3232                                     ;ONE BIT WALKING FROM R TO L IN A FIELD OF ZEROES.
3233
3234 010530 012703 000401 PATR3:: MOV #401,R3 ;INIT PATTERN REGISTER.
3235 010534 162704 000002 WLKZRO: LET R4 := R4 - #2 ;DECREMENT WORD COUNT.
010534 162704 000002 SUB #2,R4
3236 010540 100404 BMI 1# ;BR IF DONE.
3237 010542 010322 MOV R3,(R2)+ ;LOAD DATA.
3238 010544 006303 ASL R3 ;SHIFT PATTERN.
3239 010546 005503 ADC R3 ;ADD CARRY BACK INTO PATTERN.
3240 010550 000771 BR WLKZRO ;DO IT AGAIN.
3241 010552 000207 1#: RTS PC ;RETURN.
3242
3243                                     ;ZERO BIT WALKING FROM R TO L IN A FIELD OF 1'S.
3244
3245 010554 012703 177376 PATR4:: MOV #177376,R3 ;INIT PATTERN REGISTER.
3246 010560 004737 010534 JSR PC,WLKZRO ;GO GENERATE ;IT.
3247 010564 000207 RTS PC ;RETURN.
3248
3249                                     ;ALTERNATING ONE AND ZERO BITS WITH ALTERNATE BYTES
3250                                     ;COMPLEMENTED.
3251
3252 010566 012703 125125 PATR5:: MOV #125125,R3 ;INIT PATTERN REGISTER.
3253 010572 004737 010504 JSR PC,ZROPAT ;GO GENERATE IT.
3254 010576 000207 RTS PC ;RETURN.
3255
3256                                     ;ALTERNATING BYTES OF 000 AND 377.
3257
3258 010600 012703 177400 PATR6:: MOV #177400,R3 ;INIT PATTERN REGISTER.
3259 010604 162704 000002 1#: LET R4 := R4 - #2 ;DECREMENT WORD COUNT.
010604 162704 000002 SUB #2,R4
3260 010610 100402 BMI 2# ;BR IF DONE.
3261 010612 010322 MOV R3,(R2)+ ;LOAD DATA.
3262 010614 000773 BR 1# ;DO IT AGAIN.
3263 010616 000207 2#: RTS PC ;RETURN.
3264
3265                                     ;RANDOM PATTERN GENERATOR
3266
3267 010620 PATR7::
3268 010620 012737 001233 010736 MOV #RS1,RAN1 ;SET UP THE SEED
3269 010626 012737 007622 010740 MOV #RS2,RAN2 ;SET UP THE SEED
3270 010634 012737 000000 010742 MOV #RS3,RAN3 ;SET UP THE SEED
3271 010642 162704 000002 1#: LET R4 := R4 - #2 ;DECREMENT WORD COUNT
010642 162704 000002 SUB #2,R4
3272 010646 100432 BMI GIT ;BR IF DONE.
3273 010650 010346 MOV R3,-(SP)
3274 010652 013703 010736 MOV RAN1,R3 ;MOVE THE FIRST SEED INTO R3
3275 010656 000241 CLC ;CLEAR THE CARRY FLAG
3276 010660 005337 010742 DEC RAN3 ;DECREMENT THE THIRD SEED
3277 010664 006103 ROL R3 ;
3278 010666 006103 ROL R3 ;
3279 010670 063703 010740 ADD RAN2,R3 ;ADD THE SECOND SEED TO R3

```

```

3280 010674 010337 010736      MOV    R3,RAN1      ;PUT IT ALL IN THE FIRST SEED
3281 010700 063703 010742      ADD    RAN3,R3      ;PUT THE THIRD SEED INTO R3
3282 010704 006103              ROL    R3           ;
3283 010706 006103              ROL    R3           ;
3284 010710 063703 010740      ADD    RAN2,R3      ;ADD THE SECOND SEED TO R3
3285 010714 006103              ROL    R3           ;
3286 010716 006103              ROL    R3           ;
3287 010720 010337 010740      MOV    R3,RAN2      ;PUT IT IN THE SECONG SEED
3288 010724 012603              MOV    (SP),R3      ;RESTORE R3
3289 010726 013722 010740      MOV    RAN2,(R2).   ;PUT # IN BUFFER
3290 010732 000743              BR     1$           ;CONTINUE
3291 010734 000207      GIT:   RTS    PC           ;RETURN
3292
3293 010736 000000      RAN1:: .WORD    0
3294 010740 000000      RAN2:: .WORD    0
3295 010742 000000      RAN3:: .WORD    0
3296          001233      RS1   ==    1233
3297          007622      RS2   ==    7622
3298          000000      RS3   ==    0
3299
3300          ;          NO PATTERN GENERATION.
3301
3302 010744 000207      PATR8:: RTS    PC           ;RETURN.
3303
3304 010746      BGNSRV CLKSV
3305 010746      CLKSVC:: LET TIME1 := TIME1 - #1
3306 010746 005337 003444      LET @LCSR := #0          DEC    TIME1
3307 010752 005077 172506      ENDSRV #0              CLR    @LCSR
3307 010756      L10010:
3307 010756 142766 000340 000002      BICB   #340,2(SP)
3307 010764 152766 000000 000002      BISB   #0,2(SP)
3307 010772 000002      RTI
    
```

```

3309      ;      THIS SUBROUTINE INITIATES TK25 COMMAND EXECUTION
3310      ;      AND CHECKS FOR TK25 RESPONSE.
3311      ;      INPUTS:
3312      ;      OUTPUTS:
3313      ;      REGISTERS:      R2, R3.
3314      ;      CALLS:      DROPU, MOVMSG, FIRSTU, NEXTU, WSSR.
3315
3316      010774      EXCUTE::IF LCSR EQ #0 THEN
3317      010774      005737      003464
3318      011000      001004
3319      011002      LET TIME1 := #-1
3320      011002      012737      177777      003444
3321      011010      ELSE
3322      011010      000403
3323      011012      LET TIME1 := #304
3324      011012      012737      000304      003444
3325      011020      ENDF
3326      011020      REPEAT
3327      011020      ;WAIT -
3328      011020      IF LCSR EQ #0 THEN
3329      011020      005737      003464
3330      011024      001003
3331      011026      LET TIME1 := TIME1 - #1
3332      011026      005337      003444
3333      011032      ELSE
3334      011032      000403
3335      011034      LET @LCSR := #100
3336      011034      012777      000100      172422
3337      011042      ENDF
3338      011042      IF TIME1 EQ #0 THEN
3339      011042      ;IF TIMED OUT:
3340      011042      005737      003444
3341      011046      001011
3342      011050      JSR PC,MOVMSG
3343      011050      004737      012224
3344      011054      ERRDF 2,NSSRM,STAERM
3345      011054      104455
3346      011056      000002
3347      011060      004365
3348      011062      005640
3349      011064      JSR PC,DROPU
3350      011064      004737      017770
3351      011070      BR EXCRTN
3352      011072      ENDF
3353      011072      BREAK
3354      011072      104422
3355      011074      UNTIL #TS.SSR SETIN @TSSR(R5)
3356      011074      032775      000200      002466
3357      011102      001746
3358      011104      IF CMDWRD EQ #SCH THEN
3359      011104      023727      003426      140004
3360      011112      001022
3361      011114      LET R5SAVE := R5
3362      011114      010537      003460
3363      011120      JSR PC,FIRSTU
3364      011120      004737      017666

;      TST      LCSR
;      BNE      50071$
;      MOV      #-1,TIME1
;      BR      50072$
;      50071$:
;      MOV      #304,TIME1
;      50072$:
;      50073$:
;      TST      LCSR
;      BNE      50074$
;      DEC      TIME1
;      BR      50075$
;      50074$:
;      ;ARM THE INTERRUPTS
;      MOV      #100,@LCSR
;      50075$:
;      ;IF TIMED OUT:
;      TST      TIME1
;      BNE      50076$
;      ;MOVE CURRENT PACKET MSG.
;      ;REPORT TK25 NOT READY
;      TRAP     C$ERDF
;      .WORD    2
;      .WORD    NSSRM
;      .WORD    STAERM
;      ;DROP THE UNIT.
;      ;RETURN.
;      50076$:
;      TRAP     C$BRK
;      ;WAIT UNTIL DEVICE IS READY.
;      BIT      #TS.SSR,@TSSR(R5)
;      BEQ      50073$
;      ;IF WE ARE DOING A SET CHAR CMD THEN:
;      CMP      CMDWRD,#SCH
;      BNE      50077$
;      ;SAVE CURRENT DEVICE POINTER.
;      MOV      R5,R5SAVE
;      ;FIND FIRST UNIT.
    
```



```

3338 011124          WHILE DEVTBL(R5) NE #END DO
      011124
      011124 026527 002536 177777
      011132 001405
3339 011134 004737 012124      JSR PC,WSSR          ;WAIT FOR UNIT READY OR TIME OUT,
3340 011140 004737 017734      JSR PC,NEXTU        ;FIND NEXT UNIT.
3341 011144          ENDDO
      011144 000767
      011146
3342 011146          LET R5 := R5SAVE          ;RESTORE CURRENT DEVICE POINTER.
      011146 013705 003460      MOV R5SAVE,R5
3343 011152          LET SCHBK := MSGPKA(R5)      ;SET UP ADR OF MSG PKT IN SCH BLOCK.
      011152 016537 002506 002446  MOV MSGPKA(R5),SCHBK
3344 011160          ENDIF
      011160
3345 011160          LET R3 := MSGPKA(R5)      ;ADR OF THIS UNIT'S MSG PACKET.
      011160 016503 002506      MOV MSGPKA(R5),R3
3346 011164          LET R2 := #0              ;CLR COUNTER.
      011164 005002
3347 011166          WHILE R2 NE #MSGCNT DO      ;WHILE THERE ARE MORE LOCATIONS:
      011166
      011166 020227 000016      50077$:
      011172 001405          50101$:
3348 011174          LET (R3)+ := #-1          ;INIT THE MSG PACKET WITH ALL 1'S
      011174 012723 177777      BEQ 50103$
3349 011200          LET R2 := R2 + #2          ;UPDATE COUNTER.
      011200 062702 000002      MOV #-1,(R3)+
3350 011204          ENDDO
      011204 000770
      011206
3351 011206          TSTB DINT                ;ARE INTERRUPTS DISABLED.
3352 011212          BNE 1$                    ;BR IF YES.
3353 011214          IFB INTFLG(R5) GT #1 THEN  ;IF MORE THAN ONE INTERRUPT HAS OCCURED:
      011214 126527 003506 000001      CMPB INTFLG(R5),#1
      011222 003412          BLE 50104$
3354 011224          LET TSSREG := @TSSR(R5)      ;FREEZE THE CURRENT STATUS REG FOR PRINT
      011224 017537 002466 003462  MOV @TSSR(R5),TSSREG
3355 011232          ERRDF 15,TOOMM,STAERM      ;REPORT TOO MANY INTERRUPTS.
      011232 104455
      011234 000017
      011236 004554
      011240 005640
3356 011242          JSR PC,DROPU              ;DROP THE UNIT
3357 011246          BR EXCRTN                ;RETURN - UNIT HAS BEEN DROPPED.
3358 011250          ENDIF
      011250
3359 011250          LET INTFLG(R5) := #0      ;CLR INTERRUPT FLAG FOR THIS DEV.
      011250 005065 003506      CLR INTFLG(R5)
3360 011254          BIS #IE.C,CMDPKT          ;SET INT ENABLE BIT.
      052737 000200 002314      IFB ERRREC EQ #0 THEN ;IF NOT RETRYING
3361 011262          1$:
      011262 105737 003505
      011266 001005
3362 011270          LET RECCNT(R5) := RECCNT(R5) + #1
      011270 005265 003330
3363 011274          LET @DATAWT := RECCNT(R5) ;THEN UPDATE REC COUNT TO WRITE IT ON TAPE
      011274 016577 003330 172112  INC RECCNT(R5)
3364 011302          ENDIF
      MOV RECCNT(R5),@DATAWT

```

3365	011302				IF L\$TEST EQ #3 AND CMDWRD EQ #WRT THEN	50105\$:	
	011302	023727	002114	000003		CMP	L\$TEST,#3
	011310	001047				BNE	50106\$
	011312	023727	003426	104005		CMP	CMDWRD,#WRT
	011320	001043				BNE	50106\$
3366	011322				LET LOOPCT :B= LOOPCT + #1	INCB	LOOPCT
	011322	105237	003537				
3367	011326				IFB LOOPCT GT 3 THEN	CMPB	LOOPCT,3
	011326	123737	003537	000003		BLE	50107\$
	011334	003435					
3368	011336				LET LOOPCT :B= #0	CLRB	LOOPCT
	011336	105037	003537				
3369	011342				LET TIME1 := #4	MOV	#4,TIME1
	011342	012737	000004	003444			
3370	011350				REPEAT		
	011350						
3371	011350				IF LCSR NE #0 THEN	50110\$:	
	011350	005737	003464			TST	LCSR
	011354	001404				BEQ	50111\$
3372	011356				LET @LCSR := #100	MOV	#100,@LCSR
	011356	012777	000100	172100			
3373	011364				ELSE	BR	50112\$
	011364	000416				50111\$:	
	011366				DELAY 21.		
3374	011366					MOV	#21.,(PC)+
	011366	012727	000025			.WORD	0
	011372	000000				MOV	L\$DLY,(PC)+
	011374	013727	002116			.WORD	0
	011400	000000				DEC	-6(PC)
	011402	005367	177772			BNE	.-4
	011406	001375				DEC	-22(PC)
	011410	005367	177756			BNE	.-20
	011414	001367					
3375	011416				LET TIME1 := #0	CLR	TIME1
	011416	005037	003444				
3376	011422				ENDIF	50112\$:	
	011422					TST	TIME1
3377	011422				UNTIL TIME1 EQ #0	BNE	50110\$
	011422	005737	003444				
	011426	001350					
3378	011430				ENDIF	50107\$:	
	011430						
3379	011430				ENDIF	50106\$:	
	011430						
3380	011430	012775	002314	002456	MOV #CMDPKT,@TSDB(R5)		
3381							
3382	011436				IF #TS.SSR SETIN @TSSR(R5) THEN		
	011436	032775	000200	002466			
	011444	001410					
3383	011446	004737		24	JSR PC,MOVMSG		
3384	011452				ERRDF 3,TOERM,STAERM		
	011452	104455					
	011454	000003					
	011456	004303					
	011460	005640					
3385	011462	004737	017770		JSR PC,DROPU		

```

;LOAD TSDB WITH CMDPKT ADDRESS
;THIS INITIATES COMMAND EXECUTION.
;IF READY DID NOT DROP THEN:
    BIT #TS.SSR,@TSSR(R5)
    BEQ 50113$
;MOVE CURRENT MESSAGE PACKET TO COMMON.
;REPORT NO TK25 RESPONSE.
    TRAP C$ERDF
    .WORD 3
    .WORD TOERM
    .WORD STAERM
;DROP THE UNIT
    
```

```

3386 011466          ENDIF
      011466
3387 011466 000207  EXCRTN: RTS      PC          ;RETURN.          50113$:
3388
3389 ;
3390 ; THIS SUBROUTINE WAITS FOR THE TK25 INERRUPT OR DONE BIT TO SET AND ALLOWS THE
3391 ; OPERATOR TO TRANSFER CONROL TO THE SUPERVISOR.
3392 ; UPON APPEARANCE OF THE INTERRUPT OR DONE, CHECK TSSR FOR STATUS ERRORS,
3393 ; LOG BYTES AND ERRORS AND PERFORM ERROR RECOVERY IF NESSASARY.
3394 ; INPUTS:
3395 ; OUTPUTS:
3396 ; REGISTERS:      R2, R3.
3397 ; CALLS:          DROPU, MOVMSG, RECUD, CHKERR, LOG, CLRERR.
3398 011470          GOWAIT::IF DEVTBL(R5) EQ #NINUSE THEN
      011470 026527 002536 177774          CMP      DEVTBL(R5),#NINUSE
      011476 001002          BNE      50114$
3399 011500 000137 012106          JMP      1$
3400 011504          ENDIF
      011504          50114$:
3401 011504          IF LCSR EQ #0 THEN
      011504 005737 003464          TST      LCSR
      011510 001004          BNE      50115$
3402 011512          LET TIME1 := #-1          ;INIT TIME OUT COUNTER.
      011512 012737 177777 003444          MOV      #-1,TIME1
3403 011520          ELSE
      011520 000403          BR      50116$
3404 011522          LET TIME1 := #37777          ;LINE CLOCK STALL VALUE
      011522 012737 037777 003444          MOV      #37777,TIME1
3405 011530          ENDIF
3406 011530          REPEAT          ;REPEAT UNTIL INTERRUPT OCCURES:
      011530          50116$:
3407 011530          BREAK          ;HONOR SUPERVISOR BREAKS
      011530 104422          TRAP      C$BRK
3408 011532          IF PCMDWD EQ #RWD AND CMDWRD EQ #WRT THEN
      011532 023727 003432 102010          CMP      PCMDWD,#RWD
      011540 001020          BNE      50120$
      011542 023727 003426 104005          CMP      CMDWRD,#WRT
      011550 001014          BNE      50120$
3409          ;POSSIBLE FIRST WRITE ON TAPE
3410 011552          DELAY 40.
      011552 012727 000050          MOV      #40.,(PC)+
      011556 000000          .WORD      0
      011560 013727 002116          MOV      L$DLY,(PC)+
      011564 000000          .WORD      0
      011566 005367 177772          DEC      -6(PC)
      011572 001375          BNE      -.4
      011574 005367 177756          DEC      -22(PC)
      011600 001367          BNE      -.20
3411 011602          ENDIF          ;SO DELAY TO CALIBRATE TAPE
      011602          50120$:
3412 011602          IF CMDWRD EQ #RWD THEN          ;IF COMMAND WAS REWIND THEN:
      011602 023727 003426 102010          CMP      CMDWRD,#RWD
      011610 001014          BNE      50121$
3413 011612          DELAY 10.          ;WAIT EXTRA 10 MSECS EACH LOOP.
      011612 012727 000012          MOV      #10.,(PC)+

```



```

011770 005737 003444
011774 001255
011776
3430 011776          IF TIME1 EQ #0 THEN          ;IF TIME OUT HAS OCCURRED:
011776 005737 003444          TST      TIME1
012002 001022          BNE      50117$
012004          50130$:
3431 012004          LET @DATAWT := RECCNT(R5) - #1 ;RE-ADJUST REC COUNT DOWN
012004 016577 003330 171402          MOV      RECCNT(R5),@DATAWT
012012 005377 171376          DEC      @DATAWT
3432 012016 004737 012224          JSR      PC,MOVMSG          ;MOVE CURRENT MSG PACKET TO COMMON AREA.
3433 012022          ERRDF 4,NOINTM,STAERM ;REPORT NO INTERRUPT.
012022 104455          TRAP    C$ERDF
012024 000004          .WORD  4
012026 004515          .WORD  NOINTM
012030 005640          .WORD  STAERM
3434 012032 004737 017770          JSR      PC,DROPU          ;DROP THE UNIT.
3435 012036          LET R3 := #ENDERF
012036 012703 003506          MOV      #ENDERF,R3
3436 012042 004737 012110          JSR      PC,CLRERR          ;CLEAR ALL ERROR FLAGS
3437 012046          ELSE
012046 000417          BR      50132$
012050          50131$:
3438 012050 004737 012224          JSR      PC,MOVMSG          ;MOVE CURRENT MSG. PACKET TO COMMON AREA.
3439 012054 004737 012310          JSR      PC,RECU          ;UPDATE THE RECORD COUNT.
3440 012060 004737 012456          JSR      PC,CHKERR          ;CHECK FOR STATUS ERRORS.
3441 012064          IFB WRTYFG EQ #0 THEN
012064 105737 003477          ;
012070 001006          TSTB   WRTYFG
3442 012072 004737 016024          BNE      50133$
3443 012076          JSR      PC,LOG          ;LOG BYTES AND ERRORS.
012076 012703 003506          LET R3 := #ENDERF
3444 012102 004737 012110          JSR      PC,CLRERR          ;CLEAR ALL ERROR FLAGS
3445 012106          ENDIF
012106          MOV      #ENDERF,R3
3446 012106          ENDIF          50133$:
012106          50132$:
3447 012106 000207          1$: RTS      PC          ;RETURN IF DONE.

```

```

3449 ; SUBROUTINE TO CLEAR FLAGS.
3450 ; INPUTS: R3 = LWA TO BE CLEARED + 2.
3451 ; OUTPUTS:
3452 ; REGISTERS: R2
3453 ; CALLS:
3454
3455 012110 CLRERR:: LET R2 := #BGNFLG
012110 012702 003474
3456 012114 REPEAT MOV #BGNFLG,R2
012114 50134$:
3457 012114 LET (R2)+ := #0
012114 005022 CLR (R2)+
3458 012116 UNTIL R2 EQ R3 CMP R2,R3
012116 020203 BNE 50134$
012120 001375
3459 012122 000207 RTS PC
3460
3461
3462
3463 ; SUBROUTINE TO WAIT UNTIL CURRENT UNIT IS READY OR UNTIL TIME OUT.
3464 ; INPUTS:
3465 ; OUTPUTS:
3466 ; REGISTERS:
3467 ; CALLS:
3468
3469 012124 WSSR:: LET TIME1 := #-1 ;INIT TIMEOUT COUNTER.
012124 012737 177777 003444 REPEAT MOV #-1,TIME1
3470 012132 ;REPEAT UNTIL DEV READY OR TIMEOUT:
012132 50135$:
3471 012132 BREAK ;BREAK TO THE SUPERVISOR.
012132 104422 TRAP C$BRK
3472 012134 IF LCSR EQ #0 THEN
012134 005737 003464 TST LCSR
012140 001003 BNE 50136$
3473 012142 LET TIME1 := TIME1 - #1 ;UPDATE TIMEOUT COUNTER.
012142 005337 003444 DEC TIME1
3474 012146 ELSE
012146 000403 BR 50137$
012150 50136$:
3475 012150 LET @LCSR := #100 ;ARM CLOCK INTERRUPTS
012150 012777 000100 171306 MOV #100,@LCSR
3476 012156 ENDIF
012156 50137$:
3477 012156 UNTIL #TS.SSR SETIN @TSSR(R5) OR TIME1 EQ #0
012156 032775 000200 002466 BIT #TS.SSR,@TSSR(R5)
012164 001003 BNE 50140$
012166 005737 003444 TST TIME1
012172 001357 BNE 50135$
012174 50140$:
3478 ;REPEAT UNTIL DEV READY OR TIMEOUT.
3479 012174 IF TIME1 EQ #0 THEN ;IF SYSTEM TIMED OUT THEN
012174 005737 003444 TST TIME1
012200 001010 BNE 50141$
3480 012202 004737 012224 JSR PC, MOVMSG ;MOVE CURRENT PACKET MSG
3481 012206 ERRDF 20,NSSRM,STAERM ;SSR DIDN'T SET IN TIME
012206 104455 TRAP C$ERDF
012210 000024 .WORD 20

```

```

012212 004365
012214 005640
3482 012216 004737 017770 JSR PC, DROPU ;THROW THE UNIT ON THE FLOOR
3483 012222 004737 017770 ENDIF
012222 000207 RTS PC ;RETURN.
3484 012222 000207
3485
3486
3487 ; SUBROUTINE TO MOVE THE CURRENT MESSAGE PACKET TO THE COMMON AREA AND
3488 ; TO UPDATE THE CURRENT TERMINATION CLASS CODE.
3489 ; INPUTS:
3490 ; OUTPUTS:
3491 ; REGISTERS: R2, R3.
3492 ; CALLS:
3493
3494 012224 017537 002466 003462 MOVMSG:: LET TSSREG := @TSSR(R5) ;FREEZE THE STATUS REG CONTENTS
012224 017537 002466 003462 MOV @TSSR(R5),TSSREG
3495 012232 013702 003462 LET R2 := TSSREG CLR.BY #TSC.TCC ;EXTRACT THE TERMINATION CLASS CODE,
012232 013702 003462 MOV TSSREG,R2
012236 042702 177761 BIC #TSC.TCC,R2
3496 012242 010237 003456 LET CTCC := R2 SHIFT -1 ;AND SAVE IT
012242 010237 003456 MOV R2,CTCC
012246 006237 003456 ASR CTCC
3497 012252 016503 002506 LET R3 := MSGPKA(R5) ;ADR OF THIS DEVICE'S MSG.
012252 016503 002506 MOV MSGPKA(R5),R3
3498 012256 005002 LET R2 := #0 ;CLR COUNTER.
012256 005002
3499 012260 020227 000016 WHILE R2 NE #MSGCNT DO ;WHILE THERE ARE MORE LOCATIONS:
012260 020227 000016 50142$:
012264 001405 CMP R2,#MSGCNT
3500 012266 012362 002340 LET MSGPKT(R2) := (R3)+ ;MOVE MSG TO COMMON AREA.
012266 012362 002340 BEQ 50143$
3501 012272 062702 000002 LET R2 := R2 + #2 ;UPDATE COUNTER.
012272 062702 000002 MOV (R3)+,MSGPKT(R2)
3502 012276 000770 ENDDO ADD #2,R2
012276 000770 BR 50142$
012300 50143$:
3503 012300 013765 002346 003516 LET EOTFLG(R5) := MSGPKT+MS.XS0 ;MOVE XSTATO TO EOT FLAG.
012300 013765 002346 003516 MOV MSGPKT+MS.XS0,EOTFLG(R5)
3504 012306 000207 RTS PC

```

```

3506 ; SUBROUTINE TO ADJUST THE RECORD COUNT.
3507 ; INPUTS:
3508 ; OUTPUTS:
3509 ; REGISTERS:
3510 ; CALLS:
3511 ;
3512 012310 RECUD:: IFB RECLOG EQ #0 THEN ;IF RECORD HAS NOT BEEN LOGGED:
      012310 105737 003501 TSTB RECLOG
      012314 001057 BNE 50144$
3513 012316 LET RECCNT(R5) := RECCNT(R5) - #1
      012316 005365 003330 DEC RECCNT(R5)
3514 012322 IF #BITO NOTSETIN CTCC AND #X2.OPM SETIN MSGPKT+MS.XS2 THEN ;IF TAPE MOVED THEN:
      012322 032737 000001 003456 BIT #BITO,CTCC
      012330 001046 BNE 50145$
      012332 032737 100000 002352 BIT #X2.OPM,MSGPKT+MS.XS2
      012340 001442 BEQ 50145$
3515 012342 LET RECLOG :B= RECLOG + #1 ;SET RECORD LOGGED,
      012342 105237 003501 INCB RECLOG
3516 012346 IF CMDWRD EQ #RWD THEN ;IF THIS IS A REWIND CMD:
      012346 023727 003426 102010 CMP CMDWRD,#RWD
      012354 001003 BNE 50146$
3517 012356 LET RECCNT(R5) := #0 ;CLEAR RECORD COUNT,
      012356 005065 003330 CLR RECCNT(R5)
3518 012362 ELSE
      012362 000431 BR 50147$
3519 012364 IF #BRF.C SETIN CMDWRD THEN ;IF BRF USED, UPDATE RECORD COUNT.
      012364 032737 004000 003426 BIT #BRF.C,CMDWRD
      012372 001425 BEQ 50150$
3520 012374 IF #MOD.CO NOTSETIN CMDWRD THEN ;IF A FORWARD CMD:
      012374 032737 000400 003426 BIT #MOD.CO,CMDWRD
      012402 001007 BNE 50151$
3521 012404 IF #MOD.CO NOTSETIN PCMDWD THEN ;IF PREV CMD WAS A FWD ALSO:
      012404 032737 000400 003432 BIT #MOD.CO,PCMDWD
      012412 001002 BNE 50152$
3522 012414 LET RECCNT(R5) := RECCNT(R5) + #1 ;INCREMENT RECORD COUNT.
      012414 005265 003330 INC RECCNT(R5)
3523 012420 ENDIF
3524 012420 ELSE ;IF REVERSE CMD:
      012420 000412 BR 50153$
3525 012422 IF #MOD.CO SETIN PCMDWD THEN ;IF PREVIOUS CMD WAS A REV ALSO:
      012422 032737 000400 003432 BIT #MOD.CO,PCMDWD
      012430 001406 BEQ 50154$
3526 012432 IF #X0.BOT NOTSETIN EOTFLG(R5) THEN ;WHEN NOT AT BOT THEN
      012432 032765 000002 003516 BIT #X0.BOT,EOTFLG(R5)
      012440 001002 BNE 50155$
3527 012442 LET RECCNT(R5) := RECCNT(R5) - #1 ;DECREMENT RECORD COUNT.
      012442 005365 003330 DEC RECCNT(R5)
3528 012446 ENDIF
3529 012446 ENDIF
3530 012446 ENDIF
3531 012446 ENDIF

```


3532	012446				ENDIF	50150\$:
	012446					50147\$:
3533	012446				ENDIF	50145\$:
	012446				LET @DATAWT := RECCNT(R5)	MOV RECCNT(R5),@DATAWT
3534	012446	016577	003330	170740	ENDIF	50144\$:
3535	012454					
	012454					
3536	012454	000207			RTS PC ;RETURN.	

```

3538 ; THIS IS THE ERROR CHECK SUBROUTINE. AFTER INTERRUPT THIS
3539 ; SUBROUTINE IS CALLED TO CHECK THE TK25 STATUS.
3540 ; IF SPECIAL COND IS SET THEN THE TCC HANDLING SUBROUTINE IS ENTERED.
3541 ; IF THE RFC IS NON ZERO FOR A COMMAND REQUIRING A BPCR,
3542 ; THEN AN ERROR RFC IS REPORTED,
3543 ; INPUTS:
3544 ; OUTPUTS:
3545 ; REGISTERS: R2, R4.
3546 ; CALLS: TCC0-TCC7.
3547
3548 012456 ; CHKERR::IF DEVTBL(R5) NE #NINUSE THEN
012456 026527 002536 177774 ;
012464 001570 ;
3549 012466 ; IF #X1.EWN SETIN MSGPKT+MS.XS1 THEN ;IF EARLY WARNING IS SET AND
012466 032737 000010 002350 ;
012474 001461 ;
3550 012476 ; IFB EWRPNT NE #0 THEN ;IF END OF TRACK STATUS PRINTS ALLOWED
012476 105737 002216 ;
012502 001433 ;
3551 012504 ; LET R2 := MSGPKT+MS.XS1 CLR.BY #177417
012504 013702 002350 ;
012510 042702 177417 ;
3552 012514 ; LET R2 := R2 SHIFT -4
012514 006202 ;
012516 006202 ;
012520 006202 ;
012522 006202 ;
3553 012524 ; PRINTF #EWSMSG,R2 ;PRINT END OF TRACK MESSAGE
012524 010246 ;
012526 012746 013050 ;
012532 012746 000002 ;
012536 010600 ;
012540 104417 ;
012542 062706 000006 ;
3554 012546 ; PRINTF #TRKMSG,@DATAWT ;*** TRACE
012546 017746 170642 ;
012552 012746 013133 ;
012556 012746 000002 ;
012562 010600 ;
012564 104417 ;
012566 062706 000006 ;
3555 012572 ; ENDIF
012572 ;
3556 012572 ; IF CTCC EQ #4 THEN ;IF TCC4 AND WE DID A WRITE
012572 023727 003456 000004 ;
012600 001017 ;
3557 012602 ; IF CMDWRD EQ #WRT OR CMDWRD EQ #WTM THEN
012602 023727 003426 104005 ;
012610 001404 ;
012612 023727 003426 100011 ;
012620 001007 ;
012622 ;
3558 012622 ; IFB IREC EQ #0 THEN
012622 105737 002210 ;
012626 001004 ;
3559 012630 ; JSR PC,EWRTRY ;DO EW RETRY
012634 000137 013046 ; JMP 1$ ;ALAS, A "GOTO" STATEMENT, EH?

```

```

3561 012640          ENDIF
      012640
3562 012640          ENDIF
      012640
3563 012640          ENDIF
      012640
3564 012640          ENDIF
      012640
3565 012640          IF #TS.SC SETIN TSSREG THEN      ;IF SPECIAL COND STATUS IS SET THEN:
      012640 032737 100000 003462                    ;IF TCC IS NOT 2 THEN:
      012646 001441                                ;IF NOT IN ERROR RECOVERY:
3566 012650          IF CTCC NE #2 THEN
      012650 023727 003456 000002                    ;INC SC COUNTER.
      012656 001405
3567 012660          IFB ERRREC EQ #0 THEN
      012660 105737 003505
      012664 001002
3568 012666          INC          SCCNT(R5)
3569 012672          ENDIF
      012672
3570 012672          ENDIF
      012672
3571 012672          IF #TS.NXM SETIN TSSREG OR #TS.UPE SETIN TSSREG THEN ;WHEN NON-EXISTANT MEMO
      012672 032737 004000 003462                    ;AND TAPE NOT MOVED
      012700 001004                                ;SET TCC5 INDEX
      012702 032737 040000 003462                    ;TAPE MOVED, SET TCC4 INDEX
      012710 001412                                ;CURRENT TCC X 2.
      012712
3572 012712          IF #X2.OPM NOTSETIN MSGPKT*MS.XS2 THEN
      012712 032737 100000 002352                    ;GO TO THE TCC HANDLING SUBROUTINE.
      012720 001003
3573 012722          LET R2 := #5
3574 012726          ELSE
      012726 000402
      012730
3575 012730          LET R2 := #4
3576 012734          ENDIF
      012734
3577 012734          ELSE
      012734 000402
      012736
3578 012736          LET R2 := CTCC
3579 012742          ENDIF
      012742
3580 012742          LET R2 := R2 SHIFT 1
3581 012744          JSR PC,@TCCRA(R2)
3582 012750          ELSE
      012750 000430
      012752
3583 012752          IF #BRF.C SETIN CMDWRD THEN      ;IF BRF IS USED IN THIS CMD THEN:
      012752 032737 004000 003426                    ;IF THERE IS AN RFC THEN:
      012760 001424
3584 012762

```

```

012762 005737 002344
012766 001421
3585 012770 IFB RANDOM EQ #0 ORB VFYFLG NE #0 THEN
012770 105737 003533
012774 001403
012776 105737 003534
013002 001413
013004
3586
3587 013004 IFB IRE EQ #0 THEN ;IF NOT IN RANDOM OR IF CMD IS WTV:
013004 105737 003540 ;IF RFC ERROR REPORTS ARE ALLOWED:
013010 001010 TSTB IRE
3588 013012 LET HRDCNT(R5) := HRDCNT(R5) + #1 ;UPDATE HARD ERROR COUNT
013012 005265 003310 INC HRDCNT(R5)
3589 013016 ERRHRD 13,RFCERM,STAERM ;REPORT RFC ERROR
013016 104456 TRAP C$ERHRD
013020 000015 .WORD 13
013022 004350 .WORD RFCERM
013024 005640 .WORD STAERM
3590 013026 004737 014436 JSR PC,RAMDUM ;GO DO RAM DUMP
3591 013032 ENDF
3592 013032 ENDF 50202$:
3593 013032 ENDF 50201$:
3594 013032 ENDF 50177$:
3595 013032 ENDF 50176$:
3596 013032 IFB RWERR NE #0 THEN ;IF A READ/WRITE ERROR HAS OCCURRED THEN:
013032 105737 003503 TSTB RWERR
013036 001403 BEQ 50203$
3597 013040 LET CMDPKT := C$CMD V ;RESTORE CMD PACKET AFTER ERROR RECOV.
013040 013737 003430 002314 MOV C$MSAV,CMDPKT
3598 013046 ENDF
3599 013046 ENDF 50203$:
3600 013046 000207 1$: RTS PC ;RETURN. 50156$:
3601
3602 013050 045 116 045 EWMSG: .ASCIZ /%N%AEARLY WARNING DURING WRITE AT END OF TRACK #D2/
013053 101 105 101
013056 122 114 131
013061 040 127 101
013064 122 116 111
013067 116 107 040
013072 104 125 122
013075 111 116 107
013100 040 127 122
013103 111 124 105
013106 040 101 124
013111 040 105 116
013114 104 040 117
013117 106 040 124
013122 122 101 103
013125 113 040 045

```

	013130	104	062	000	
3603	013133	045	101	040	TRKMSG: .ASCIZ /#A RECORD NO #D5/
	013136	122	105	103	
	013141	117	122	104	
	013144	040	116	117	
	013147	040	045	104	
	013152	065	000		

3604 .EVEN

3605

3606

3607

3608

: ADDRESSES OF TCC HANDLING ROUTINES FOR TERMINATION CLASS CODES 0 - 7.

3609	013154	013306	TCCRA: TCC0
3610	013156	013330	TCC1
3611	013160	013346	TCC2
3612	013162	013514	TCC3
3613	013164	013536	TCC4
3614	013166	014256	TCC5
3615	013170	014360	TCC6
3616	013172	014414	TCC7

```

3618      ;      SUBROUTINE TO HANDLE EARLY WARNING WRITE ERRORS.
3619      ;
3620      ;      THIS ROUTINE WILL SIMPLY TAKE CARE OF EARLY WARNING WRITE ERRORS
3621      ;      BY REISSUING THE COMMAND WHICH FAILED WITH THE RETRY MODIFIER SET.
3622      ;      NOTE THAT NO ERROR FLAGS, COUNTS, ETC ARE CHANGED.
3623      ;      (EW ERRORS WILL NOT SHOW UP IN ERROR TALLIES).
3624      ;
3625      013174      EWRTRY::
3626      013174      LET -(SP) := CMDPKT          ;SAVE THE COMMAND PACKET
3627      013174      013746      002314      MOV      CMDPKT,-(SP)
3628      013200      052737      001000      002314      LET CMDPKT := CMDPKT SET.BY #MOD.C1      ;SET THE RETRY MODIFIER IN COMMAND
3629      013206      112737      000001      003505      BIS      #MOD.C1,CMDPKT
3630      013214      004737      010774      JSR      PC,EXCUTE          ;DO THE COMMAND
3631      013220      105037      003505      LET ERRREC :B= #1          ;SHOW ERROR RECOVERY IN PROCESS
3632      013224      012637      002314      MOV      #1,ERRREC
3633      013230      012602      JSR      PC,WSSR          ;NO MORE ERROR RECOVERY
3634      013232      004737      012124      LET CMDPKT := (SP)+      ;RESTORE THE ORIGINAL COMMAND STATUS
3635      013236      012746      000012      LET R2 := (SP)+      ;MODIFY THE RETURN ADDRESS
3636      013242      REPEAT          ;WAIT FOR THE COMMAND TO FINISH
3637      013242      012727      000372      LET -(SP) := #10.      ;OVERALL LOOP CONTROL
3638      013246      000000      DELAY 250.          ;LOOP
3639      013250      013727      002116      MOV      #10,-(SP)
3640      013254      000000      012727      000372      50204$:
3641      013256      005367      177772      MOV      #250.,(PC)+
3642      013262      001375      012727      000372      .WORD      0
3643      013264      005367      177756      MOV      L$DLY,(PC)+
3644      013270      001367      012727      000372      .WORD      0
3645      013272      005316      012727      000372      DEC      -6(PC)
3646      013274      005716      012727      000372      BNE      -4
3647      013276      001361      012727      000372      DEC      -22(PC)
3648      013300      062706      000002      BNE      -20
3649      013304      000207      LET (SP) := (SP) - #1          ;ONE LESS ITERATION
3650      013304      UNTIL (SP) EQ #0          ;UNTIL = 0
3651      013304      LET SP := SP + #2          ;CORRECT THE STACK
3652      013304      RTS      PC          ;RETURN TO CALLER
3653      013304      ADD      #2,SP

```

```

3642 ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 0, UNDEFINED SPECIAL
3643 ; CONDITION ERROR.
3644 ; INPUTS:
3645 ; OUTPUTS:
3646 ; REGISTERS:
3647 ; CALLS:
3648
3649 013306 TCC0:: LET HRDCNT(R5) := HRDCNT(R5) + #1 ;UPDATE HARD ERROR COUNT.
013306 005265 003310 INC HRDCNT(R5)
3650 013312 ERRHRD 5,SCERM,STAERM ;REPORT SPECIAL CONDITION ERROR.
013312 104456 TRAP C$ERHRD
013314 000005 .WORD 5
013316 004324 .WORD SCERM
013320 005640 .WORD STAERM
3651 013322 004737 014436 JSR PC,RAMDUM ;GO DO RAM DUMP
3652 013326 000207 RTS PC ;RETURN.
3653
3654
3655
3656
3657
3658 ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 1, ATTENTION CONDITION.
3659 ; THIS TCC INDICATES THAT THE DRIVE HAS UNDERGONE A STATUS CHANGE
3660 ; SUCH AS GOING OFFLINE OR COMING ONLINE.
3661 ; INPUTS:
3662 ; OUTPUTS:
3663 ; REGISTERS: R2,R4
3664 ; CALLS: DROPU
3665
3666 013330 TCC1:: ERRDF 6,ATTNM,STAERM ;REPORT ATTENTION-UNIT OFF LINE.
013330 104455 TRAP C$ERDF
013332 000006 .WORD 6
013334 004431 .WORD ATTNM
013336 005640 .WORD STAERM
3667 013340 004737 014436 JSR PC,RAMDUM ;GO DO RAM DUMP
3668 013344 000207 RTS PC ;RETURN.

```

```

3670 ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 2, TAPE STATUS ALERT.
3671 ; A STATUS CONDITION HAS BEEN ENCOUNTERED THAT MAY HAVE SIGNIFICANCE
3672 ; TO THE PROGRAM. BITS OF INTEREST INCLUDE TMK, RLS, LET, RLL, BOT, EOT.
3673 ; INPUTS:
3674 ; OUTPUTS:
3675 ; REGISTERS:
3676 ; CALLS:
3677 ;
3678 013346 TCC2:: IF #X0.BOT SETIN MSGPKT+MS.XSO ANDB EXPBOT NE #0 THEN
013346 032737 000002 002346 BIT #X0.BOT,MSGPKT+MS.XSO
013354 001404 BEQ 50205$
013356 105737 003532 TSTB EXPBOT
013362 001401 BEQ 50205$
3679 ;IF AT BOT AND BOT IS EXPECTED:
3680 013364 000452 BR TC2RTN ;RETURN-TCC2 CAUSED BY EXPECTED BOT.
3681 013366 ENDF
013366 50205$:
3682 013366 IF L$TEST EQ #3 THEN CMP L$TEST,#3
013366 023727 002114 000003 BNE 50206$
013374 001011
3683 013376 IF PCMDWD EQ #WTM AND CMDWRD EQ #SRR THEN CMP PCMDWD,#WTM
013376 023727 003432 100011 BNE 50207$
013404 001005 CMP CMDWRD,#SRR
013406 023727 003426 104410 BNE 50207$
013414 001001
3684 013416 000435 BR TC2RTN
3685 013420 ENDF
013420 50207$:
3686 013420 ENDF
013420 50206$:
3687 013420 IF #X0.RLS!X0.RLL!X0.TMK!X0.LET!X0.BOT SETIN MSGPKT+MS.XSO THEN
013420 032737 170002 002346 BIT #X0.RLS!X0.RLL!X0.TMK!X0.LET
013426 001431 BEQ 50210$
3688 ;IF TCC2 CAUSED BY ANYTHING BUT EOT:
3689 013430 IFB RANDOM EQ #0 ORB VFYFLG NE #0 THEN
013430 105737 003533 TSTB RANDOM
013434 001403 BEQ 50211$
013436 105737 003534 TSTB VFYFLG
013442 001423 BEQ 50212$
013444 50211$:
3690 ;IF NOT IN RANDOM OR IF CMD IS WTV:
3691 013444 IFB IRE EQ #0 THEN ;IF RFC ERROR REPORTS ARE ALLOWED:
013444 105737 003540 TSTB IRE
013450 001020 BNE 50213$
3692 013452 IFB ERRREC NE #0 THEN ;IF WE ARE IN ERROR RECOVERY THEN:
013452 105737 003505 TSTB ERRREC
013456 001403 BEQ 50214$
3693 013460 LET UNREC :B= UNREC + #1 ;SET UNRECOVERABLE FLAG FOR LOG.
013460 105237 003504 INCB UNREC
3694 013464 ELSE ;ELSE - IF NOT IN ERROR RECOVERY:
013464 000402 BR 50215$
013466 50214$:
3695 013466 LET SCCNT(R5) := SCCNT(R5) + #1 ;INCREMENT THE SPEC COND COUNTER.
013466 005265 003270 INC SCCNT(R5)
3696 013472 ENDF
013472
3697 013472 LET HRDCNT(R5) := HRDCNT(R5) + #1 ;UPDATE HARD ERROR COUNT.
50215$:

```



```

3698 013472 005265 003310
      013476 104456          ERRHRD 7,TSAM,STAERM          ;REPORT TAPE STATUS ALERT.
      013500 000007          ;                               TRAP      C$ERHRD
      013502 004532          ;                               .WORD      7
      013504 005640          ;                               .WORD      TSAM
3699 013506 004737 014436          ;                               .WORD      STAERM
3700 013512          JSR      PC,RAMDUM          ;GO DO RAM DUMP
      013512          ENDIF
3701 013512          ENDIF          50213$:
      013512          ENDIF          50212$:
3702 013512          ENDIF          50210$:
3703 013512 000207          TC2RTN: RTS PC          ;RETURN.
3704
3705
3706
3707
3708
3709
3710          ; THE SPECIFIED FUNCTION WAS NOT INITIATED. BITS OF INTEREST ARE
3711          ; RMR, OFL, VCK, BOT, ILC, WLE, ILA, AND NBA.
3712          ; INPUTS:
3713          ; OUTPUTS:
3714          ; REGISTERS:      R2,R4
3715          ; CALLS:          DROPU
3716
3717 013514          TCC3:: ERRDF 8,FUNRM,STAERM          ;REPORT FUNCTION REJECT.
      013514 104455          ;                               TRAP      C$ERDF
      013516 000010          ;                               .WORD      8
      013520 004447          ;                               .WORD      FUNRM
      013522 005640          ;                               .WORD      STAERM
3718 013524 004737 014436          JSR      PC,RAMDUM          ;GO DO RAM DUMP
3719 013530 004737 017770          JSR      PC,DROPU          ;DROP THE UNIT.
3720 013534 000207          RTS PC          ;RETURN.

```

```

3722 ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 4, RECOVERABLE ERROR.
3723 ; TAPE POSITION IS ONE RECORD BEYOND WHAT ITS POSITION WAS WHEN
3724 ; THE FUNCTION WAS INITIATED. RECOVERY PROCEDURE IS TO LOG THE
3725 ; ERROR AND ISSUE THE APPROPRIATE RETRY COMMAND.
3726 ; 2 WRITE-ERROR-RECOVERY ALGORITHMS CAN BE SELECTED:
3727 ; THE FIRST ONE, VIA BADTSW SWITCH, DOES DETECT BAD SPOTS ON TAPE.
3728 ; IT CALLS A WRITE RETRY SUBR UNTIL THE RECORD IS RECOVERED
3729 ; OR 20 BAD SPOTS HAVE BEEN LOGGED. AFTER LOGGING 256 BAD
3730 ; SPOTS, A BAD TAPE OVERFLOW MSG IS PRINTED AND THE
3731 ; UNIT DROPPED.
3732 ; THE SECOND ALGORITHM ISSUES THE TK25 WRITE RETRY COMMAND
3733 ; UP TO 16 TIMES BEFORE DROPPING THE UNIT OR PROCEEDING
3734 ; WITH THE NEXT RECORD ON RECOVERY.
3735 ; INPUTS:
3736 ; OUTPUTS:
3737 ; REGISTERS: R2,R4.
3738 ; CALLS: RTLE, EXECUTE, GOWAIT, DROPU, WRTY
3739
3740 013536 TCC4:: IF DEVTBL(R5) EQ #NINUSE THEN
013536 026527 002536 177774 CMP DEVTBL(R5),#NINUSE
013544 001002 BNE 50216$
3741 013546 000137 014254 JMP 3$
3742 013552 ENDIF
013552
3743 013552 IF CMDLG EQ #2 ANDB BADTSW NE #0 THEN 50216$:
013552 023727 003434 000002 CMP CMDLG,#2
013560 001152 BNE 50217$
013562 105737 002211 TSTB BADTSW
013566 001547 BEQ 50217$
3744 013570 IFB ERRREC EQ #0 ANDB ERCVER NE #0 THEN
013570 105737 003505 TSTB ERRREC
013574 001011 BNE 50220$
013576 105737 002207 TSTB ERCVER
013602 001406 BEQ 50220$
3745 013604 ERRSOFT 9,RERM,STAERM ;
013604 104457 TRAP C$ERSOFT
013606 000011 .WORD 9
013610 004644 .WORD RERM
013612 005640 .WORD STAERM
3746 013614 JSR PC,RAMDUM ;GO DO RAM DUMP
3747 013620 ENDIF
013620
3748 013620 IFB IREC EQ #0 THEN ; 50220$:
013620 105737 002210 TSTB IREC
013624 001125 BNE 50221$
3749 013626 LET ERRREC :B= ERRREC + #1 ;RETRY FLAG FOR EXECUTE SUBR: DON'T UPDATE REC CNT
013626 105237 003505 INCB ERRREC
3750 013632 LET WRTYER :B= WRTYER + #1 ;REWRITE ERROR FLAG FOR WRTY SUBR
013632 105237 003500 INCB WRTYER
3751 013636 IFB WRTYFG EQ #0 THEN ;FIRST RETRY ON THIS RECORD: SUBSEQUENT
013636 105737 003477 TSTB WRTYFG
013642 001115 BNE 50222$
3752 ;RETRIES WITH TCC4 ERRORS BY-PASS THIS SECTION
3753 013644 LET WTYWRD := CMDWRD ;SAVE WRITE COMMAND PACKET
013644 013737 003426 015316 MOV CMDWRD,WTYWRD
3754 013652 LET WTYCMD := CMDPKT ;
013652 013737 002314 015314 MOV CMDPKT,WTYCMD

```

3755	013660				LET WTYBRF := CMDPKT+CP.CNT	
	013660	013737	002322	015320		MOV CMDPKT+CP.CNT,WTYBRF
3756	013666				LET RWERR :B= RWERR + #1 ;LOG SUBR FLAG:	COUNT WRT ERRORS
	013666	105237	003503			INCB RWERR
3757	013672				LET WRTYFG :B= WRTYFG + #1 ;RETRY IN PROGRESS FLAG	INCB WRTYFG
	013672	105237	003477			
3758	013676				REPEAT	
	013676					50223\$:
3759	013676					
3760	013676			1\$:	LET WRTYCT(R5) := WRTYCT(R5) + #1 ;COUNT GLOBAL WRITE RETRIES	INC WRTYCT(R5)
	013676	005265	003250		LET RETRYC := #0 ;CLEAR # OF RETRIES PER RECORD	CLR RETRYC
3761	013702				LET RPTCNT :B= #0 ;CLEAR # OF REPEATS	CLRB RPTCNT
	013702	005037	003474			
3762	013706				JSR PC,WRTY ;CALL WRITE RETRY	
	013706	105037	003476		IF DEVTBL(R5) EQ #NINUSE THEN	
3763	013712	004737	014736			CMP DEVTBL(R5),#NINUSE
3764	013716					BNE 50224\$
	013716	026527	002536	177774	BR 3\$	
	013724	001001			ENDIF	
3765	013726	000552			UNTILB WRTYER EQ #0 OR @BTPT HIS #40.	50224\$:
3766	013730					;REPEAT RETRIES ON SAME RECORD
	013730					TSTB WRTYER
3767	013730	105737	003500			BEQ 50225\$
	013734	001404				CMP @BTPT,#40.
	013736	027727	167564	000050		BLO 50223\$
	013744	103754				50225\$:
	013746					;UNTIL RECOVERED OR 20 BAD SPOTS
3768	013746				IFB WRTYER NE #0 THEN	
3769	013746	105737	003500			TSTB WRTYER
	013752	001442				BEQ 50226\$
3770	013754				IF @BTPT HIS #40. THEN	;WHEN 20 BAD SPOTS LOGGED (TWR)
	013754	027727	167546	000050		CMP @BTPT,#40.
	013762	103405				BLO 50227\$
3771	013764				LET TC4STO :B= ERCVER	
	013764	113737	002207	003530		MOVB ERCVER,TC4STO
3772	013772				LET ERCVER :B= #0 ;REMOVE THE MASK	CLRB ERCVER
	013772	105037	002207		ENDIF	
3773	013776				IF @BTPT HIS #512. THEN	50227\$:
	013776	027727	167524	001000		CMP @BTPT,#512.
3774	013776	103423				BLO 50230\$
3775	014006				PRINTB #BTMSG2 ;BAD TAPE OVERFLOW MESSAGE	
	014006	012746	015407			MOV #BTMSG2,-(SP)
	014012	012746	000001			MOV #1,-(SP)
	014016	010600				MOV SP,R0
	014020	104414				TRAP C\$PNTB
	014022	062706	000004			ADD #4,SP
3776	014026	004737	017770		JSR PC,DROPU ;DROP THE UNIT.	
3777	014032				LET RECCNT(R5) := #0 ;	
	014032	005065	003330			CLR RECCNT(R5)
3778	014036				LET ERCVER :B= TC4STO ;RESTORE THE MASK	MOVB TC4STO,ERCVER
	014036	113737	003530	002207		
3779	014044				LET @TSDB(R5) := #RWCPK ;REWIND UNIT	MOV #RWCPK,@TSDB(R5)
	014044	012775	002334	002456		

```

3780 014052          ELSE
      014052 000402
      014054
3781 014054 000137 013676          JMP      1$          ;LOOP
3782 014060          ENDIF
      014060
3783 014060          ENDIF          50231$:
      014060
3784 014060          LET WRTYFG :B= #0          ;RETRY COMPLETE FLAG          50226$:
      014060 105037 003477          CLR      WRTYFG
3785 014064          LET MISCFG :B= MISCFG + #1          ;DO NOT HALT ON THIS CMD FLG          50221$:
      014064 105237 003551          INC      MISCFG
3786 014070          LET PCMDWD := WTYWRD          ;RESTORE ORIGINAL WRT CMD AFTER RECOVERY          50232$:
      014070 013737 015316 003432          MOV      WTYWRD,PCMDWD
3787 014076          ENDIF
      014076
3788 014076          ELSE          50222$:
      014076 000402
      014100
3789 014100          LET UNREC :B= UNREC + #1          ;          50232$:
      014100 105237 003504          ENDIF          INCB      UNREC
3790 014104
      014104
3791 014104          ELSE          50232$:
      014104 000463
      014106
3792 014106          JSR PC,RTLE          ;CHECK FOR RETRY LIMIT EXCEEDED.          50217$:
3793 014112          IF CMDLG GT #2 THEN          ;IF READ CMD THEN:
      014112 023727 003434 000002          CMP      CMDLG,#2
      014120 003411          BLE      50234$
3794 014122          LET R2 := #RRECL SHIFT -1          ;R2=READ RETRY COUNT LIMIT / 2
      014122 012702 000020          MOV      #RRECL,R2
      014126 006202          ASR      R2
3795 014130          IF RETRYC GE R2 THEN          ;IF RETRY COUNT IS MORE THAN HALF LIMIT:
      014130 023702 003474          CMP      RETRYC,R2
      014134 002403          BLT      50235$
3796 014136          LET CMDPKT := CMDPKT SET.BY #OPP.C          ;SET OPPOSITE BIT FOR RETRY2.
      014136 052737 020000 002314          BIS      #OPP.C,CMDPKT
3797 014144          ENDIF
      014144
3798 014144          ENDIF          50235$:
      014144
3799 014144          IF RETRYC EQ #0 ANDB ERerver NE #0 THEN          ;IF THIS IS THE ORIGINAL ERROR THEN:          50234$:
      014144 005737 003474          TST      RETRYC
      014150 001011          BNE      50236$
      014152 105737 002207          TSTB     ERerver
      014156 001406          BEQ      50236$
3800 014160          ERRSOFT 9,RERM,STAERM          ;REPORT RECOVERABLE ERROR          50236$:
      014160 104457          TRAP     C$ERSOFT
      014162 000011          .WORD   9
      014164 004644          .WORD   RERM
      014166 005640          .WORD   STAERM
3801 014170          JSR PC,RAMDUM          ;GO DO RAM DUMP
3802 014174          ENDIF          ;PROVIDED OPERATOR HAS ENABLED THE REPORT
      014174
3803 014174          LET RETRYC := RETRYC + #1          ;UPDATE RETRY COUNT.          50236$:
      014174 005237 003474          INC      RETRYC

```

```

3804 014200          LET CNDPKT := CNDPKT SET.BY #MOD.C1 ;SET RETRY BIT IN CMD PACKET.
      014200 052737 001000 002314          BIS #MOD.C1,CNDPKT
3805 014206          IFB IREC EQ #0 THEN ;IF ERROR RECOVERY ENABLED:
      014206 105737 002210          TSTB IREC
      014212 001016          BNE 50237$
3806 014214          IF DEVTBL(R5) EQ #NINUSE THEN
      014214 026527 002536 177774          CMP DEVTBL(R5),#NINUSE
      014222 001001          BNE 50240$
3807 014224 000413          BR 3$
3808 014226          ENDIF
3809 014226          LET ERRREC :B= ERRREC + #1 ;SET ERROR RECOVERY FLAG.
      014226 105237 003505          INCB ERRREC
3810 014232          POP R2,R2 ;POP 2 RTN ADRS FROM STACK.
      014232 012602          MOV (SP)+,R2
      014234 012602          MOV (SP)+,R2
3811 014236 004737 010774          JSR PC,EXCUTE ;GO EXECUTE THE RETRY COMMAND.
3812 014242 000137 011470          JMP GOWAIT ;GO WAIT FOR INTERRUPT + CHECK STATUS.
3813 014246          ELSE ;ELSE IF ERROR RECOVERY IS NOT ENABLED:
      014246 000402          BR 50241$
      014250          50237$:
3814 014250          LET UNREC :B= UNREC + #1 ;SET UNRECOVERABLE ERROR FLAG.
      014250 105237 003504          INCB UNREC
3815 014254          ENDIF
      014254          50241$:
3816 014254          ENDIF
      014254          50233$:
3817 014254 000207          3$: RTS PC ;RETURN

```

```

3819 ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 5, RECOVERABLE ERROR.
3820 ; TAPE POSITION HAS NOT CHANGED. RECOVERY PROCEDURE IS TO LOG THE
3821 ; ERROR AND RE-ISSUE THE ORIGINAL COMMAND.
3822 ; INPUTS:
3823 ; OUTPUTS:
3824 ; REGISTERS: R2,R4.
3825 ; CALLS: RTLE, EXCUTE, GOWAIT, DROPU.
3826
3827 014256 004737 014600 TCC5:: JSR PC,RTLE ;CHECK FOR RETRY LIMIT EXCEEDED
3828 014262 005737 003474 IF RETRYC EQ #0 THEN ;IF THIS IS THE ORIGINAL ERROR THEN:
014262 001006 TST RETRYC
014266 001006 BNE 50242$
3829 014270 ERRSOFT 10,RERM,STAERM ;REPORT RECOVERABLE ERROR.
014270 104457 TRAP C$ERSOFT
014272 000012 .WORD 10
014274 004644 .WORD RERM
014276 005640 .WORD STAERM
3830 014300 004737 014436 JSR PC,RAMDUM ;GO DO RAM DUMP
3831 014304 014304 ENDIF
3832 014304 005237 003474 LET RETRYC := RETRYC + #1 ;UPDATE RETRY COUNTER.
014304 005237 003474 IFB IREC EQ #0 THEN ;IF ERROR RECOVERY IS ENABLED:
3833 014310 105737 002210 TSTB IREC
014314 001016 BNE 50243$
3834 014316 105237 003505 LET ERRREC :B= ERRREC + #1 ;SET ERROR RECOVERY FLAG.
014316 105237 003505 LET RECCNT(R5) := RECCNT(R5) + #1 ;UPDATE REC COUNT
3835 014322 005265 003330 INC RECCNT(R5)
014322 005265 003330 LET @DATAWT := RECCNT(R5) ;AND INSERT IT INTO WRT BFR
3836 014326 016577 003330 167060 MOV RECCNT(R5),@DATAWT
3837 014334 012602 POP R2,R2 ;POP 2 RTN ADRS FROM STACK.
014334 012602 MOV (SP)+,R2
014336 012602 MOV (SP)+,R2
3838 014340 004737 010774 JSR PC,EXCUTE ;GO RE-ISSUE THE COMMAND.
3839 014344 000137 011470 JMP GOWAIT ;GO WAIT FOR INTERRUPT + CHECK STATUS.
3840 014350 000402 ELSE ;ELSE IF ERROR RECOVERY IS NOT ENABLED:
014350 000402 BR 50244$
014352 50243$:
3841 014352 105237 003504 LET UNREC :B= UNREC + #1 ;SET UNRECOVERABLE ERROR FLAG.
014352 105237 003504 INCB UNREC
3842 014356 014356 ENDIF
014356 014356 50244$:
3843 014356 000207 RTS PC ;RETURN.
3844
3845

```

```

3847 ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 6, UNRECOVERABLE ERROR.
3848 ; TAPE POSITION HAS BEEN LOST. THE ONLY VALID RECOVERY PROCEDURE
3849 ; IS TO REWIND AND START OVER AT BOT UNLESS THE TAPE HAS LABELS OR
3850 ; SEQUENCE NUMBERS. THIS DIAGNOSTIC WILL REWIND AND RETRY THE
3851 ; COMMAND ONLY IF DENSITY CHECK IS SET, OTHERWISE THE UNIT WILL BE
3852 ; DROPPED FROM THE TEST SEQUENCE.
3853 ; INPUTS:
3854 ; OUTPUTS:
3855 ; REGISTERS: R2, R4
3856 ; CALLS: RTLE, WSSR, EXCUTE, GOWAIT, DROPU
3857
3858 014360 TCC6:: LET @TSDB(R5) := #RWCPK ;ISSUE A REWIND COMMAND,
014360 012775 002334 002456 ; MOV #RWCPK,@TSDB(R5)
3859 014366 004737 012124 JSR PC,WSSR ;WAIT FOR SUBSYSTEM READY,
3860 014372 ERRDF 11,URERM,STAERM ;REPORT UNRECOVERABLE ERROR.
014372 104455 TRAP C$ERDF
014374 000013 .WORD 11
014376 004666 .WORD URERM
014400 005640 .WORD STAERM
3861 014402 004737 014436 JSR PC,RAMDUM ;GO DO RAM DUMP
3862 014406 004737 017770 JSR PC,DROPU ;DROP THE UNIT.
3863 014412 000207 RTS PC ;RETURN

```

```

3865 ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 7, FATAL SUBSYSTEM
3866 ; ERROR. THE SUBSYSTEM IS INCAPABLE OF PROPERLY PERFORMING
3867 ; COMMANDS OR AT LEAST ITS INTEGRITY IS SERIOUSLY QUESTIONABLE.
3868 ; REFER TO THE FATAL CLASS CODE FIELD IN THE TSSR REGISTER FOR
3869 ; ADDITIONAL INFORMATION ON THE TYPE OF FATAL ERROR.
3870 ; INPUTS:
3871 ; OUTPUTS:
3872 ; REGISTERS: R2, R4
3873 ; CALLS:
3874 ;
3875 014414 TCC7:: ERRDF 12,FATSM,STAERM ;REPORT FATAL SUBSYSTEM ERROR.
      014414 104455 TRAP C$ERRDF
      014416 000014 .WORD 12
      014420 004467 .WORD FATSM
      014422 005640 .WORD STAERM
3876 014424 004737 014436 JSR PC,RAMDUM ;GO DO RAM DUMP
3877 014430 004737 017770 JSR PC,DROPU ;DROP THE UNIT.
3878 014434 000207 RTS PC ;RETURN.
3879
3880
3881
3882
3883 014436 RAMDUM::
3884 014436 IFB RAMWRT NE #0 THEN
      014436 105737 002214 TSTB RAMWRT
      014442 001455 BEQ 50245$
3885 014444 PRINTX #RAMFHR
      014444 012746 005320 MOV #RAMFHR,-(SP)
      014450 012746 000001 MOV #1,-(SP)
      014454 010600 MOV SP,R0
      014456 104415 TRAP C$PNTX
      014460 062706 000004 ADD #4,SP
3886 014464 012737 000010 003406 MOV #8.,RAMSIZ ;RAM FIELD IS 8 BYTES LONG
3887 014472 012737 000020 003342 MOV #20,RAMHLD ;COMMAND PACKET ADDRESS
3888 014500 004737 016324 JSR PC,RAMER ;READ AND PRINT THEM
3889 014504 012737 000200 003342 MOV #200,RAMHLD ;CHARACTERISTICS PACKET ADDRESS
3890 014512 004737 016324 JSR PC,RAMER ;READ AND PRINT THEM
3891 014516 012737 000016 003406 MOV #14.,RAMSIZ ;RAM FIELD IS 8 BYTES LONG
3892 014524 012737 000214 003342 MOV #214,RAMHLD ;MESSAGE PACKET ADDRESS
3893 014532 004737 016324 JSR PC,RAMER ;READ AND PRINT THEM
3894 014536 012737 000020 003406 MOV #16.,RAMSIZ ;RAM FIELD IS SIXTEEN BYTES LONG
3895 014544 012737 000060 003342 MOV #60,RAMHLD ;SENCE BYTES ADDRESS
3896 014552 004737 016324 JSR PC,RAMER ;READ AND PRINT THEM
3897 014556 PRINTX #RAMLIN
      014556 012746 005462 MOV #RAMLIN,-(SP)
      014562 012746 000001 MOV #1,-(SP)
      014566 010600 MOV SP,R0
      014570 104415 TRAP C$PNTX
      014572 062706 000004 ADD #4,SP
3898 014576 ENDIF
      014576
3899 014576 000207 RTS PC 50245$:
3900

```



```

3902      ;      SUBROUTINE TO CHECK FOR RETRY LIMIT EXCEEDED. PRINTS ERROR MESSAGE
3903      ;      IF EXCEEDED AND DROP UNIT UNLESS COMMAND IS A READ.
3904      ;      INPUTS:
3905      ;      OUTPUTS:
3906      ;      REGISTERS:      R2, R4.
3907      ;      CALLS:      DROPU
3908
3909
3910 014600      RTLE:: IF CMDLG EQ #0 THEN      ;IF CMD IS NOT A READ OR WRITE THEN:
      014600      005737      003434      TST      CMDLG
      014604      001012      BNE      50246$
3911 014606      ERRDF 11,URERM,STAERM      ;REPORT UNRECOVERABLE ERROR.
      014606      104455      TRAP      C$ERDF
      014610      000013      .WORD      11
      014612      004666      .WORD      URERM
      014614      005640      .WORD      STAERM
3912 014616      004737      014436      JSR      PC,RAMDUM      ;GO DO RAM DUMP
3913 014622      004737      017770      JSR      PC,DROPU      ;DROP THE UNIT.
3914 014626      012602      POP      R2
3915 014630      000441      BR      RTLRTN      ;AND RETURN.
3916 014632      ENDIF      MOV      (SP)+,R2
3917 014632      LET RWERR :B= RWERR * #1      ;SET READ/WRITE ERROR FLAG.
      014632      105237      003503      INCB      RWERR
3918 014636      IF CMDLG EQ #2 THEN      ;IF CMD IS A WRT OR WTM:
      014636      023727      003434      000002      CMP      CMDLG,#2
      014644      001020      BNE      50247$
3919 014646      IF RETRYC EQ #WRECL THEN      ;IF RETRY COUNT HAS REACHED LIMIT:
      014646      023727      003474      000020      CMP      RETRYC,#WRECL
      014654      001013      BNE      50250$
3920 014656      LET UNREC :B= UNREC * #1      ;SET UNRECOVERABLE FLAG
3921 014662      105237      003504      INCB      UNREC
      014662      104455      ERRDF 14,RLEXM,STAERM      ;REPORT RETRY LIMIT EXCEEDED.
      014664      000016      TRAP      C$ERDF
      014666      004404      .WORD      14
      014670      005640      .WORD      RLEXM
3922 014672      004737      014436      JSR      PC,RAMDUM      ;GO DO RAM DUMP
3923 014676      004737      017770      JSR      PC,DROPU      ;DROP THE UNIT.
3924 014702      012602      POP      R2
3925 014704      ENDIF      MOV      (SP)+,R2
3926 014704      ELSE      ;ELSE - CMD IS A READ:
      014704      000413      BR      50251$
3927 014706      IF RETRYC EQ #RRECL THEN      ;IF RETRY COUNT HAS REACHED LIMIT:
      014706      023727      003474      000020      CMP      RETRYC,#RRECL
      014714      001007      BNE      50252$
3928 014716      LET UNREC :B= UNREC * #1      ;SET UNRECOVERABLE FLAG
3929 014722      105237      003504      INCB      UNREC
      014722      104456      ERRHRD 14,RLEXM,STAERM      ;REPORT RECOVERABLE ERROR.
      014724      000016      TRAP      C$ERHRD
      014726      004404      .WORD      14
      014730      005640      .WORD      RLEXM
      .WORD      STAERM

```

```

3930 014732          POP R2
3931 014732 012602          MOV    (SP)+,R2
3932 014734          ENDIF
3933 014734 000207          RTLRTN: RTS PC          ;RETURN
3934
3935          ; SUBR TO REWRITE A BAD, BUT RECOVERABLE WRITTEN RECORD.
3936          ; REWRITE RECORD ON SAME SPOT: REPEAT 4 TIMES.
3937          ; IF ALL 4 REPEATS GOOD, RECORD IS RECOVERED
3938          ; AND A RECOVERABLE WRITE ERROR IS LOGGED.
3939          ; IF ANY OF 4 REPEATS BAD, ERASE BAD RECORD, LOG SUSPECTED
3940          ; BAD SPOT, RETRY AGAIN. RETRY 4 TIMES, UP TO 4 REPEATS EACH.
3941          ; IF RECORD NOT GOOD AFTER 4 RETRIES, ERASE IT, EXIT WITH
3942          ; ERROR FLAG WRTYER SET, PRINTING RETRY FAILED.
3943          ; THIS ALL SCHEME IS REENTERED 20 TIMES MAX, IE 20 BAD
3944          ; SPOTS MAX ARE ALLOWED.
3945
3946          ; INPUTS:
3947          ; OUTPUTS:
3948          ; REGISTERS:      R3,R4
3949          ; CALLS:          BORERS, REWRT
3950
3951 014736          WRTY:: IF DEVTBL(R5) NE #NINUSE THEN          ;IF DRIVE NOT DROPPED
3952 014736 026527 002536 177774          CMP    DEVTBL(R5),#NINUSE
3953 014744 001537          BEQ    50253$
3954 014746          BEGIN RETRY
3955 014746          REPEAT
3956 014746 004737 015534          JSR PC,BORERS          ;BACKSPACE/ERASE ONE RECORD
3957 014752          LET WRTYER :B= #0          ;CLEAR WRITE RETRY ERROR
3958 014752 105037 003500          CLR B WRTYER
3959 014756 004737 015710          JSR PC,REWRT          ;REWRITE RECORD ON SAME SPOT
3960 014762          IF DEVTBL(R5) EQ #NINUSE THEN
3961 014770 026527 002536 177774          CMP    DEVTBL(R5),#NINUSE
3962 014772 001003          BNE    50260$
3963 014772 112737 000003 003476          LET RPTCNT :B= #3
3964 015000          ENDIF
3965 015000          LET RPTCNT :B= RPTCNT + #1          ;COUNT REPEATS
3966 015000 105237 003476          UNTILB RPTCNT EQ #4 ORB WRTYER NE #0          ;LIMIT: 4 REPEATS OR RECOVERED
3967 015004 123727 003476 000004          CMPB  RPTCNT,#4
3968 015012 001403          BEQ    50261$
3969 015014 105737 003500          TSTB  WRTYER
3970 015020 001752          BEQ    50257$
3971 015022          50261$:
3972 015022          ;
3973 015022          ;COUNT RETRIES
3974 015022 005237 003474          LET RETRYC := RETRYC + #1          INC    RETRYC
3975 015026          IF DEVTBL(R5) EQ #NINUSE THEN

```



```

3989 015174 105037 003476
      015200          ENDIF
3990 015200          UNTIL RETRYC EQ #4
      015200 023727 003474 000004
      015206 001257
3991 015210          END RETRY
      015210
3992 015210          IFB WRTYER NE #0 THEN
      015210 105737 003500
      015214 001413
3993 015216          IFB ERCVER NE #0 THEN
      015216 105737 002207
      015222 001410
3994 015224          PRINTB #BTMSG3
      015224 012746 015464
      015230 012746 000001
      015234 010600
      015236 104414
      015240 062706 000004
3995 015244          ENDIF
      015244
3996 015244          ENDIF
      015244
3997 015244          ENDIF
      015244
3998 015244          3$: LET -(SP) := #10.          ;LOOP CONTROL
      015244 012746 000012
      015250          REPEAT
      015250          DELAY 250.          ;WAIT FOR THE UNIT TO STOP
4000 015250 012727 000372
      015254 000000
      015256 013727 002116
      015262 000000
      015264 005367 177772
      015270 001375
      015272 005367 177756
      015276 001367
4001 015300          LET (SP) := (SP) - #1          ;ONE LESS ITERATION
      015300 005316
      015302          UNTIL (SP) EQ #0          ;'TILL ZERO!
      015302 005716
      015304 001361
4003 015306          LET SP := SP + #2          ;POP THE STACK
      015306 062706 000002
      015312 000207
      015312          RTS PC
4004
4005
4006
4007
4008
4009
4010
4011 015314 000000          WTYCMD: .WORD 0          ;STORAGE FOR WRITE CMD WHILE RETRYING
4012 015316 000000          WTYWRD: .WORD 0          ;STORAGE FOR WRITE CMD WORD WHILE RETRYING
4013 015320 000000          WTYBRF: .WORD 0          ;STORAGE FOR WRITE BPCR WHILE RETRYING
4014

```

```

      CLRB  PPTCNT
      50265$:
;LIMIT: 4 RETRIES
      CMP  RETRYC,#4
      BNE  50255$
      50254$:
      TSTB WRTYER
      BEQ  50271$
      TSTB ERCVER
      BEQ  50272$
;PRINT RETRY FAILED
      MOV  #BTMSG3,-(SP)
      MOV  #1,-(SP)
      MOV  SP,R0
      TRAP C$PNTB
      ADD  #4,SP
      50272$:
      50271$:
      50253$:
      MOV  #10,-(SP)
      50273$:
      MOV  #250,(PC)+
      .WORD 0
      MOV  L$DLY,(PC)+
      .WORD 0
      DEC  -6(PC)
      BNE  -4
      DEC  -22(PC)
      BNE  -20
      DEC  (SP)
      TST  (SP)
      BNE  50273$
      ADD  #2,SP

```

```

4015
4016 015322      045      101      123  BTMSG1: .ASCIZ  /%ASUSPECT BAD SPOT AFTER %D1%A RETRY, %D1%A REPEAT%N/
      015325      125      123      120
      015330      105      103      124
      015333      040      102      101
      015336      104      040      123
      015341      120      117      124
      015344      040      101      106
      015347      124      105      122
      015352      040      045      104
      015355      061      045      101
      015360      040      122      105
      015363      124      122      131
      015366      054      040      045
      015371      104      061      045
      015374      101      040      122
      015377      105      120      105
      015402      101      124      045
      015405      116      000
4017 015407      045      116      045  BTMSG2: .ASCIZ  /%N%ABAD TAPE OVERFLOW: CHANGE CARTRIDGE!%N%N/
      015412      101      102      101
      015415      104      040      124
      015420      101      120      105
      015423      040      117      126
      015426      105      122      106
      015431      114      117      127
      015434      072      040      103
      015437      110      101      116
      015442      107      105      040
      015445      103      101      122
      015450      124      122      111
      015453      104      107      105
      015456      041      045      116
      015461      045      116      000
4018 015464      045      101      122  BTMSG3: .ASCIZ  /%ARETRY FAILED ON BAD SPOT...ERASED!%N/
      015467      105      124      122
      015472      131      040      106
      015475      101      111      114
      015500      105      104      040
      015503      117      116      040
      015506      102      101      104
      015511      040      123      120
      015514      117      124      056
      015517      056      056      105
      015522      122      101      123
      015525      105      104      041
      015530      045      116      000
4019                                     .EVEN

```

```

4021 ; SUBR TO BACSPACE ONE RECORD
4022 ; IF THE ERASE FLAG IS SET, THEN ERASE THAT RECORD
4023 ; INPUTS: ERSFLG 1 = DO ERASE
4024 ; OUTPUTS:
4025 ; REGISTERS:
4026 ; CALLS: EXECUTE, GOWAIT, CKHAE
4027
4028 015534 BORERS:: LET PCMDWD := CMDWRD ;SET COMMAND TO SPACE REV
015534 013737 003426 003432 MOV CMDWRD,PCMDWD
4029 015542 LET CMDWRD := #SRR ;
015542 012737 104410 003426 MOV #SRR,CMDWRD
4030 015550 LET CNDPKT := CMDWRD CLR.BY #BRF.C ;
015550 013737 003426 002314 MOV CMDWRD,CNDPKT
015556 042737 004000 002314 BIC #BRF.C,CNDPKT
4031 015564 LET CMDSAV := CNDPKT ;
015564 013737 002314 003430 MOV CNDPKT,CMDSAV
4032 015572 LET CNDPKT+CP.ADL := #1 ;
015572 012737 000001 002316 MOV #1,CNDPKT+CP.ADL
4033 015600 LET CMDLG := #0 ;
015600 005037 003434 CLR CMDLG
4034 015604 JSR PC,CMDAC ;
4035 015610 JSR PC,EXECUTE ;
4036 015614 JSR PC,GOWAIT ;
4037 015620 JSR PC,CKHAE ;
4038 015624 IFB ERSFLG NE #0 THEN ;WHEN ERASE FLAG IS SET, DO ERASE
015624 105737 003545 TSTB ERSFLG
015630 001426 BEQ 50274$
4039 015632 LET PCMDWD := CMDWRD ;
015632 013737 003426 003432 MOV CMDWRD,PCMDWD
4040 015640 LET CMDWRD := #ERS ;
015640 012737 100411 003426 MOV #ERS,CMDWRD
4041 015646 LET CNDPKT := CMDWRD ;
015646 013737 003426 002314 MOV CMDWRD,CNDPKT
4042 015654 LET CMDSAV := CNDPKT ;
015654 013737 002314 003430 MOV CNDPKT,CMDSAV
4043 015662 JSR PC,CMDAC ;
4044 015666 JSR PC,EXECUTE ;
4045 015672 JSR PC,GOWAIT ;
4046 015676 JSR PC,CKHAE ;
4047 015702 LET ERSFLG :B= #0
015702 105037 003545 CLRB ERSFLG
4048 015706 ENDIF
015706 50274$:
4049 015706 000207 RTS PC
4050 ; SUBR TO REWRITE A BADLY WRITTEN RECORD
4051
4052 015710 REWRT: IF DEVTBL(R5) NE #NINUSE THEN ;IF DRIVE NOT DROPPED
015710 026527 002536 177774 CMP DEVTBL(R5),#NINUSE
015716 001441 BEQ 50275$
4053 015720 LET PCMDWD := CMDWRD ;RESTORE WRITE COMMAND PACKET
015720 013737 003426 003432 MOV CMDWRD,PCMDWD
4054 015726 LET CMDWRD := WTYWRD ;
015726 013737 015316 003426 MOV WTYWRD,CMDWRD
4055 015734 LET CNDPKT := WTYCMD ;
015734 013737 015314 002314 MOV WTYCMD,CNDPKT
4056 015742 LET CMDSAV := CNDPKT ;
015742 013737 002314 003430 MOV CNDPKT,CMDSAV

```

4057	015750				LET CMDPKT+CP.ADL := DATAWT	;		
	015750	013737	003414	002316			MOV	DATAWT,CMDPKT+CP.ADL
4058	015756				LET CMDPKT+CP.CNT := WTYBRF	;		
	015756	013737	015320	002322			MOV	WTYBRF,CMDPKT+CP.CNT
4059	015764				LET CMDLG := #2	;		
	015764	012737	000002	003434			MOV	#2,CMDLG
4060	015772	004737	007672		JSR PC,CMDAC			
4061	015776	004737	010774		JSR PC,EXCUTE			
4062	016002				IF DEVTBL(R5) NE #NINUSE THEN			
	016002	026527	002536	177774			CMP	DEVTBL(R5),#NINUSE
	016010	001404					BEQ	50276\$
4063	016012	004737	011470		JSR PC,GOWAIT	;		
4064	016016	004737	020274		JSR PC,CKHAE	;		
4065	016022				ENDIF			
	016022							50276\$:
4066	016022				ENDIF			
	016022							50275\$:
4067	016022	000207			RTS PC			

```

4069 ; SUBROUTINE TO LOG BYTES READ/WRITTEN.
4070 ; ALSO UPDATES READ/WRITE ERROR COUNTERS.
4071 ; INPUTS:
4072 ; OUTPUTS:
4073 ; REGISTERS: R2, R3, R4.
4074 ; CALLS:
4075
4076 016024 LOG:: IFB ERLOG EQ #0 THEN ;IF DATA AND ERRORS HAVE NOT BEEN LOGGED THEN:
      016024 105737 003502 ;IF DATA AND ERRORS HAVE NOT BEEN LOGGED THEN:
      016030 001126 ;IF DATA AND ERRORS HAVE NOT BEEN LOGGED THEN:
4077 016032 LET ERLOG :B= ERLOG + #1 ;SET LOG DONE FLAG.
      016032 105237 003502 ;SET LOG DONE FLAG.
4078 016036 LET R4 := CMDLG ;GET CURRENT CMD LOGGING CODE.
      016036 013704 003434 ;GET CURRENT CMD LOGGING CODE.
4079 016042 IF R4 NE #0 THEN ;IF THERE IS A CODE THEN:
      016042 005704 ;IF THERE IS A CODE THEN:
      016044 001520 ;IF THERE IS A CODE THEN:
4080 016046 LET R4 := R4 - #2 ;ADJUST THE CODE FOR TABLE INDEX.
      016046 162704 000002 ;ADJUST THE CODE FOR TABLE INDEX.
4081 016052 LET R2 := R5 + BINC(R4) + #CNTBGN ;R2 = ADR OF BYTE COUNT LSW.
      016052 010502 ;R2 = ADR OF BYTE COUNT LSW.
      016054 066402 016310 ;R2 = ADR OF BYTE COUNT LSW.
      016060 062702 002560 ;R2 = ADR OF BYTE COUNT LSW.
4082 016064 LET (R2) := (R2) + BRFCNT ;ADD BRFCNT TO LSW.
      016064 063712 003424 ;ADD BRFCNT TO LSW.
4083 016070 IF MSGPKT+MS.RFC LOS BRFCNT THEN ;IF THE RFC IS LOWER OR THE SAME AS BRFCNT THEN:
      016070 023737 002344 003424 ;IF THE RFC IS LOWER OR THE SAME AS BRFCNT THEN:
      016076 101002 ;IF THE RFC IS LOWER OR THE SAME AS BRFCNT THEN:
4084 016100 LET (R2) := (R2) - MSGPKT+MS.RFC ;SUBTRACT RFC FROM EXPECTED BRFCNT.
      016100 163712 002344 ;SUBTRACT RFC FROM EXPECTED BRFCNT.
4085 016104 ENDIF
      016104
4086 016104 LET R3 := R2 + #10 ;R3 = ADR OF 2ND WORD.
      016104 010203 ;R3 = ADR OF 2ND WORD.
      016106 062703 000010 ;R3 = ADR OF 2ND WORD.
4087 016112 WHILE (R2) GT #999. DO
      016112 ;WHILE (R2) GT #999. DO
      016112 021227 001747 ;WHILE (R2) GT #999. DO
      016116 003404 ;WHILE (R2) GT #999. DO
4088 016120 LET (R2) := (R2) - #1000. ;UPDATE BYTE COUNT
      016120 162712 001750 ;UPDATE BYTE COUNT
4089 016124 LET (R3) := (R3) + #1 ;2ND WORD.
      016124 005213 ;2ND WORD.
4090 016126 ENDDO
      016126 000771 ;2ND WORD.
      016130 ;2ND WORD.
4091 016130 LET R2 := R3 + #10 ;R2 = ADR OF 3RD WORD.
      016130 010302 ;R2 = ADR OF 3RD WORD.
      016132 062702 000010 ;R2 = ADR OF 3RD WORD.
4092 016136 WHILE (R3) GT #999. DO
      016136 ;WHILE (R3) GT #999. DO
      016136 021327 001747 ;WHILE (R3) GT #999. DO
      016142 003404 ;WHILE (R3) GT #999. DO
4093 016144 LET (R3) := (R3) - #1000. ;UPDATE BYTE COUNT
      016144 162713 001750 ;UPDATE BYTE COUNT
4094 016150 LET (R2) := (R2) + #1 ;3RD WORD.
      016150 005212 ;3RD WORD.

```



```

4095 016152          ENDDO
      016152 000771
      016154
4096 016154          LET R3 := R2 + #10          ;R3 = ADR OF 4TH WORD.
      016154 010203          ;R3 = ADR OF 4TH WORD.
      016156 062703 000010          ;R3 = ADR OF 4TH WORD.
4097 016162          WHILE (R2) GT #999. DO
      016162
      016162 021227 001747          ;R3 = ADR OF 4TH WORD.
      016166 003404          ;R3 = ADR OF 4TH WORD.
4098 016170          LET (R2) := (R2) - #1000. ;UPDATE BYTE COUNT
      016170 162712 001750          ;UPDATE BYTE COUNT
4099 016174          LET (R3) := (R3) + #1          ;4TH WORD.
      016174 005213          ;4TH WORD.
4100 016176          ENDDO
      016176 000771
      016200
4101 016200          IFB RWERR NE #0 THEN          ;IF R/W ERROR, UPDATE ERROR COUNT.
      016200 105737 003503          ;IF R/W ERROR, UPDATE ERROR COUNT.
      016204 001440          ;IF R/W ERROR, UPDATE ERROR COUNT.
4102 016206          LET R2 := R5 + EINC(R4) + #WRREC ;R2 = ADR OF COUNTER.
      016206 010502          ;R2 = ADR OF COUNTER.
      016210 066402 016316          ;R2 = ADR OF COUNTER.
      016214 062702 002720          ;R2 = ADR OF COUNTER.
4103 016220          IFB UNREC NE #0 THEN          ;IS THE ERROR UNRECOVERABLE?
      016220 105737 003504          ;IS THE ERROR UNRECOVERABLE?
      016224 001404          ;IS THE ERROR UNRECOVERABLE?
4104 016226          LET R2 := R2 + #10          ;YES, POINT TO NEXT COUNTER.
      016226 062702 000010          ;YES, POINT TO NEXT COUNTER.
4105 016232          LET (R2) := (R2) + #1          ;UPDATE THE ERROR COUNTER
      016232 005212          ;UPDATE THE ERROR COUNTER
4106 016234          ELSE          ;ELSE - IF ERROR IS RECOVERABLE:
      016234 000424          ;ELSE - IF ERROR IS RECOVERABLE:
      016236
4107 016236          LET (R2) := (R2) + #1          ;UPDATE THE ERROR COUNTER
      016236 005212          ;UPDATE THE ERROR COUNTER
4108 016240          IFB IREC EQ #0 THEN          ;IF ERROR RECOVERY IS ENABLED:
      016240 105737 002210          ;IF ERROR RECOVERY IS ENABLED:
      016244 001020          ;IF ERROR RECOVERY IS ENABLED:
4109 016246          IFB DROPED EQ #0 ANDB ERCVER NE #0 THEN ;IF UNIT HAS NOT BEEN DROPPED:
      016246 105737 003541          ;IF UNIT HAS NOT BEEN DROPPED:
      016252 001015          ;IF UNIT HAS NOT BEEN DROPPED:
      016254 105737 002207          ;IF UNIT HAS NOT BEEN DROPPED:
      016260 001412          ;IF UNIT HAS NOT BEEN DROPPED:
4110 016262          PRINTB #NURTY1,RETRYC          ;PRINT # OF RETRIES TO RECOVER
      016262 013746 003474          ;PRINT # OF RETRIES TO RECOVER
      016266 012746 005167          ;PRINT # OF RETRIES TO RECOVER
      016272 012746 000002          ;PRINT # OF RETRIES TO RECOVER
      016276 010600          ;PRINT # OF RETRIES TO RECOVER
      016300 104414          ;PRINT # OF RETRIES TO RECOVER
      016302 062706 000006          ;PRINT # OF RETRIES TO RECOVER
4111 016306          ENDIF          ;PROVIDED PRINT HAS BEEN ENABLED
      016306          ;PROVIDED PRINT HAS BEEN ENABLED
4112 016306          ENDIF
      016306          ;PROVIDED PRINT HAS BEEN ENABLED
4113 016306          ENDIF
      016306          ;PROVIDED PRINT HAS BEEN ENABLED

```

```

4114 016306          ENDIF
      016306
4115 016306          ENDIF
      016306
4116 016306          ENDIF
      016306
4117 016306 000207
4118
4119 016310 000000
      ;BINC:
4120 016312 000040
      0
      ;WRITE.
4121 016314 000100
      40
      ;READ REV.
      100
      ;READ FWD.
4122
      ;EINC:
4123 016316 000000
      INDEXES TO READ/WRITE ERROR COUNTERS.
      0
      ;WRITE.
4124 016320 000020
      20
      ;READ REV.
4125 016322 000040
      40
      ;READ FWD.
4126
4127
4128

```

50310\$:

50300\$:

50277\$:

```

4130          .SBTTL RAMER - READ AND DISPLAY SELECTED RAM
4131          ;+
4132          ;ROUTINE TO READ THE SELECTED RAM LOCATIONS
4133          ;-
4134 016324 010546          RAMER:: MOV     R5,-(SP)
4135 016326 010446          MOV     R4,-(SP)
4136 016330 010346          MOV     R3,-(SP)
4137 016332 010246          MOV     R2,-(SP)
4138 016334 010146          MOV     R1,-(SP)
4139 016336 012701 003346  MOV     #RAMDATA,R1      ;ADDRESS TO SAVE THE RAM DATA
4140 016342 013702 003342  MOV     RAMHLD,R2      ;BYTE ADDRESS OF THE FIRST RAM DATA
4141 016346 013703 003406  MOV     RAMSIZ,R3      ;SET THE SIZE OF THE READ UP
4142 016352 016504 002456  MOV     TSDB(R5),R4    ;MOV THE TSDB ADDRESS INTO R4
4143 016356 0052C4          INC     R4              ;ADD 1 TO IT
4144 016360 000240          10$:  NOP
4145 016362 004737 012124  JSR     PC,WSSR        ;WAIT FOR THE SSR TO SET
4146 016366 110214          MOVVB  R2,(R4)         ;SELECT NEXT RAM ADDRESS
4147 016370 004737 012124  JSR     PC,WSSR        ;WAIT FOR SSR TO SET
4148 016374 117521 002456  MOVVB  @TSBA(R5),(R1)+ ;READ THE RAM DATA
4149 016400 062702 000001  20$:  ADD     #1,R2          ;ADDRESS OF THE NEXT RAM LOCATION
4150 016404 077313          SOB    R3,10$         ;NUMBER OF LOCATIONS COUNTER
4151 016406 013704 003406  MOV     RAMSIZ,R4      ;GET THE RAM SIZE
4152 016412 013702 003342  MOV     RAMHLD,R2      ;GET THE STARTING RAM ADDRESS
4153 016416 060204          ADD    R2,R4          ;CALCULATE THE END ADDRESS
4154 016420 162704 000001  SUB    #1,R4           ;CORRECT VALUE OF PRINTOUT
4155 016424          PRINTX #RAMIOP,R2,R4 ;RAM ADDRESS = 10 - 17, ETC.
                                MOV     R4,-(SP)
                                MOV     R2,-(SP)
                                MOV     #RAMIOP,-(SP)
                                MOV     #3,-(SP)
                                MOV     SP,R0
                                TRAP   C$PNTX
                                ADD    #10,SP
4156 016450 012701 000010  MOV     #RAMDATA,R1    ;ADDRESS OF WHERE RAM DATA IS
4157 016454 013703 003406  MOV     RAMSIZ,R3      ;THE SIZE OF THE RAM FIELD READ
4158 016460 005004          30$:  CLR     R4              ;NO EXTRA DATA LEFT OVER
4159 016462 112104          MOVVB  (R1)+,R4        ;PICK UP BYTE OF RAM DATA
4160 016464 042704 177400  BIC    #177400,R4      ;GET RID OF SIGN EXTEND
4161 016470          PRINTX #RAMPD,R4    ;"010 211 111 222 377 000 123 134 ETC."
                                MOV     R4,-(SP)
                                MOV     #RAMPD,-(SP)
                                MOV     #2,-(SP)
                                MOV     SP,R0
                                TRAP   C$PNTX
                                ADD    #6,SP
4162 016512 077316          SOB    R3,30$         ;LOOP UNTIL ALL PRINTED
4163 016514 012601          MOV    (SP)+,R1
4164 016516 012602          MOV    (SP)+,R2
4165 016520 012603          MOV    (SP)+,R3
4166 016522 012604          MOV    (SP)+,R4
4167 016524 012605          MOV    (SP)+,R5
4168 016526 000207          50$:  RTS     PC              ;RETURN
4169          ; IF A WRITE/VERIFY COMMAND IS ISSUED, CONTROL IS THEN
4170          ; TRANSFERRED TO THIS SUBROUTINE TO READ REVERSE, CHECK DATA,
4171          ; READ FORWARD, CHECK DATA, THEN CONTINUE TO NEXT COMMAND.
4172          ; INPUTS:
4173          ; OUTPUTS:

```

```

4174 ; REGISTERS:
4175 ; CALLS: VFEXC.
4176
4177 016530 VFYDAT:: LET CMPDAT := #0 ;BTL
      016530 005037 003410 ;IF DATA IS TO BE VERIFIED:
4178 016534 IFB VFYFLG NE #0 THEN ;SAVE THE PREVIOUS COMMAND WORD.
      016534 105737 003534 ;COMMAND IS READ REV.
      016540 001437 ;SET UP CMD LOGGING INDEX.
4179 016542 IFB STREAM EQ #0 THEN ;GO READ ALL THE RECORDS REV.
      016542 105737 003544 ;BTL
      016546 001015
4180 016550 LET PCMDWD := CMDWRD ;SAVE THE PREVIOUS COMMAND WORD.
      016550 013737 003426 003432 ;COMMAND IS READ REV.
4181 016556 LET CMDWRD := #RDR ;SET UP CMD LOGGING INDEX.
      016556 012737 104401 003426 ;GO READ ALL THE RECORDS REV.
4182 016564 LET CMDLG := #4 ;BTL
      016564 012737 000004 003434
4183 016572 JSR PC,VFEXC
4184 016576 LET CMPDAT := #0
      016576 005037 003410
4185 016602 ENDIF
      016602
4186 016602 IF DEVTBL(R5) NE #NINUSE THEN
      016602 026527 002536 177774
      016610 001413
4187 016612 LET PCMDWD := CMDWRD ;SAVE THE PREVIOUS COMMAND WORD.
      016612 013737 003426 003432 ;COMMAND IS READ FWD.
4188 016620 LET CMDWRD := #RDF ;SET UP CMD LOGGING INDEX.
      016620 012737 104001 003426 ;GO READ ALL RECORDS FWD.
4189 016626 LET CMDLG := #6
      016626 012737 000006 003434
4190 016634 JSR PC,VFEXC
4191 016640 ENDIF
      016640
4192 016640 ENDIF
      016640
4193 016640 000207 RTS PC ;RETURN.

```

```

4195      ;      SUBROUTINE TO EXECUTE THE READ AND VERIFY, FORWARD OR REVERSE.
4196      ;      INPUTS:
4197      ;      OUTPUTS:
4198      ;      REGISTERS:      R2
4199      ;      CALLS:          CMDAC, FIRSTU, VFISU, NEXTU, CKHAE.
4200
4201 016642 VFEXC:: LET CMDPKT := CMDWRD CLR.BY #BRF.C ;COMMAND PACKET = READ REV OR FWD.
      016642 013737 003426 002314      MOV      CMDWRD,CMDPKT
      016650 042737 004000 002314      BIC      #BRF.C,CMDPKT
4202 016656      IFB SWBFLG NE #0 THEN      ;IF BYTES ARE TO BE SWAPPED:
      016656 105737 003536      TSTB     SWBFLG
      016662 001403      BEQ      50320$
4203 016664      LET CMDPKT := CMDPKT SET.BY #SWB.C ;SET SWAB BIT IN CMD PACKET.
      016664 052737 010000 002314      BIS      #SWB.C,CMDPKT
4204 016672      ENDIF
4205 016672      LET CMDSAV := CMDPKT      ;SAVE COMMAND PACKET 1ST WORD.
      016672 013737 002314 003430      MOV      CMDPKT,CMDSAV
4206 016700      MOV      DATARD,CMDPKT*CP.ADL ;SAVE BUFFER START ADDRESS.
4207 016706      LET NCNT := #0      ;CLEAR NUMBER OF OPERATIONS.
      016706 005037 003420      CLR      NCNT
4208 016712      WHILE NCNT LT NCNT1 DO      ;WHILE THERE ARE RECORDS REMAINING:
      016712      50321$:
      016712 023737 003420 003422      CMP      NCNT,NCNT1
      016720 002101      BGE      50322$
4209 016722      JSR PC,CMDAC      ;STORE CMD ASCII IN ERROR MSG.
4210 016726      TSTB STREAM      ;CHECK IF WE ARE STREAMING
4211 016732      BNE 1$      ;BRANCH OVER DEVTBL CHECK. THIS ENABLES
4212      ;US TO TEST ONE DRIVE AT A TIME.
4213 016734      JSR PC,FIRSTU      ;SET UP FOR FIRST UNIT.
4214 016740      WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE DEVICES REMAINING:
      016740      50323$:
      016740 026527 002536 177777      CMP      DEVTBL(R5),#END
      016746 001445      BEQ      50324$
4215 016750      1$:      IF #MOD.CO SETIN CMDWRD THEN ;IF CMD IS REVERSE THEN:
      016750 032737 000400 003426      BIT      #MOD.CO,CMDWRD
      016756 001421      BEQ      50325$
4216 016760      IF #X0.BOT NOTSETIN EOTFLG(R5) THEN ;IF NOT AT BOT
      016760 032765 000002 003516      BIT      #X0.BOT,EOTFLG(R5)
      016766 001014      BNE      50326$
4217 016770      IF #X0.EOT SETIN EOTFLG(R5) THEN ;BUT IF AT EOT
      016770 032765 000001 003516      BIT      #X0.EOT,EOTFLG(R5)
      016776 001406      BEQ      50327$
4218 017000      IFB ALLEOT NE #0 THEN      ;AND ALL OTHERS AT EOT
      017000 105737 003543      TSTB     ALLEOT
      017004 001402      BEQ      50330$
4219 017006      JSR PC,VFISU      ;THEN READ VERIFY
4220 017012      ENDIF      ;IF NOT ALL AT EOT, FREEZE UNIT(S) AT E
      017012      50330$:
4221 017012      ELSE      ;IF NOT AT BOT AND
      017012 000402      BR      50331$
      017014      50327$:
4222 017014      JSR PC,VFISU      ;NOT AT EOT, READ VFY
4223 017020      ENDIF
      017020      50331$:
4224 017020      ENDIF
      017020      50326$:

```

```

4225 017020                ELSE                                ;ELSE IF CMD IS NOT REVERSE:
      017020 000412                BR 50332$
      017022                50325$:
4226 017022                IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN
      017022 032765 000001 003516                BIT #X0.EOT,EOTFLG(R5)
      017030 001404                BEQ 50333$
      017032 032737 000001 003426                BIT #CMD.CO,CMDWRD
      017040 001002                BNE 50334$
      017042                50333$:
4227                ;IF NOT AT EOT OR NOT A MOTION CMD THEN:
4228 017042 004737 017126                JSR PC,VFISU                ;ISSUE CMD, CHECK STATUS AND DATA.
4229 017046                ENDIF
      017046                50334$:
4230 017046                ENDIF
      017046                50332$:
4231 017046 105737 003544                TSTB STREAM                ;CHECK FOR TEST OF ON UNIT AT A TIME.
4232 017052 001003                BNE 2$                ;BRANCH, IF STREAMING TESTS.
4233 017054 004737 017734                JSR PC,NEXTU                ;GO FIND THE NEXT UNIT.
4234 017060                ENDDO
      017060 000727                BR 50323$
      017062                50324$:
4235 017062 004737 020274                2$: JSR PC,CKHAE                ;CHECK FOR HALT AFTER EACH CMD.
4236 017066                IF DEVTBL(R5) EQ #NINUSE THEN ;IF DRIVES BEEN DROPPED EXIT
      017066 026527 002536 177774                CMP DEVTBL(R5),#NINUSE
      017074 001005                BNE 50335$
4237 017076                LET NCNT := NCNT1 - #1
      017076 013737 003422 003420                MOV NCNT1,NCNT
      017104 005337 003420                DEC NCNT
4238 017110                ENDIF
      017110                50335$:
4239 017110                LET NCNT := NCNT + #1                ;UPDATE THE RECORD COUNT.
      017110 005237 003420                INC NCNT
4240 017114                LET PCMDWD := CMDWRD                ;SAVE PREVIOUS COMMAND WORD.
      017114 013737 003426 003432                MOV CMDWRD,PCMDWD
4241
4242 017122                ENDDO
      017122 000673
      017124
4243 017124 000207                RTS PC                ;RETURN.
      BR 50321$
      50322$:
  
```

```

4245      :      SUBROUTINE TO ISSUE COMMAND, AWAIT INTERRUPT,
4246      :      CHECK STATUS, CHECK DATA.
4247      :      INPUTS:
4248      :      OUTPUTS:
4249      :      REGISTERS:      R2
4250      :      CALLS:          EXECUTE, GOWAIT, CKDATA.
4251
4252 017126 VFISU::
4253 017126      LET R2 := DATARD * #8.      ;INIT READ BUFFER POINTER.
      017126 013702 003416      MOV      DATARD,R2
      017132 062702 000010      ADD      #8.,R2
4254 017136      WHILE R2 NE DATARD DO      ;UNTIL 8 BYTES HAVE BEEN SET,
      017136      50336$:
      017136 020237 003416      CMP      R2,DATARD
      017142 001403      BEQ      50337$
4255 017144      LET -(R2) := #-1      ;INIT READ BUFFER.
      017144 012742 177777      MOV      #-1,-(R2)
4256 017150      ENDDO
      017150 000772      BR      50336$
      017152      50337$:
4257 017152      JSR PC,EXECUTE      ;GO EXECUTE THE COMMAND.
4258 017156      IFB DROPED EQ #0 THEN      ;IF UNIT HAS NOT BEEN DROPPED THEN:
      017156 105737 003541      TSTB     DROPED
      017162 001002      BNE      50340$
4259 017164      JSR PC,GOWAIT      ;GO WAIT FOR DONE BIT.
4260 017170      ENDIF
      017170      50340$:
4261 017170      IFB DROPED EQ #0 THEN      ;IF UNIT HAS NOT BEEN DROPPED THEN:
      017170 105737 003541      TSTB     DROPED
      017174 001025      BNE      50341$
4262 017176      IF #X0.BOT NOTSETIN EOTFLG(R5) THEN      ;WHEN NOT REVERSED INTO BOT, THEN
      017176 032765 000002 003516      BIT     #X0.BOT,EOTFLG(R5)
      017204 001021      BNE      50342$
4263 017206      IF L$TEST NE #3 THEN
      017206 023727 002114 000003      CMP     L$TEST,#3
      017214 001403      BEQ     50343$
4264 017216      JSR PC,CKDATA      ;GO VERIFY DATA.
4265 017222      ELSE
      017222 000412      BR      50344$
      017224      50343$:
4266 017224      INC CMPDAT      ;ONE MORE XFER BEFORE A COMPARISON
4267 017230      IF CMPDAT GT DATRAT THEN
      017230 023737 003410 003412      CMP     CMPDAT,DATRAT
      017236 003404      BLE     50345$
4268 017240      JSR PC,CKDATA
4269 017244      LET CMPDAT := #0
      017244 005037 003410      CLR     CMPDAT
4270 017250      ENDIF
      017250      50345$:
4271 017250      ENDIF
      017250      50344$:
4272 017250      ENDIF
      017250      50342$:
4273 017250      ENDIF
      017250      50341$:
4274 017250 000207      RTS PC
4275

```

```

4277      ;      SUBROUTINE TO COMPARE DATA BETWEEN READ AND WRITE BUFFERS
4278      ;      AND PRINT ERROR MESSAGE ON MISCOMPARE.
4279      ;      INPUTS:
4280      ;      OUTPUTS:
4281      ;      REGISTERS:      R2, R3, R4.
4282      ;      CALLS:      GCMDA
4283
4284      017252      CKDATA:: LET R3 := BRFCNT - MSGPKT+MS.RFC ;COMPUTE REC LENGTH READ
      017252      013703      003424      MOV      BRFCNT,R3
      017256      163703      002344      SUB      MSGPKT+MS.RFC,R3
4285      017262      IF R3 EQ #0 THEN      ;WHEN NO DATA RECEIVED
      017262      005703      TST      R3
      017264      001015      BNE      50346$
4286      017266      ERRHRD 17,WTVERM,DTAERM      ;PRINT ERROR AND EXIT
      017266      104456      TRAP      C$ERHRD
      017270      000021      .WORD      17
      017272      004260      .WORD      WTVERM
      017274      005472      .WORD      DTAERM
4287      017276      PRINTB #DTAER4      ;COMPARE ROUTINE
      017276      012746      005104      MOV      #DTAER4,-(SP)
      017302      012746      000001      MOV      #1,-(SP)
      017306      010600      MOV      SP,R0
      017310      104414      TRAP      C$PNTB
      017312      062706      000004      ADD      #4,SP
4288      017316      ELSE
      017316      000560      BR      50347$
4289      017320      IF R3 HI BRFCNT THEN      ;WHEN REC READ IS LONGER
      017320      020337      003424      CMP      R3,BRFCNT
      017324      101417      BLOS      50350$
4290      017326      ERRHRD 17,WTVERM,DTAERM      ;THAN EXPECTED, PRINT
      017326      104456      TRAP      C$ERHRD
      017330      000021      .WORD      17
      017332      004260      .WORD      WTVERM
      017334      005472      .WORD      DTAERM
4291      017336      PRINTB #DTAERS,CMDPKT+CP.CNT      ;AN ERROR MESSAGE
      017336      013746      002322      MOV      CMDPKT+CP.CNT,-(SP)
      017342      012746      005125      MOV      #DTAERS,-(SP)
      017346      012746      000002      MOV      #2,-(SP)
      017352      010600      MOV      SP,R0
      017354      104414      TRAP      C$PNTB
      017356      062706      000006      ADD      #6,SP
4292      017362      ELSE      ;AND EXIT ROUTINE
      017362      000536      BR      50351$
4293      017364      LET CKDCNT := R3 - #1      ;SAVE VERIFICATION LENGTH - 1.
      017364      010337      017662      MOV      R3,CKDCNT
      017370      005337      017662      DEC      CKDCNT
4294      017374      005037      017664      CLR CKDFF      ;CLEAR # OF BYTES IN ERROR COUNTER.
4295      017400      005002      CLR R2      ;INIT BYTE COUNTER
4296      017402      LET R3 := DATAW      ;GET WRITE BUFFER ADDRESS.
      017402      013703      003414      MOV      DATAW,R3
4297      017406      LET R4 := DATARD      ;GET READ BUFFER ADDRESS.
      017406      013704      003416      MOV      DATARD,R4
4298      017412      IFB T1SWB NE #0 THEN      ;WHEN RUNNING TEST 1-SUB
      017412      105737      003542      TSTB      T1SWB
      017416      001401      BEQ      50352$

```



```

4299 017420 000313 SWAB (R3) ;SWAP FIRST WORD OF WRT BFR
4300 017422 ENDIF ;WHICH CONTAINS THE RECORD COUNT
017422 50352$:
4301 017422 REPEAT ;REPEAT UNTIL ALL DATA IS COMPARED:
017422 50353$:
4302 017422 IF R2 EQ CKDCNT THEN ;IF THIS IS THE LAST BYTE THEN:
017422 020237 017662 CMP R2,CKDCNT
017422 001011 BNE 50354$
4303 017430 IFB SWBFLG NE #0 THEN ;IF BYTE SWAPPING IS ENABLED THEN:
017430 105737 003536 TSTB SWBFLG
017434 001406 BEQ 50355$
4304 017436 IF #BIT00 NOTSETIN CKDCNT THEN ;IF RECORD LENGTH IS ODD THEN:
017436 032737 000001 017662 BIT #BIT00,CKDCNT
017444 001002 BNE 50356$
4305 017446 105723 TSTB (R3)+ ;LAST BYTE WILL BE IN
4306 017450 105724 TSTB (R4)+ ;THE UPPER BYTE.
4307 017452 ENDIF 50356$:
017452 4308 017452 ENDIF 50355$:
017452 4309 017452 ENDIF 50354$:
4310 017452 121314 CMPB (R3),(R4) ;ARE THEY EQUAL.
4311 017454 001452 BEQ 3$ ;BR IF SO.
4312 017456 005737 017664 TST CKDFF ;1 ST TIME THRU?
4313 017462 001010 BNE 2$ ;BR IF NOT.
4314 017464 005265 003300 INC VFYCNT(R5) ;INC THE VERIFY ERROR COUNTER.
4315 017470 005265 003310 INC HRDCNT(R5) ;INC THE HARD ERROR COUNT.
4316 017474 ERRHRD 17,WTVERM,DTAERM ;REPORT WRITE/VERIFY ERROR.
017474 104456 TRAP C$ERHRD
017476 000021 .WORD 17
017500 004260 .WORD WTVERM
017502 005472 .WORD DTAERM
4317 017504 2$: LET CKDFF := CKDFF + #1 ;INCREMENT # OF BYTES IN ERROR.
017504 005237 017664 INC CKDFF
4318 017510 111437 003444 MOVB (R4),TIME1 ;SAVE WAS DATA FOR TYP0UT.
4319 017514 042737 177400 003444 BIC #177400,TIME1 ;CLEAR GARBAGE.
4320 017522 111337 003446 MOVB (R3),TIME2 ;SAVE SHOULD BE DATA FOR TYP0UT.
4321 017526 042737 177400 003446 BIC #177400,TIME2 ;CLEAR GARBAGE.
4322 017534 IF CKDFF LT #11. THEN ;IF ERROR BYTE COUNT IS LESS THAN 11:
017534 023727 017664 000013 CMP CKDFF,#11.
017542 002017 BGE 50357$
4323 017544 PRINTX #DTAER2,R2,<B,TIME1>,<B,TIME2> ;PRINT EXP + ACT DATA.
017544 005046 CLR -(SP)
017546 153716 003446 BISB TIME2,(SP)
017552 005046 CLR -(SP)
017554 153716 003444 BISB TIME1,(SP)
017560 010246 MOV R2,-(SP)
017562 012746 004773 MOV #DTAER2,-(SP)
017566 012746 000004 MOV #4,-(SP)
017572 010600 MOV SP,RO
017574 104415 TRAP C$PNTX
017576 062706 000012 ADD #12,SP
4324 017602 ENDIF
017602 4325 017602 105723 3$: TSTB (R3)+
4326 017604 105724 TSTB (R4)+
;UPDATE WRITE BUFFER ADDRESS.
;UPDATE READ BUFFER ADDRESS.

```

```

4327 017606 105722
4328 017610 020237 017662
      017610 003702
4329 017616 005237 017662
      017616 005237 017662
4330 017622 005737 017664
      017622 005737 017664
      017626 001414
4331 017630
      017630 013746 017662
      017634 013746 017664
      017640 012746 005042
      017644 012746 000003
      017650 010600
      017652 104414
      017654 062706 000010
4332 017660
      017660
4333 017660
      017660
4334 017660
      017660
4335 017660 000207
4336
4337 017662 000000
4338 017664 000000

      TSTB (R2)+
      UNTIL R2 GT CKDCNT
      LET CKDCNT := CKDCNT + #1
      IF CKDFF NE #0 THEN
          PRINTB #DTAER3,CKDFF,CKDCNT
      ENDIF
      ENDIF
      ENDIF
      RTS PC
      CKDCNT: .WORD 0
      CKDFF: .WORD 0

      ;UPDATE BYTE COUNTER.
      ;END OF DATA COMPARE REPEAT LOOP.
      CMP R2,CKDCNT
      BLE 50353$
      INC CKDCNT
      ;CKDCNT EQUALS RECORD LENGTH.
      ;IF COMPARE ERROR HAS OCCURED THEN:
      TST CKDFF
      BEQ 50360$
      ;PRINT # OF BYTES IN ERROR.
      MOV CKDCNT,-(SP)
      MOV CKDFF,-(SP)
      MOV #DTAER3,-(SP)
      MOV #3,-(SP)
      MOV SP,R0
      TRAP C$PNTB
      ADD #10,SP
50360$:
50351$:
50347$:
      ;OTHERWISE, RETURN.
      ;# OF BYTES TO BE VERIFIED -1.
      ;# OF BYTES IN ERROR COUNTER.
    
```

```

4340 ; SUBROUTINE TO FIND THE FIRST DEVICE IN THE TEST SEQUENCE.
4341 ; INPUTS:
4342 ; OUTPUTS:
4343 ; REGISTERS:
4344 ; CALLS:
4345 ;
4346 017666 FIRSTU::LET DROPED :B= #0 ;CLR UNIT DROPPED FLAG
017666 105037 003541 ;CLR UNIT DROPPED FLAG
4347 017672 LET R5 := #0 ;CLR DEVICE POINTER.
017672 005005 ;CLR DEVICE POINTER.
4348 017674 WHILE DEVTBL(R5) EQ #NINUSE DO ;WHILE DEVICES ARE NOT IN USE:
017674 ;WHILE DEVICES ARE NOT IN USE:
017674 026527 002536 177774 50361$:
017702 001003 CMP DEVTBL(R5),#NINUSE
4349 017704 LET R5 := R5 + #2 ;POINT TO NEXT DEVICE.
017704 062705 000002 BNE 50362$
4350 017710 ENDDO ADD #2,R5
017710 000771 BR 50361$
4351 017712 IF DEVTBL(R5) EQ #END THEN ;IF ALL UNITS HAVE BEEN DROPPED THEN:
017712 026527 002536 177777 50362$:
017720 001001 CMP DEVTBL(R5),#END
4352 017722 DOCLN ;DO CLEAN CODE AND TERMINATE PASS.
017722 104444 BNE 50363$
4353 017724 ENDDO TRAP C$DCLN
017724
4354 017724 LET L$LUN := DEVTBL(R5) ;SET UNIT # IN "HEADER" FOR ERROR REPORT
017724 016537 002536 002074 MOV DEVTBL(R5),L$LUN
4355 017732 000207 RTS PC ;RETURN WITH 1ST DEVICE IN R5.
4356
4357
4358
4359
4360
4361 ; SUBROUTINE TO FIND THE NEXT UNIT IN THE TEST CYCLE.
4362 ; INPUTS:
4363 ; OUTPUTS:
4364 ; REGISTERS:
4365 ; CALLS:
4366 ;
4367 017734 NEXTU:: LET DROPED :B= #0 ;CLR UNIT DROPPED FLAG
017734 105037 003541 ;CLR UNIT DROPPED FLAG
4368 017740 042705 177770 BIC #177770,R5
4369 017744 REPEAT ;REPEAT UNTIL THE NEXT DEVICE IS FOUND.
017744 ;REPEAT UNTIL THE NEXT DEVICE IS FOUND.
4370 017744 LET R5 := R5 + #2 ;UPDATE DEVICE TABLE POINTER.
017744 062705 000002 50364$:
4371 017750 UNTIL DEVTBL(R5) NE #NINUSE ;UPDATE DEVICE TABLE POINTER.
017750 026527 002536 177774 ADD #2,R5
017756 001772
4372 017760 LET L$LUN := DEVTBL(R5) ;SET UNIT # IN "HEADER" FOR ERROR REPORT
017760 016537 002536 002074 CMP DEVTBL(R5),#NINUSE
4373 017766 000207 BEQ 50364$
4374 MOV DEVTBL(R5),L$LUN
4375
4376
    
```

```

4378      ;      SUBROUTINE TO DROP A DEVICE FROM THE TEST SEQUENCE.
4379      ;      INPUTS:
4380      ;      OUTPUTS:
4381      ;      REGISTERS:
4382      ;      CALLS:          MOVMSG, PRXST, LOG
4383
4384 017770      DROPU:: LET R5SAVE := R5
017770      010537 003460
4385 017774      LET FTLCNT(R5) := FTLCNT(R5) + #1 ;INCREMENT THE FATAL ERROR COUNT.
017774      005265 003320
4386 020000      LET R4 := MSGPKT+MS.XS3 CLR.BY #377 ;GET UDIAG ERROR CODE FROM XSTAT3.
020000      013704 002354
020004      042704 000377
4387 020010      LET R3 := MSGPKA(R5) ;ADR OF THIS UNIT'S MSG PACKET.
020010      016503 002506
4388 020014      LET R2 := #0 ;CLR COUNTER.
020014      005002
4389 020016      WHILE R2 NE #MSGCNT DO ;WHILE THERE ARE MORE LOCATIONS:
020016
020016      020227 000016
020022      001405
4390 020024      LET (R3)+ := #-1 ;INIT THE MSG PACKET WITH ALL 1'S
020024      012723 177777
4391 020030      LET R2 := R2 + #2 ;UPDATE COUNTER.
020030      062702 000002
4392 020034      ENDDO
020034      000770
020036
4393 020036      LET @TSDB(R5) := #GSCPK ;INITIATE A GET STATUS COMMAND.
020036      012775 002324 002456
4394 020044      JSR PC,WSSR ;WAIT A WHILE FOR SSR=1
020044      004737 012124
4395 020050      JSR PC,MOVMSG ;MOVE MSG PACKET TO COMMON AREA.
020050      004737 012224
4396 020054      IF R4 EQ #X3.RNY THEN ;IF WE HAVE A CAPSTAN RUNAWAY THEN:
020054      020427 157400
020060      001005
4397 020062      ERRDF 16,RNYM,STAERM ;REPORT CAPSTAN RUNAWAY WITH TACH CNT.
020062      104455
020064      000020
020066      004600
020070      005640
4398 020072      ELSE ;ELSE-IF NOT A RUNAWAY:
020072      000402
020074
4399 020074      JSR PC,PRXST ;PRINT EXTENDED STATUS REGISTERS.
020074      004737 020212
4400 020100      ENDIF
020100
4401 020100      IFB RECLOG NE #0 THEN ;IF THE RECORD HAS BEEN LOGGED THEN:
020100      105737 003501
020104      001404
4402 020106      LET DROPED :B= DROPED + #1 ;SET UNIT DROPPED FLAG.
020106      105237 003541
4403 020112      JSR PC,LOG ;LOG DATA BYTES + RD/WR ERRORS.
020112      004737 016024
4404 020116      ENDIF
020116
4405 020116      DORPT ;PRINT PERFORMANCE REPORT
020116      104424
4406 020120      DROPUA: IF PASCNT(R5) NE #0 THEN

```



```

4426 ; SUBROUTINE TO PRINT EXTENDED STATUS REGISTERS.
4427 ; INPUTS:
4428 ; OUTPUTS:
4429 ; REGISTERS:
4430 ; CALLS:
4431 ;
4432 PRXST:: PRINTX #GETSTM
      020212 012746 005253      MOV #GETSTM,-(SP)
      020212 012746 000001      MOV #1,-(SP)
      020216 012746 000001      MOV SP,R0
      020222 010600      TRAP C$PNTX
      020224 104415      ADD #4,SP
      020226 062706 000004      PRINTX #STAERS,MSGPKT+MS.XS0,MSGPKT+MS.XS1,MSGPKT+MS.XS2,MSGPKT+MS.XS3
4433 020232      MOV MSGPKT+MS.XS3,-(SP)
      020232 013746 002354      MOV MSGPKT+MS.XS2,-(SP)
      020236 013746 002352      MOV MSGPKT+MS.XS1,-(SP)
      020242 013746 002350      MOV MSGPKT+MS.XS0,-(SP)
      020246 013746 002346      MOV #STAERS,-(SP)
      020252 012746 006465      MOV #5,-(SP)
      020256 012746 000005      MOV SP,R0
      020262 010600      TRAP C$PNTX
      020264 104415      ADD #14,SP
      020266 062706 000014      RTS PC
4434 020272 000207
4435
4436
4437
4438
4439 ; SUBROUTINE TO HALT AFTER EACH COMMAND.
4440 ; INPUTS:
4441 ; OUTPUTS:
4442 ; REGISTERS: R3, R4
4443 ; CALLS:
4444 ;
4445 CKHAE:: IFB HAE NE #0 THEN ;IF HALT FLAG IS SET:
      020274 105737 002206      TSTB HAE
      020274 001430      BEQ 50375$
4446 020302      IFB MISCFG EQ #0 THEN ;
      020302 105737 003551      TSTB MISCFG
      020306 001023      BNE 50376$
4447 020310      MANUAL ;IS MANUAL INTERVENTION ALLOWED?
      020310 104450      TRAP C$MANI
4448 020312      BNCOMPLETE CKHRTN ;BR IF NOT.
      020312 103023      BCC CKHRTN
4449 020314      LET R4 := CMDWRD ;COMMAND WORD.
      020314 013704 003426      MOV CMDWRD,R4
4450 020320 004737 007746      JSR PC,GCMDA ;FETCH ADR OF CMD ASCII.
4451 020324      LET HALTM :B= (R3)+ ;MOVE CMD ASCII
      020324 112337 004136      MOV (R3)+,HALTM
4452 020330      LET HALTM+1 :B= (R3)+ ;
      020330 112337 004137      MOV (R3)+,HALTM+1
4453 020334      LET HALTM+2 :B= (R3) ;INTO MESSAGE.
      020334 111337 004140      MOV (R3),HALTM+2
4454 020340      GMANIL HALTM,TIME1,1,YES ;HALT - WAIT FOR AN OEPRTOR INPUT.
      020340 104443      TRAP C$GMAN
      020342 000404      BR 10000$
      020344 003444      .WORD TIME1
      020346 000130      .WORD T$CODE

```



```

4461 ; SUBROUTINE TO CREATE THE SEQUENCE FOR A WRITE TAPE MARK
4462 ; COMMAND. WILL EXECUTE COMMAND TO UUT.
4463 ; INPUTS:
4464 ; OUTPUTS: CMDSEQ
4465 ; CALLS: SETUP, CMDAC, EXCUTE, GOWAIT
4466
4467 020364 WRITEM::
4468 020364 LET R1 := #CMDSEQ
020364 012701 003554
4469 020370 LET (R1)+ := #WTM ;COMMAND MOV #CMDSEQ,R1
020370 012721 100011 ;BRF MOV #WTM,(R1)+
4470 020374 LET (R1)+ := #1 ;ITERATIONS MOV #1,(R1)+
020374 012721 000001 ;PATTERN MOV #1,(R1)+
4471 020400 LET (R1)+ := #END ;TERMINATOR MOV #END,(R1)+
020400 012721 000001
4472 020404 LET R1 := #CMDSEQ ;TOP OF BUFFER MOV #CMDSEQ,R1
4473 020406 JSR PC, SETUP ;SET UP THE TABLE
020406 012721 177777 JSR PC, CMDAC ;LOAD THE ASCII
4474 020412 JSR PC, EXCUTE ;ISSUE THE WTM COMMAND
020412 012701 003554 JSR PC, GOWAIT ;WAIT FOR THE COMMAND TO FINISH
4475 020416 RTS PC ;RETURN TO CALLER
020416 004737 010000
4476 020422 .EVEN
020422 004737 007672
4477 020426
020426 004737 010774
4478 020432
020432 004737 011470
4479 020436
020436 000207
4480
4481
4482
4483 020440 ENDMOD

```



```

4495
4496
4497      .TITLE MISCELLANEOUS SECTIONS
4506      .SBTTL REPORT CODING SECTION
4507 020440      BGNMOD
4508
4509
4510      ;**
4511      ; THE REPORT CODING SECTION CONTAINS THE
4512      ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
4513      ;--
4514 020440      BGNRPT
4515      020440      L$RPT::
4521 020440      LET      RSSAVE := R5          ;SAVE CURRENT DEVICE POINTER.
4522 020444 010537 003460      MOV      R5,RSSAVE
4523 020450      JSR      PC,FIRSTU          ;FIND THE FIRST UNIT.
4523 020450      WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
4523 020450      50400$:
4523 020450 026527 002536 177777      CMP      DEVTBL(R5),#END
4523 020456 001562      BEQ      50401$
4524 020460      PRINTS      #RPT1A,DEVTBL(R5),PASCNT(R5),RECCNT(R5)
4524 020460 016546 003330      MOV      RECCNT(R5),-(SP)
4524 020464 016546 003260      MOV      PASCNT(R5),-(SP)
4524 020470 016546 002536      MOV      DEVTBL(R5),-(SP)
4524 020474 012746 021302      MOV      #RPT1A, -(SP)
4524 020500 012746 000004      MOV      #4, -(SP)
4524 020504 010600      MOV      SP,R0
4524 020506 104416      TRAP     C$PNTS
4524 020510 062706 000012      ADD      #12,SP
4525 020514      PRINTS      #RPT1B,WRBC+30(R5),WRBC+20(R5),WRBC+10(R5),WRBC(R5)
4525 020514 016546 002560      MOV      WRBC(R5),-(SP)
4525 020520 016546 002570      MOV      WRBC+10(R5),-(SP)
4525 020524 016546 002600      MOV      WRBC+20(R5),-(SP)
4525 020530 016546 002610      MOV      WRBC+30(R5),-(SP)
4525 020534 012746 021357      MOV      #RPT1B, -(SP)
4525 020540 012746 000005      MOV      #5, -(SP)
4525 020544 010600      MOV      SP,R0
4525 020546 104416      TRAP     C$PNTS
4525 020550 062706 000014      ADD      #14,SP
4526 020554      PRINTS      #RPT1C,RRBC+30(R5),RRBC+20(R5),RRBC+10(R5),RRBC(R5)
4526 020554 016546 002620      MOV      RRBC(R5),-(SP)
4526 020560 016546 002630      MOV      RRBC+10(R5),-(SP)
4526 020564 016546 002640      MOV      RRBC+20(R5),-(SP)
4526 020570 016546 002650      MOV      RRBC+30(R5),-(SP)
4526 020574 012746 021430      MOV      #RPT1C, -(SP)
4526 020600 012746 000005      MOV      #5, -(SP)
4526 020604 010600      MOV      SP,R0
4526 020606 104416      TRAP     C$PNTS
4526 020610 062706 000014      ADD      #14,SP
4527 020614      PRINTS      #RPT1D,RFBC+30(R5),RFBC+20(R5),RFBC+10(R5),RFBC(R5)
4527 020614 016546 002660      MOV      RFBC(R5),-(SP)
4527 020620 016546 002670      MOV      RFBC+10(R5),-(SP)
4527 020624 016546 002700      MOV      RFBC+20(R5),-(SP)
4527 020630 016546 002710      MOV      RFBC+30(R5),-(SP)
4527 020634 012746 021501      MOV      #RPT1D, -(SP)
4527 020640 012746 000005      MOV      #5, -(SP)

```

```

020644 010600
020646 104416
020650 062706 000014
4528 020654 PRINTS #RPT1F,WRREC(R5),RRREC(R5),RFREC(R5)
020654 016546 002760
020660 016546 002740
020664 016546 002720
020670 012746 021605
020674 012746 000004
020700 010600
020702 104416
020704 062706 000012
4529 020710 PRINTS #RPT1G,WRUNR(R5),RRUNR(R5),RFUNR(R5)
020710 016546 002770
020714 016546 002750
020720 016546 002730
020724 012746 021656
020730 012746 000004
020734 010600
020736 104416
020740 062706 000012
4530 020744 IFB BADTSW NE #0 THEN ;
020744 105737 002211
020750 001402
4531 020752 004737 021034 JSR PC,BTRPT ;GO PRINT BAD TAPE SPOTS WHEN
4532 020756 ENDF ;ENABLED
020756 50402$:
4533 020756 PRINTS #RPT1I,SCCNT(R5),HRDCNT(R5),FTLCNT(R5),VFYCNT(R5)
020756 016546 003300
020762 016546 003320
020766 016546 003310
020772 016546 003270
020776 012746 022046
021002 012746 000005
021006 010600
021010 104416
021012 062706 000014
4534 021016 004737 017734 JSR PC,NEXTU ;FIND THE NEXT UNIT.
4535 021022 ENDDO
021022 000612
021024 BR 50400$
4536 021024 LET R5 := R5SAVE ;RESTORE CURRENT DEVICE POINTER.
021024 013705 003460 MOV R5SAVE,R5
4537 021030 EXIT RPT
021030 000167
021032 001206 .WORD J$JMP
. WORD L10011-2-.

4538
4539 ; SUBR TO PRINT BAD TAPES SPOTS DURING THE REPORT PRINTS
4540 ; WRITE RETRIES: CUMULATIVE COUNT
4541 ; BAD TAPE SPOTS: COUNT PER TAPE PASS ONLY, NOT CUMULATIVE.
4542 ; COUNT OF RECOVERABLE WRITE ERRORS EXCLUDES BAD TAPE SPOTS.
4543
4544 021034 BTRPT: PRINTS #RPT1E,WRTYCT(R5) ;PRINT GLOBAL WRITE RETRY COUNT
021034 016546 003250
021040 012746 021727
021044 012746 000002
021050 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$PNTS
ADD #12,SP
TSTB BADTSW
BEQ 50402$
50402$:
MOV VFYCNT(R5),-(SP)
MOV FTLCNT(R5),-(SP)
MOV HRDCNT(R5),-(SP)
MOV SCCNT(R5),-(SP)
MOV #RPT1I, -(SP)
MOV #5, -(SP)
MOV SP,RO
TRAP C$PNTS
ADD #14,SP
BR 50400$
50401$:
MOV R5SAVE,R5
. WORD J$JMP
. WORD L10011-2-.
MOV WRTYCT(R5),-(SP)
MOV #RPT1E, -(SP)
MOV #2, -(SP)
MOV SP,RO

```

021052	104416								
4545	021054	062706	000006					TRAP	C:PNTS
	021060			LET BTPT := BTADDR(R5)	;BTPT IS BOTH THE BAD TAPE SPOT			ADD	#6,SP
4546	021060	016537	002550	003526				MOV	COUNTER
	021066			LET R3 := @BTPT SHIFT -1	;AND THE LOGGING INDEX			MOV	BTADDR(R5),BTPT
	021066	017703	162434					MOV	@BTPT,R3
4547	021072	006203						ASR	R3
	021074			PRINTS @RPT1J,R3	;PRINT # OF BAD TAPE SPOTS				
	021074	010346						MOV	R3,-(SP)
	021076	012746	021757					MOV	@RPT1J,-(SP)
	021102	012746	000002					MOV	#2,-(SP)
	021106	010600						MOV	SP,RO
	021110	104416						TRAP	C:PNTS
4548	021112	062706	000006					ADD	#6,SP
	021116			IF R3 NE #0 THEN	;PRINT RECORD # IF BAD SPOTS DETECTED				
	021116	005703						TST	R3
4549	021120	001457						BEQ	50403\$
	021122			IF R3 HI #20. THEN					
	021122	020327	000024					CMP	R3,#20.
4550	021126	101402						BLOS	50404\$
	021130			LET R3 := #20.	;20 BAD SPOTS IS THE LIMIT				
4551	021130	012703	000024					MOV	#20.,R3
	021134			ENDIF					
4552	021134			PRINTS @CRLFSP					50404\$:
	021134	012746	005312					MOV	@CRLFSP,-(SP)
	021140	012746	000001					MOV	#1,-(SP)
	021144	010600						MOV	SP,RO
	021146	104416						TRAP	C:PNTS
4553	021150	062706	000004					ADD	#4,SP
	021154			LET R4 := BTPT * #2	;FETCH A BAD SPOT ID				
	021154	013704	003526					MOV	BTPT,R4
4554	021160	062704	000002					ADD	#2,R4
	021164			LET R2 := #0	;R2 = PRINT COUNT PER LINE: 10 MAX				
4555	021164	005002		REPEAT				CLR	R2
	021166								
4556	021166			PRINTS @RPT1K,(R4)	;PRINT A BAD SPOT ID				50405\$:
	021166	011446						MOV	(R4),-(SP)
	021170	012746	022037					MOV	@RPT1K,-(SP)
	021174	012746	000002					MOV	#2,-(SP)
	021200	010600						MOV	SP,RO
	021202	104416						TRAP	C:PNTS
4557	021204	062706	000006					ADD	#6,SP
	021210			LET R2 := R2 + #1	;COUNT PRINTS				
4558	021210	005202						INC	R2
	021212			LET R4 := R4 + #2	;NEXT				
4559	021212	062704	000002					ADD	#2,R4
	021216			IF R2 EQ #10. THEN					
	021216	020227	000012					CMP	R2,#10.
4560	021222	001014						BNE	50406\$
	021224			PRINTS @CRLFSP	;GO TO NEXT PRINT LINE PAST 10 PRINTS				
	021224	012746	005312					MOV	@CRLFSP,-(SP)
	021230	012746	000001					MOV	#1,-(SP)
	021234	010600						MOV	SP,RO
	021236	104416						TRAP	C:PNTS
	021240	062706	000004					ADD	#4,SP

```

4561 021244          LET R3 := R3 - #10.      ;ADJUST BAD SPOT COUNT
      021244 162703 000012          SUB      #10.,R3
4562 021250          LET R2 := R2 - #10.      ;ADJUST PRINT COUNT
      021250 162702 000012          SUB      #10.,R2
4563 021254          ENDIF
      021254
4564 021254          UNTIL R2 EQ R3          ;LIMIT: # OF BAD SPOTS
      021254 020203          50406$:
      021256 001343          CMP      R2,R3
4565 021260          ENDIF
      021260
4566 021260          PRINTS #CRLF           ;
      021260 012746 005307          50403$:
      021264 012746 000001          MOV      #CRLF,-(SP)
      021270 010600          MOV      #1,-(SP)
      021272 104416          MOV      SP,R0
      021274 062706 000004          TRAP   C$PNTS
4567 021300 000207          ADD      #4,SP

```

```

4568
4580          .NLIST BEX
4581 021302      045      116      045 RPT1A: .ASCIZ /#N#AUNIT #D1#S3#APASS:#D5#S3#ARECORD:#D5#N/
4582 021357      045      101      102 RPT1B: .ASCIZ /#ABYTES WRITTEN #D3#A,#Z3#A,#Z3#A,#Z3#N/
4583 021430      045      101      102 RPT1C: .ASCIZ /#ABYTES READ REV #D3#A,#Z3#A,#Z3#A,#Z3#N/
4584 021501      045      101      102 RPT1D: .ASCIZ /#ABYTES READ FWD #D3#A,#Z3#A,#Z3#A,#Z3#N/
4585 021551      045      123      062 .ASCIZ /#S23#AWRT#S4#ARDR#S4#ARDF#N/
4586 021605      045      101      122 RPT1F: .ASCIZ /#ARECOVERABLE ERRORS #D5#S2#D5#S2#D5#N/
4587 021656      045      101      125 RPT1G: .ASCIZ /#AUNRECOVERABLE ERRORS #D5#S2#D5#S2#D5#N/
4588 021727      045      101      127 RPT1E: .ASCIZ /#AWRITE RETRIES#S8#D5#N/
4589 021757      045      116      045 RPT1J: .ASCIZ /#N#D3#A BAD SPOTS THIS PASS PRECEDING RECORD #:/
4590 022037      045      104      065 RPT1K: .ASCIZ /#D5#S1/
4591 022046      045      101      123 RPT1I: .ASCII "#ASPEC COND#S3#AHARD#S3#AFATAL#S3#ACOMPARE#N"
4592 022122      045      123      063 .ASCIZ /#S3#D5#S3#D5#S3#D5#S3#D5#N#N/
4593 022157      045      116      045 TAPCAP: .ASCIZ /#N#ATAPE IN CARTRIDGE MUST BE 600' IN LENGTH.#N#N/
4594          .LIST BEX
4595          .EVEN
4596 022242          ENDRPT

```

```

022242 104425          L10011:
022242          TRAP   C$RPT

```

```

4597
4598          ;LOAD DEVICE PROTECTION TABLE
4599          ;TABLE FOR SUPERVISOR TO IDENTIFY THE P-TBL FOR THE LOAD DEV
4600          ;THE SUPERVISOR USES THE TBL TO WARN THE OPERATOR WHEN HE TRIES TO TEST THE LOAD DEV
4601

```

```

4602 022244          BGNPROT
      022244          L$PROT::
4603 022244 000000          .WORD 0          ;P-TBL OFFSET OF TSSR, THE TK25 CSR
4604 022246 177777          .WORD -1         ;P-TBL OFFSET OF MASS BUS UNIT #: -1 = NOT A MASS BUS DEV
4605 022250 177777          .WORD -1         ;P-TBL OFFSET OF DRIVE #: -1 = NONE, ONE DRIVE PER UNIBUS AD
4606 022252          ENDRPT

```

```

4608          .SBTTL INITIALIZE SECTION
4609
4610          ;**
4611          ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
4612          ; AT THE BEGINNING OF EACH PASS.
4613          ;--
4614
4615 022252          BGNINIT
4616 022252          L$INIT::
4626 022252 032727 000003 002314  INIT10: IF #BIT0!BIT1 SETIN #CMDPKT THEN ;IF CMD PACKET IS NOT ON MODULO 4 BOUNDARY:
4627 022260 001421          ERRSF 1,CMDPKM          ;PRINT ERROR MSG,          BIT          #BIT0!BIT1,#CMDPKT
4628 022262          TRAP          C$ERSF
4629 022262 104454          ;GO TO SUPERVISOR, WAIT 2 SECONDS.          .WORD          1
4630 022264 000001          MOV          #20.,(PC)+          .WORD          CMDPKM
4631 022266 004176          .WORD          0          .WORD          0
4632 022270 000000          DELAY 20.          ;GO TO SUPERVISOR, WAIT 2 SECONDS.          MOV          #20.,(PC)+
4633 022272 012727 000024          .WORD          0          MOV          L$DLY,(PC)+
4634 022276 000000          .WORD          0          DEC          -6(PC)
4635 022300 013727 002116          .WORD          0          BNE          -.4
4636 022304 000000          .WORD          0          DEC          -22(PC)
4637 022306 005367 177772          .WORD          0          BNE          -.20
4638 022312 001375          BR          INIT10          ;
4639 022314 005367 177756          ENDIF          ;
4640 022320 001367          ;
4641 022322 000753          ;
4642 022324          50407$:
4643 022324          IFB CLRFLG NE #0 THEN          ;IF CLR COUNTERS FLAG SET:
4644 022324 105737 002204          TSTB          CLRFLG
4645 022330 001413          BEQ          50410$
4646 022332 105037 002204          CLR          CLRFLG          ;INIT CLR FLAG.
4647 022336 005002          CLR          R2
4648 022340          WHILE R2 NE #CNTLEN DO
4649 022340          50411$:
4650 022340 020227 000550          CMP          R2,#CNTLEN
4651 022344 001405          BEQ          50412$
4652 022346          ;CLR ALL STATISTICAL COUNTERS.
4653 022346 005062 002560          CLR          WRBC(R2)
4654 022352 062702 000002          ADD          #2,R2
4655 022352 062702 000002          BR          50411$
4656 022356          ENDDO          50412$:
4657 022356 000770          ;
4658 022360          50410$:
4659 022360          IFB RRANV NE #0 THEN          ;IF RESET RANDOM VARIABLE FLAG IS SET THEN:
4660 022360 105737 002205          TSTB          RRANV
4661 022364 001406          BEQ          50413$
4662 022366          ;RESET RANDOM BASE #.
4663 022366 012737 153624 003440          MOV          #RRANV,RRANV
4664 022374          ;RESET RANDOM SAVE LOCATION.

```

```

4644 022374 012737 032561 003442          MOV      #RANSC,RANS
4645 022402          ENDIF
4645 022402          CLOCK L,R5          ;LINE CLOCK PRESENT?
4645 022402 012700 000114          ;50413$:
4645 022406 104462          MOV      #'L,RO
4645 022410 010005          TRAP    C$CLCK
4646 022412          BNCOMPLETE 1$          ;SKIP NEXT IF NOT
4646 022412 103036          MOV      (R5)+,LCSR
4647 022414          LET LCSR := (R5)+
4648 022420          LET LBRVEC := (R5)+
4648 022420 012537 003464          MOV      (R5)+,LBRVEC
4649 022424          LET LBRVEC := LBRVEC SHIFT 5
4649 022424 006337 003466          ASL     LBRVEC
4649 022430 006337 003466          ASL     LBRVEC
4649 022434 006337 003466          ASL     LBRVEC
4649 022440 006337 003466          ASL     LBRVEC
4649 022444 006337 003466          ASL     LBRVEC
4650 022450          LET LVECT := (R5)+
4650 022450 012537 003470          MOV      (R5)+,LVECT
4651 022454          LET LHERTZ := (R5)+
4651 022454 012537 003472          MOV      (R5)+,LHERTZ
4652 022460          SETVEC LVECT,#CLKSVC,LBRVEC ;SET UP FOR LINE CLOCK SERVICE
4652 022460 013746 003466          MOV     LBRVEC,-(SP)
4652 022464 012746 010746          MOV     #CLKSVC,-(SP)
4652 022470 013746 003470          MOV     LVECT,-(SP)
4652 022474 012746 000003          MOV     #3,-(SP)
4652 022500 104437          TRAP    C$SVEC
4652 022502 062706 000010          ADD     #10,SP
4653 022506 000402          BR      2$
4654 022510          1$: LET LCSR := #0
4654 022510 005037 003464          CLR     LCSR
4655 022514          2$:
4656 022514          READEF #EF.START          ;READ START COMMAND EVENT FLAG.
4656 022514 012700 000040          MOV     #EF.START,RO
4656 022520 104447          TRAP    C$REFG
4657 022522          BNCOMPLETE INIT15          ;BRANCH IF NOT STARTING.
4657 022522 103030          MOV     INIT15
4658 022524          LET STAFLG :B= STAFLG + #1          ;SET START COMMAND FLAG.
4658 022524 105237 003546          BCC    INIT15
4659 022530          LET HERE :B= #0          ;RESET THE TEST 3 ASCII SEMAPHORE
4659 022530 105037 003340          INCB   STAFLG
4660 022534          LET R5 := #6          CLR     HERE
4661 022540          REPEAT          ;INITIATE UNIT NUMBER TABLE
4661 022540          LET DEVTBL(R5) := #NINUSE          ;BY STORING NOT IN USE IN EACH LOCATION.
4662 022540 012765 177774 002536          MOV     #NINUSE,DEVTBL(R5)
4663 022546          LET R5 := R5 - #2
4664 022552          UNTIL R5 EQ #0
4664 022552 005705          SUB     #2,R5
4664 022554 001371          TST    R5
4665 022556          LET R5 := L$UNIT SHIFT 1          BNE    50414$
4665 022556 013705 002012          MOV     L$UNIT,R5
4665 022562 006305          ASL    R5

```

```

4666 022564          REPEAT          ;STORE ALL UNIT
      022564          ;NUMBERS IN DEVTBL. 50415$:
4667 022564          LET R5 := R5 - #2
      022564 162705 000002          LET DEVTBL(R5) := R5 SHIFT -1
4668 022570          ;HAS THERE BE A POWER FAILURE?
      022570 010565 002536          ;IF SO - SET THE START FLAG.
      022574 006265 002536          ;IF SO - SET THE POWER FAIL FLAG.
4669 022600          UNTIL R5 EQ #0
      022600 005705          ;READ AND STORE FLAGS SET BY OPERATOR
      022602 001370          ;CLEAR EVENT FLAG
4670          ;BRANCH IF NOT.
4671 022604          INIT15: READEF #EF.PWR
      022604 012700 000034          ;IF POWER FAIL HAS NOT OCCURRED THEN:
      022610 104447          ;UPDATE PASS COUNT WHEN
4672 022612          BNCOMPLETE INIT16
      022612 103004          ;SUPERVISOR IS IN NEW PASS
4673 022614          LET STAFLG :B= STAFLG + #1
      022614 105237 003546          ;AND DIAG WAS NEITHER STARTED
4674 022620          LET PWRFLG :B= PWRFLG + #1
      022620 105237 003547          ;NOR
4675          ;RESTARTED
4676 022624          INIT16: RFLAGS OPFLAG
      022624 104421          ;DO IT
      022626 010037 003552          ;SET 1ST PASS IF NEW PASS AND
4677 022632          LET R3 := #0
      022632 005003          ;RESTARTING
4678 022634          IFB PWRFLG EQ #0 THEN
      022634 105737 003547          ;STARTING
4679 022642          READEF #EF.NEW
      022642 012700 000035          ;SET 1ST PASS IF NEW PASS AND
4680 022650          IFCOND CS THEN
      022650 103014          ;RESTARTING
4681 022652          IFB STAFLG EQ #0 THEN
      022652 105737 003546          ;STARTING
4682 022660          READEF #EF.RES
      022660 012700 000037          ;SET 1ST PASS IF NEW PASS AND
4683 022666          IFCOND CC THEN
      022666 103402          ;RESTARTING
4684 022670          LET R3 := COMP R3
      022670 005103          ;STARTING
4685 022672          ELSE
      022672 000401          ;SET 1ST PASS IF NEW PASS AND
4686 022674          LET R3 := R3 + #1
      022674 005203          ;RESTARTING
4687 022676          ENDIF
      022676          ;STARTING
4688 022676          ELSE
      022676 000401          ;SET 1ST PASS IF NEW PASS AND
4689 022700          LET R3 := R3 + #1
      022700 005203          ;RESTARTING
4690 022702          ENDIF

```

```

4691 022702          022702          ENDIF          ;DO NOT UPDATE IT ON CONTINUE
4692 022702          022702          ENDIF          ;OR ON POWER FAIL
4693 022702 004737 017666 JSR PC,FIRSTU ;INIT DEVICE POINTER.
4694 022706          022706          LET R2 := #0   ;INIT DEVICE COUNTER.
4695 022710          022710          WHILE DEVTBL(R5) NE #END DO
                                022710 026527 002536 177777
                                022711 001450
4696 022720          022720          LET R2 := R2 + #1
4697 022722          022722          LET R0 := R5 SHIFT -1
                                022723 010500
                                022724 006200
4698 022726          022726          GPHARD R0,R0      ;GET HARDWARE P TABLE FROM SUPER.
4699 022730          022730          IFCOND CS THEN
                                022731 104442
                                022732 103036
4700 022732          022732          LET TSSR(R5) := (R0)      ;SAVE TSSR ADDRESS.
4701 022736          022736          LET TSDB(R5) := (R0)+ - #2 ;SAVE TSDB ADDRESS.
                                022737 012065 002456
                                022738 162765 000002 002456
4702 022750          022750          LET TSVCT(R5) := (R0)      ;SAVE INTERRUPT VECTOR ADDRESS.
4703 022754          022754          SETVEC TSVCT(R5),TS4INT(R5),#INTPRI ;SET UP INTERUPT PROCESSING CONDITIONS.
                                022755 012746 000340
                                022756 016546 002516
                                022757 016546 002476
                                022758 012746 000003
                                022759 104437
                                022760 062706 000010
4704 023002          023002          IF R3 NE #0 THEN      ;ACTUAL PASSCOUNT UPDATE PER R3
                                023003 005703
                                023004 001410
4705 023006          023006          IF R3 LT #0 THEN
                                023007 005703
                                023008 002003
4706 023012          023012          LET PASCNT(R5) := PASCNT(R5) + #1
4707 023016          023016          ELSE
                                023017 000403
4708 023020          023020          LET PASCNT(R5) := #1
4709 023026          023026          ENDIF
                                023027 012765 000001 003260
4710 023026          023026          ENDIF
4711 023026          023026          ENDIF
4712 023026          023026          LET RECCNT(R5) := #0      ;CLEAR RECORD COUNT
4713 023032          023032          JSR PC,NEXTU          ;DO IT FOR ALL DEVICES.

```



```

4714 023036          ENDDO
      023036 000724
      023040
      BR      50424$
      50425$:
4715
4716 023040          IF R2 EQ #0 THEN          ;IF THERE ARE NO UNITS:
      023040 005702
      023042 001026
      PRINTF #AUDRPM          ;PRINT ALL UNITS DROPPED,
      MOV     #AUDRPM,-(SP)
      MOV     #1,-(SP)
      MOV     SP,R0
      TRAP   C$PNTF
      ADD     #4,SP
4717 023044          PRINTF #AUDRPM          ;GO TO SUPERVISOR, WAIT 2 SECONDS.
      023044 012746 004741
      023050 012746 000001
      023054 010600
      023056 104417
      023060 062706 000004
      DELAY 20.
      MOV     #20.,(PC)+
      .WORD  0
      MOV     L$DLY,(PC)+
      .WORD  0
      DEC     -6(PC)
      BNE     -.4
      DEC     -22(PC)
      BNE     -.20
4718 023064          DELAY 20.
      023064 012727 000024
      023070 000000
      023072 013727 002116
      023076 000000
      023100 005367 177772
      023104 001375
      023106 005367 177756
      023112 001367
4719 023114          BREAK          ;GO TO SUPERVISOR, CHECK TTY.
      023114 104422
      DOCLN          ;DO CLEAN CODE + ABORT PASS.
      TRAP   C$BRK
4720 023116          DOCLN
      023116 104444
      TRAP   C$DCLN
4721 023120          ENDF
      023120
      50432$:
4722
4723 023120          SETPRI #PRI00          ;LOWER CPU PRIORITY TO 0
      023120 012700 000000
      023124 104441
      MOV     #PRI00,R0
      TRAP   C$SPRI
4724 023126          IFB IREC EQ #0 AND #ADR NOTSETIN OPFLAG THEN ;IF ERROR RECOVERY IS ENABLED
      023126 105737 002210
      023132 001145
      023134 032737 000020 003552
      023142 001141
      TSTB   IREC
      BNE   50433$
      BIT   #ADR,OPFLAG
      BNE   50433$
4725 023144          JSR PC,FIRSTU          ;AND AUTO-DROP NOT CALLED, THEN SET UP FOR FIRST UNI
4726 023150          WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
      023150
      023150 026527 002536 177777
      023156 001533
      50434$:
      CMP   DEVTBL(R5),#END
      BEQ   50435$
4727 023160          BEGIN COUNTER          ;START 3.5 MINUTE COUNTER
4728 023160          INCR TIME1 FROM #1 TO #25 BY #1
      023160 012737 000001 003444
      023166 000402
      023170
      MOV     #1,TIME1
      BR     50437$
      50440$:
      INC   TIME1
      50437$:
      CMP   TIME1,#25
      BGT   50441$
4729 023204          LET @TSDB(R5) := #GSCP ;AND GET UNITS STATUS
      023204 012775 002324 002456
      MOV     #GSCP,@TSDB(R5)
4730 023212          DELAY 250.          ;WAIT 25 MSEC.
      023212 012727 000372
      023216 000000
      023220 013727 002116
      MOV     #250.,(PC)+
      .WORD  0
      MOV     L$DLY,(PC)+

```



```

4744 023416 000664          ENDINC          50450$:
023416 000664          ENDINC          BR      50440$
023420          END COUNTER          50441$:
4745 023420          END COUNTER          50436$:
023420          IF TIME1 GT #25 THEN      ;IF OFF LINE FOR 3.5 MINUTES
023420 023727 003444 000025          CMP      TIME1,#25
023426 003404          JSR PC,MOVMSG          BLE      50451$
4747 023430 004737 012224          JSR PC,MOVMSG          ;GET MESSAGE PACKET
4748 023434 004737 013330          JSR PC,TCC1          ;PRINT ERROR AND DROP OFF LINE UNIT
4749 023440          ENDIF
023440          JSR PC,NEXTU          50451$:
4750          ENDDO          ;REPEAT UNTIL ON LINE OR TIMED OUT.
4751 023440 004737 017734          JSR PC,NEXTU          ;SET UP FOR NEXT UNIT.
4752 023444 000641          ENDDO
023444 000641          BR      50434$
023446          ENDIF          50435$:
4753 023446          IFB PWRFLG EQ #0 THEN      50433$:
023446          MEMORY DATAW          TSTB     PWRFLG
023446 105737 003547          MEMORY DATAW          BNE      50452$
023452 001026          LET DATARD := DATAW + #DATCNT      ;REQUEST MEMORY FROM SUPER FOR RD/WR BUFFERS.
4755 023454 104431          LET DATARD := DATAW + #DATCNT      ;SET RD BFR AD
023456 010037 003414          LET DATARD := DATAW + #DATCNT      MOV      DATAW,DATARD
4756 023462 013737 003414 003416          IF @DATAWT LT #DATCNT THEN      ;WHEN NOT ENOUGH FREE MEMO AVAILABLE
023470 062737 010000 003416          PRINTF #MEMOM          ;WARN OPERATOR
4757 023476 027727 157712 010000          PRINTF #MEMOM          ;WARN OPERATOR
023504 002011          DOCLN          ;AND ABORT PASS
4758 023506 012746 023554          DOCLN          ;AND ABORT PASS
023512 012746 000001          ENDIF          ;DIAG MUST BE RE-LOADED IN A CPU WITH LARGER MEMO
023516 010600          ENDIF          50453$:
023520 104417          ENDIF          50452$:
023522 062706 000004          LET CHGFLG :B= #0          ;CLR CHANGE CMD SEQ TBL FLAG.
4759 023526 104444          LET R3 := #ENDFLG          CLRB     CHGFLG
023530          JSR PC,CLRERR          ;CLEAR ALL FLAGS.
4760 023530          LET PWRFLG :B= #0          ;CLEAR THE POWER FAIL FLAG.
023530          EXIT INIT          CLRB     PWRFLG
4761 023530          EXIT INIT          TRAP     C$EXIT
023530          TRAP     C$EXIT          .WORD     L10013-.
4762          045          101          106 MEMOM: .ASCII /%AFREE MEMO TOO SMALL FOR RD-WR BFRS#N/
4763 023530 105037 002217          LET CHGFLG :B= #0          ;CLR CHANGE CMD SEQ TBL FLAG.
023530 105037 002217          LET R3 := #ENDFLG          CLRB     CHGFLG
4764 023534 012703 003546          JSR PC,CLRERR          ;CLEAR ALL FLAGS.
023534 012703 003546          LET PWRFLG :B= #0          ;CLEAR THE POWER FAIL FLAG.
4765 023540 004737 012110          JSR PC,CLRERR          ;CLEAR ALL FLAGS.
4766 023544 105037 003547          LET PWRFLG :B= #0          ;CLEAR THE POWER FAIL FLAG.
4767          EXIT INIT          CLRB     PWRFLG
4768 023550          EXIT INIT          TRAP     C$EXIT
023550 104432          EXIT INIT          .WORD     L10013-.
023552 000104          EXIT INIT          TRAP     C$EXIT
4780 023554 045          101          106 MEMOM: .ASCII /%AFREE MEMO TOO SMALL FOR RD-WR BFRS#N/

```

	023557	122	105	105
	023562	040	115	105
	023565	115	117	040
	023570	124	117	117
	023573	040	123	115
	023576	101	114	114
	023601	040	106	117
	023604	122	040	122
	023607	104	055	127
	023612	122	040	102
	023615	106	122	123
	023620	045	116	
4781	023622	045	101	122
	023625	105	055	114
	023630	117	101	104
	023633	040	111	116
	023636	040	114	101
	023641	122	107	105
	023644	122	040	115
	023647	105	115	117
	023652	045	116	000

.ASCIZ /*ARE-LOAD IN LARGER MEMO*/

4782				.EVEN
4783				ENDINIT
4784	023656			
	023656		L10013:	
	023656	104411		

TRAP C\$INIT

```

4786          .SBTTL AUTO DROP SECTION
4787
4788          ;**
4789          ;SECTION EXECUTED AFTER THE INIT CODE WHEN "ADR" FLAG IS SET BY OPERATOR
4790          ;SECTION CHEKS FOR A VALID INTERFACE LOCATION. DROPS UNIT IF NO RESPONSE
4791          ;FROM INTERFACE
4792          ;--
4793
4794 023660          BGNAUTO
4795          L$AUTO::
4796 023660 004737 017666          JSR PC,FIRSTU          ;FIND FIRST UNIT
4797 023664          WHILE DEVTBL(R5) NE #END DO          ;
4798          023664 026527 002536 177777          50454$:
4799          023672 001525          CMP          DEVTBL(R5),#END
4798 023674          LET TRAPD4 :B= #0          ;
4799 023700          SETVEC #4,#TRAP4,#PRI07          ;SET VECTOR 4          CLRB          TRAPD4
4799 023700 012746 000340          MOV          #PRI07,-(SP)
4799 023704 012746 024274          MOV          #TRAP4,-(SP)
4799 023710 012746 000004          MOV          #4,-(SP)
4799 023714 012746 000003          MOV          #3,-(SP)
4799 023720 104437          TRAP          C$SVEC
4799 023722 062706 000010          ADD          #10,SP
4800 023726          LET R2 := @TSSR(R5)          ;ADDRESS TK25 INTERFACE
4801 023726 017502 002466          MOV          @TSSR(R5),R2
4801 023732          CLRVEC #4          ;CLEAR VECTOR AT 4
4801 023732 012700 000004          MOV          #4,R0
4802 023736 104436          TRAP          C$CVEC
4802 023740          IFB TRAPD4 NE #0 THEN
4802 023740 105737 003550          TSTB          TRAPD4
4803 023744 001423          BEQ          50456$
4803 023746          LET FTLCNT(R5) := FTLCNT(R5) + #1          INC          FTLCNT(R5)
4804 023752          PRINTF #AUTODM,TSSR(R5)          ;PRINT ERROR
4804 023752 016546 002466          MOV          TSSR(R5),-(SP)
4804 023756 012746 024150          MOV          #AUTODM,-(SP)
4804 023762 012746 000002          MOV          #2,-(SP)
4804 023766 010600          MOV          SP,R0
4804 023770 104417          TRAP          C$PNTF
4804 023772 062706 000006          ADD          #6,SP
4805 023776          LET DROPN := DEVTBL(R5)          ;SAVE # OF UNIT TO BE DROPPED.
4806 023776 016537 002536 020210          MOV          DEVTBL(R5),DROPN
4806 024004          LET R0 := R5 SHIFT -1          ;R0=LOGICAL DEVICE NUMBER
4807 024006 010500          MOV          R5,R0
4807 024010 006200          ASR          R0
4807 024010          DODU R0          ;DROP THE UNIT: EXEC BGNDU-ENDDU CODE IF IDU = 0
4808 024012          ELSE          TRAP          C$DODU
4808 024012 000452          BR          50457$
4809 024014          LET @TSDB(R5) := #GSCP          ;SEND GET STATUS COMMAND
4809 024014 012775 002324 002456          MOV          #GSCP,@TSDB(R5)
4810 024022 004737 012124          JSR PC,WSSR          ;WAIT
4811 024026          IF #TS.SSR SETIN @TSSR(R5) THEN
4811 024026 032775 000200 002466          BIT          #TS.SSR,@TSSR(R5)

```

```

4812 024034 001423
      024036 032775 000100 002466
      024044 001416
4813 024046
      024046 005265 003320
4814 024052
      024052 016546 002536
      024056 012746 005223
      024062 012746 000002
      024066 010600
      024070 104417
      024072 062706 000006
4815 024076 004737 020120
4816 024102
      024102
4817 024102
      024102 000416
      024104
4818 024104
      024104 005265 003320
4819 024110
      024110 016546 002536
      024114 012746 024244
      024120 012746 000002
      024124 010600
      024126 104417
      024130 062706 000006
4820 024134 004737 020120
4821 024140
      024140
4822 024140
      024140
4823 024140 004737 017734
4824 024144
      024144 000647
      024146
4825
4826 024146
      024146
      024146 104461
4827
4828 024150 045 101 102
      024153 125 123 040
      024156 124 122 101
      024161 120 040 101
      024164 124 040 045
      024167 117 066 045
      024172 116
4829 024173 045 101 111
      024176 116 124 105
      024201 122 106 101
      024204 103 105 040
      024207 102 101 104
      024212 040 117 122
      024215 040 116 117
      024220 124 040 123
    
```

```

IF #TS.OFL SETIN @TSSR(R5) THEN

LET FTLCNT(R5) := FTLCNT(R5) + #1

PRINTF #OFLINM,DEVTBL(R5)

JSR PC,DROPUA
ENDIF

ELSE

LET FTLCNT(R5) := FTLCNT(R5) + #1

PRINTF #NRDYM,DEVTBL(R5)

JSR PC,DROPUA
ENDIF

ENDIF

JSR PC,NEXTU
ENDDO

ENDAUTO

L10014:
AUTODM: .ASCII /*ABUS TRAP AT %06%N/

.ASCIIZ /*AINTERFACE BAD OR NOT SET TO ABOVE AD%N/
    
```

```

BEQ 50460$
BIT #TS.OFL,@TSSR(R5)
BEQ 50461$
INC FTLCNT(R5)
MOV DEVTBL(R5),-(SP)
MOV #OFLINM,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #6,SP

50461$:
BR 50462$
50460$:
INC FTLCNT(R5)
MOV DEVTBL(R5),-(SP)
MOV #NRDYM,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #6,SP

50462$:
50457$:
BR 50454$
50455$:
TRAP C$AUTO
    
```

	024223	105	124	040	
	024226	124	117	040	
	024231	101	102	117	
	024234	126	105	040	
	024237	101	104	045	
	024242	116	000		
4830	024244	045	101	125	NRDYM: .ASCIZ /#AUNIT #D1#A NOT RDY#N/
	024247	116	111	124	
	024252	040	045	104	
	024255	061	045	101	
	024260	040	116	117	
	024263	124	C40	122	
	024266	104	131	045	
	024271	116	000		

```

4831
4832
4833
4834
4835
4836
4837 024274
      024274 105237 003550
4838 024300 000002
4839
4840
4841

```

```

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

```

INCB TRAPD4


```

4885          .SBTTL  DROP UNIT SECTION
4886
4887          ;**
4888          ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
4889          ; TO NO LONGER BE TESTED.  THAT CODE SHALL BE EXECUTED WHEN DODU
4890          ;MACRO IS CALLED WHILE IDU FLAG IS NOT SET BY OPERATOR
4891          ;--
4892
4893 024360      BGNDU
4894 024360      L$DU::
4900 024360      LET R5 := R0 SHIFT 1          ;R5 = LOGICAL DEVICE NUMBER X 2.
         024360 010005      MOV          R0,R5
         024362 006305      ASL          R5
4901 024364      LET DEVTBL(R5) := #NINUSE      ;SET NOT IN USE FLAG FOR THE DEVICE.
         024364 012765 177774 002536      MOV          #NINUSE,DEVTBL(R5)
4902 024372      PRINTF #DROPPM,DROPN          ;PRINT DROP DEVICE MESSAGE
         024372 013746 020210      MOV          DROPN,-(SP)
         024376 012746 004712      MOV          #DROPPM,-(SP)
         024402 012746 000002      MOV          #2,-(SP)
         024406 010600      MOV          SP,R0
         024410 104417      TRAP         C$PNTF
         024412 062706 000006      ADD          #6,SP
4903 024416      EXIT  DU
         024416 000167
         024420 000000
         .WORD  J$JMP
         .WORD  L10016-2-.
4915          .EVEN
4916
4917 024422      ENDDU
         024422      L10016:
         024422 104453      TRAP         C$DU
    
```

```

4920          .SBTTL  ADD UNIT SECTION
4921
4922          ;**
4923          ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
4924          ; TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING.  IF
4925          ; "EF.AUNIT" IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
4926          ;--
4927
4928 024424      BGNAU
4929 024424      L$AU::
4935
4936 024424      LET R5 := R0 SHIFT 1          ;R5 = LOGICAL DEVICE NUMBER X 2.
4937 024424 010005      MOV      R0,R5
4938 024426 006305      ASL      R5
4939 024430      LET DEVTBL(R5) := R0          ;STORE UNIT # IN DEVICE TABLE.
4940 024430 010065 002536      MOV      R0,DEVTBL(R5)
4941 024434      GPHARD R0,R0                  ;GET HARDWARE P TABLE FROM SUPER.
4942 024434 104442      TRAP     C$GPHRD
4943 024436      LET TSSR(R5) := (R0)          ;SAVE TSSR ADDRESS.
4944 024436 011065 002466      MOV      (R0),TSSR(R5)
4945 024442      LET TSDB(R5) := (R0)+ - #2    ;SAVE TSDB ADDRESS.
4946 024442 012065 002456      MOV      (R0)+,TSDB(R5)
4947 024446 162765 000002 002456      SUB      #2,TSDB(R5)
4948 024454      LET TSVCT(R5) := (R0)          ;SAVE INTERRUPT VECTOR ADDRESS.
4949 024454 011065 002476      MOV      (R0),TSVCT(R5)
4950 024460      SETVEC TSVCT(R5),TS4INT(R5),#INTPRI ;SET UP INTERRUPT PROCESSING CONDITIONS.
4951 024460 012746 000340      MOV      #IN      (SP)
4952 024464 016546 002516      MOV      TS4INT  -(SP)
4953 024470 016546 002476      MOV      TSVCT(  -(SP)
4954 024474 012746 000003      MOV      #3,-(SP)
4955 024500 104437      TRAP     C$SVEC
4956 024502 062706 000010      ADD      #10,SP
4957 024506      LET INTFLG(R5) := #0          ;CLEAR INTERRUPT FLAGS.
4958 024506 005065 003506      CLR      INTFLG(R5)
4959
4960          EXIT      AU
4961
4962          .EVEN
4963          ENDAU
4964          L10017:
4965          TRAP     C$AU
4966
4967          ENDMOD

```

```

4966
4977          .TITLE HARDWARE TESTS
4978
4979          .SBTTL TEST 1: BASIC FUNCTIONS.
4980
4981          ;**
4982          ; TEST TO EXECUTE ALL TK25 FUNCTIONS.
4983          ;--
4984
4985 024520          BGNMOD
4986
4987 024520          BGNTST
024520          T1::
4988
4989 024520          LET RANDOM :B= #0          ;CLR THE RANDOM OPERATIONS FLAG.
024520 105037 003533          CLRB          RANDOM
4990 024524          LET EXPBOT :B= #0          ;CLR EXPECT BOT FLAG.
024524 105037 003532          CLRB          EXPBOT
4991
4992 024530          BGNSUB          ;SUBTEST 1 - SET CHAR, DRIVE INIT, GET STATUS.
024530          T1.1:          TRAP          C$BSUB
024530 104402
4993
4994 024532          LET R2 := #BFSEQ0          ;ADR OF CMD SEQ.
024532 012702 025356          MOV          #BFSEQ0,R2
4995 024536          JSR          PC,BFSEQ          ;SET UP CMD SEQ.
024536 004737 025332          JSR          PC,EXALL          ;EXECUTE CMD SEQ ON ALL DEVICES.
4996 024542          JSR          PC,FIRSTU          ;FIND THE FIRST UNIT.
024542 004737 007036          JSR          PC,FIRSTU          ;FIND THE FIRST UNIT.
4997 024546          WHILE DEVTBL(R5) NE #END DO          ;WHILE THERE ARE MORE DEVICES:
4998 024552          50466$:          CMP          DEVTBL(R5),#END
024552 026527 002536 177777          BEQ          50467$
024552 001434          LET R2 := MSGPKA(R5)          ;GET MSG PACKET ADR.
4999 024562          LET R2 := R2 + #12          ;GET XSTAT2 ADR.
024562 016502 002506          MOV          MSGPKA(R5),R2
5000 024566          LET TS4CL(R5) := (R2) CLR.BY #177400 ;STORE CODE LEVEL FROM DTR BYTE.
024566 062702 000012          ADD          #12,R2
5001 024572          IF PASCNT(R5) EQ #1 THEN          ;IF THIS IS PASS 1 THEN:
024572 011265 002526          CMP          PASCNT(R5),#1
024576 042765 177400 002526          BNE          50470$
5002 024604          PRINTF #CODELM,DEVTBL(R5),TS4CL(R5) ;PRINT THE TK25 MICROCODE LEVEL.
024604 026527 003260 000001          MOV          TS4CL(R5),-(SP)
024612 001014          MOV          DEVTBL(R5),-(SP)
5003 024614          MOV          #CODELM, -(SP)
024614 016546 002526          MOV          #3, -(SP)
024620 016546 002536          MOV          SP,R0
024624 012746 004066          TRAP          C$PNTF
024630 012746 000003          ADD          #10,SP
024634 010600
024636 104417
024640 062706 000010
5004 024644          ENDIF
024644
5005 024644          JSR          PC,NEXTU          ;FIND NEXT UNIT.
5006 024650          ENDDO          50470$:
024650 000740          BR          50466$
024652          50467$:
5007 024652          ENDSUB

```

```

024652          L10021:
5008 024652 104403          TRAP      C$ESUB
5009 024654          BGNSUB          ;SUBTEST 2 - REWIND.
024654          T1.2:
024654 104402          TRAP      C$BSUB
5010
5011 024656          LET R2 := #BFSEQ1          ;ADR OF CMD SEQ.
024656 012702 025430          JSR      PC,BFSEQ          MOV      #BFSEQ1,R2
5012 024662 004737 025332          JSR      PC,EXALL          ;SET UP CMD SEQ.
5013 024666 004737 007036          JSR      PC,EXALL          ;EXECUTE CMD SEQ ON ALL DEVICES.
5014 024672          LET STAFLG :B= #0          ;CLEAR START FLAG
024672 105037 003546          CLR      STAFLG
5015 024676          ENDSUB
024676          L10022:
024676 104403          TRAP      C$ESUB
5016
5017 024700          BGNSUB          ;SUBTEST 3 - WRITE/VERIFY.
024700          T1.3:
024700 104402          TRAP      C$BSUB
5018
5019 024702          LET R2 := #BFSEQ2          ;ADR OF CMD SEQ.
024702 012702 025442          JSR      PC,BFSEQ          MOV      #BFSEQ2,R2
5020 024706 004737 025332          JSR      PC,EXALL          ;SET UP CMD SEQ.
5021 024712 004737 007036          JSR      PC,EXALL          ;EXECUTE CMD SEQ ON ALL DEVICES.
5022 024716          ENDSUB
024716          L10023:
024716 104403          TRAP      C$ESUB
5023
5024 024720          BGNSUB          ;SUBTEST 4 - WRITE TAPE MARK, ERASE.
024720          T1.4:
024720 104402          TRAP      C$BSUB
5025
5026 024722          LET R2 := #BFSEQ3          ;ADR OF CMD SEQ.
024722 012702 025534          JSR      PC,BFSEQ          MOV      #BFSEQ3,R2
5027 024726 004737 025332          JSR      PC,EXALL          ;SET UP CMD SEQ.
5028 024732 004737 007036          JSR      PC,EXALL          ;EXECUTE CMD SEQ ON ALL DEVICES.
5029 024736          ENDSUB
024736          L10024:
024736 104403          TRAP      C$ESUB
5030
5031 024740          BGNSUB          ;SUBTEST 5 - SPACE FILES.
024740          T1.5:
024740 104402          TRAP      C$BSUB
5032
5033 024742          LET R2 := #BFSEQ4          ;ADR OF CMD SEQ.
024742 012702 025606          JSR      PC,BFSEQ          MOV      #BFSEQ4,R2
5034 024746 004737 025332          JSR      PC,EXALL          ;SET UP CMD SEQ.
5035 024752 004737 007036          JSR      PC,EXALL          ;EXECUTE CMD SEQ ON ALL DEVICES.
5036 024756          ENDSUB
024756          L10025:
024756 104403          TRAP      C$ESUB
5037
5038 024760          BGNSUB          ;SUBTEST 6 - SPACE RECORDS.
024760          T1.6:
024760 104402          TRAP      C$BSUB
5039

```

```

5040 024762          LET R2 := #BFSEQ5          ;ADR OF CMD SEQ.
      024762 012702 025650          ;SET UP CMD SEQ. MOV #BFSEQ5,R2
5041 024766 004737 025332          JSR PC,BFSEQ
5042 024772 004737 007036          JSR PC,EXALL
5043 024776          ENDSUB          ;EXECUTE CMD SEQ ON ALL DEVICES.
      024776          L10026:
      024776 104403          TRAP C$ESUB
5044
5045 025000          BGNSUB          ;SUBTEST 7 - WRITE RETRY.
      025000          T1.7:
      025000 104402          TRAP C$BSUB
5046
5047 025002          LET R2 := #BFSEQ6          ;ADR OF CMD SEQ.
      025002 012702 025722          ;SET UP CMD SEQ. MOV #BFSEQ6,R2
5048 025006 004737 025332          JSR PC,BFSEQ
5049 025012 004737 007036          JSR PC,EXALL
5050 025016          ENDSUB          ;EXECUTE CMD SEQ ON ALL DEVICES.
      025016          L10027:
      025016 104403          TRAP C$ESUB
5051
5052 025020          BGNSUB          ;SUBTEST 8 - READ REV RETRY.
      025020          T1.8:
      025020 104402          TRAP C$BSUB
5053
5054 025022          LET R2 := #BFSEQ7          ;ADR OF CMD SEQ.
      025022 012702 025754          ;SET UP CMD SEQ. MOV #BFSEQ7,R2
5055 025026 004737 025332          JSR PC,BFSEQ
5056 025032 004737 007036          JSR PC,EXALL
5057 025036          ENDSUB          ;EXECUTE CMD SEQ ON ALL DEVICES.
      025036          L10030:
      025036 104403          TRAP C$ESUB
5058
5059 025040          BGNSUB          ;SUBTEST 9 - READ FWD RETRY.
      025040          T1.9:
      025040 104402          TRAP C$BSUB
5060
5061 025042          LET R2 := #BFSEQ8          ;ADR OF CMD SEQ.
      025042 012702 026006          ;SET UP CMD SEQ. MOV #BFSEQ8,R2
5062 025046 004737 025332          JSR PC,BFSEQ
5063 025052 004737 007036          JSR PC,EXALL
5064 025056          ENDSUB          ;EXECUTE CMD SEQ ON ALL DEVICES.
      025056          L10031:
      025056 104403          TRAP C$ESUB
5065
5066 025060          BGNSUB          ;SUBTEST 10- CLEAN.
      025060          T1.10:
      025060 104402          TRAP C$BSUB
5067
5068 025062          LET R2 := #BFSEQ9          ;ADR OF CMD SEQ.
      025062 012702 026040          ;SET UP CMD SEQ. MOV #BFSEQ9,R2
5069 025066 004737 025332          JSR PC,BFSEQ
5070 025072 004737 007036          JSR PC,EXALL
5071 025076          ENDSUB          ;EXECUTE CMD SEQ ON ALL DEVICES.
      025076          L10032:
      025076 104403          TRAP C$ESUB
5072
5073 025100          BGNSUB          ;SUBTEST 11 - WTV SWAPPED DATA BYTES.

```

```

025100          T1.11:
025100 104402          TRAP      C$BSUB
5074
5075 025102          LET R2 := #BFSE10          ;ADR OF CMD SEQ.
025102 012702 026062          JSR      PC,BFSEQ          ;SET UP CMD SEQ.
5076 025106 004737 025332          JSR      PC,EXALL          ;WRITE/VERIFY RECORDS 1 AND 2.
5077 025112 004737 007036          LET SWBFLG :B= #1          ;ENABLE BYTE SWAPPING.
5078 025116          LET SWBFLG :B= #1          ;WRITE/VERIFY RECORDS 3 AND 4.
025116 112737 000001 003536          JSR      PC,EXALL          ;DISABLE BYTE SWAPPING.
5079 025124 004737 007036          LET SWBFLG :B= #0
5080 025130          ENDSUB
025130 105037 003536
5081 025134          L10033:
025134          TRAP      C$ESUB
025134 104403
5082
5083 025136          LET R2 := DATAWT + #10.          ;INIT WRITE BUFFER POINTER.
025136 013702 003414          MOV      DATAWT,R2
025142 062702 000012          ADD      #10.,R2
5084 025146          WHILE R2 NE DATAWT DO          ;UNTIL 10 BYTES HAVE BEEN SWAPPED.
025146          50471$:
025146 020237 003414          CMP      R2,DATAWT
025152 001402          BEQ      50472$
5085 025154 000342          SWAB -(R2)          ;SWAP DATA BYTES IN WRITE BUFFER.
5086 025156          ENDDO
025156 000773          BR      50471$
025160          50472$:
5087 025160          LET T1SWB :B= T1SWB + #1          ;SET T1 SWAP BYTES FLAG FOR "CKDATA" SUBR
025160 105237 003542          INCB   T1SWB
5088
5089 025164          BGNSUB          ;SUBTEST 2 - READ SWAPPED DATA BYTES.
025164          T1.12:
025164 104402          TRAP      C$BSUB
5090
5091 025166          LET CMDWRD := #RDR          ;CMD IS READ REV.
025166 012737 104401 003426          JSR      PC,VFEXC          ;VERIFY ODD LENGTH SWAP (RECORD 4).
5092 025174 004737 016642          LET CNDPKT+CP.CNT := #12          ;CHANGE BYTE COUNT TO 10.
5093 025200          MOV      #12,CNDPKT+CP.CNT
025200 012737 000012 002322          JSR      PC,VFEXC          ;VERIFY EVEN LENGTH SWAP (RECORD 3).
5094 025206 004737 016642          LET SWBFLG :B= #1          ;ENABLE BYTE SWAPPING.
5095 025212          MOV      #1,SWBFLG
025212 112737 000001 003536          LET CNDPKT+CP.CNT := #11          ;CHANGE BYTE COUNT TO 9.
5096 025220          MOV      #11,CNDPKT+CP.CNT
025220 012737 000011 002322          JSR      PC,VFEXC          ;VERIFY ODD LENGTH SWAP (RECORD 2).
5097 025226 004737 016642          LET CNDPKT+CP.CNT := #12          ;CHANGE BYTE COUNT TO 10.
5098 025232          MOV      #12,CNDPKT+CP.CNT
025232 012737 000012 002322          JSR      PC,VFEXC          ;VERIFY EVEN LENGTH SWAP (RECORD 1).
5099 025240 004737 016642          LET CMDWRD := #RDF          ;CMD IS READ FWD.
5100 025244          MOV      #RDF,CMDWRD
025244 012737 104001 003426          JSR      PC,VFEXC          ;VERIFY EVEN LENGTH SWAP (RECORD 1).
5101 025252 004737 016642          LET CNDPKT+CP.CNT := #11          ;CHANGE BYTE COUNT TO 9.
5102 025256          MOV      #11,CNDPKT+CP.CNT
025256 012737 000011 002322          JSR      PC,VFEXC          ;VERIFY ODD LENGTH SWAP (RECORD 2).
5103 025264 004737 016642          LET SWBFLG :B= #0          ;DISABLE BYTE SWAPPING.
5104 025270          CLR      SWBFLG
025270 105037 003536          LET CNDPKT+CP.CNT := #12          ;CHANGE BYTE COUNT TO 10.
5105 025274

```

5106	025274	012737	000012	002322					
5107	025302	004737	016642		JSR	PC,VFEXC			MOV #12,CMDPKT+CP.CNT
	025306				LET	CMDPKT+CP.CNT := #11			;VERIFY EVEN LENGTH SWAP (RECORD 3).
	025306	012737	000011	002322					;CHANGE BYTE COUNT TO 9.
5108	025314	004737	016642		JSR	PC,VFEXC			MOV #11,CMDPKT+CP.CNT
5109									;VERIFY ODD LENGTH SWAP (RECORD 4).
5110	025320				ENDSUB				
	025320				L10034:				
	025320	104403							TRAP C\$ESUB
5111									
5112	025322				LET	T1SWB :B= #0			;CLEAR T1 SWAP BYTES FLAG
	025322	105037	003542						CLR T1SWB
5113									
5114									
5115	025326				EXIT	TST			TRAP C\$EXIT
	025326	104432							.WORD L10020-
	025330	000554							

```

5117 ; SUBROUTINE TO MOVE A COMMAND SEQUENCE TO THE SEQUENCE TABLE.
5118 ; INPUTS: R2 = FWA OF COMMAND SEQUENCE.
5119 ; OUTPUTS:
5120 ; REGISTERS:
5121 ; CALLS:
5122
5123 025332 BFSEQ: LET R1 := #CMDSEQ ;INIT SEQ TABLE ADDRESS.
      025332 012701 003554 MOV #CMDSEQ,R1
5124 025336 WHILE (R2) NE #END DO ;WHILE THERE ARE MORE COMMANDS:
      025336 CMP (R2),#END
      025336 021227 177777 BEQ 50473$
      025342 001402 ;MOVE COMMANDS TO SEQ TABLE.
5125 025344 LET (R1)+ := (R2)+ MOV (R2)+,(R1)+
      025344 012221 ENDDO
5126 025346 BR 50473$
      025346 000773 50474$:
      025350 LET (R1) := #END ;STORE END OF SEQUENCE CODE.
5127 025350 012711 177777 MOV #END,(R1)
      025350 000207 RTS PC ;RETURN.
5128
5129
5130 ; BASIC FUNCTION COMMAND SEQUENCE
5131
5132
5133 BFSEQ0: .WORD SCH ;SET CHAR. 200. (1)
5134 025356 140004 200
5135 025360 000200 1
5136 025362 000001 0
5137 025364 000000 0
5138 025366 100013 DRI ;DRIVE INIT. (2)
5139 025370 000001 1
5140 025372 000001 1
5141 025374 000000 0
5142 025376 140004 SCH ;SET CHAR. 20 (3)
5143 025400 000020 20
5144 025402 000001 1
5145 025404 000000 0
5146 025406 100017 GES ;GET STATUS. (4)
5147 025410 000001 1
5148 025412 000001 1
5149 025414 000000 0
5150 025416 140004 SCH ;SET CHAR. 40. (5)
5151 025420 000040 40
5152 025422 000001 1
5153 025424 000000 0
5154 025426 177777 .WORD END
5155
5156 025430 102010 BFSEQ1: RWD ;REWIND TWICE. (6)
5157 025432 000001 1
5158 025434 000002 2
5159 025436 000000 0
5160 025440 177777 .WORD END
5161
5162 025442 104105 BFSEQ2: WTV ;WRITE/VERIFY PAT 1. (7)
5163 025444 010000 DATCNT
5164 025446 000001 1
5165 025450 000001 1

```


5166	025452	104105		WTV		;WTV PAT 2.	(8)
5167	025454	010000		DATCNT			
5168	025456	000001		1			
5169	025460	000002		2			
5170	025462	104105		WTV		;WTV PAT 3.	(9)
5171	025464	010000		DATCNT			
5172	025466	000001		1			
5173	025470	000003		3			
5174	025472	104105		WTV		;WTV PAT 4.	(10)
5175	025474	010000		DATCNT			
5176	025476	000001		1			
5177	025500	000004		4			
5178	025502	104105		WTV		;WTV PAT 5.	(11)
5179	025504	010000		DATCNT			
5180	025506	000001		1			
5181	025510	000005		5			
5182	025512	104105		WTV		;WTV PAT 6.	(12)
5183	025514	010000		DATCNT			
5184	025516	000001		1			
5185	025520	000006		6			
5186	025522	104105		WTV		;WTV PAT 0.	(13)
5187	025524	010000		DATCNT			
5188	025526	000001		1			
5189	025530	000000		0			
5190	025532	177777	.WORD	END			
5191							
5192	025534	100011	BFSEQ3:	WTM		;WRITE TAPE MARK.	(14)
5193	025536	000001		1			
5194	025540	000001		1			
5195	025542	000000		0			
5196	025544	104005		WRT		;WRITE 10 RECORDS.	(15)
5197	025546	010000		DATCNT			
5198	025550	000010		10			
5199	025552	000001		1			
5200	025554	100411		ERS		;ERASE 10 TIMES.	(16)
5201	025556	000001		1			
5202	025560	000010		10			
5203	025562	000000		0			
5204	025564	100011		WTM		;WRITE TAPE MARK.	(17)
5205	025566	000001		1			
5206	025570	000001		1			
5207	025572	000000		0			
5208	025574	101011		WTR		;WTR RETRY	(18)
5209	025576	000001		1			
5210	025600	000001		1			
5211	025602	000000		0			
5212	025604	177777	.WORD	END			
5213							
5214	025606	105410	BFSEQ4:	SFR		;SPACE 2 FILES REV.	(19)
5215	025610	000002		2			
5216	025612	000001		1			
5217	025614	000000		0			
5218	025616	105010		SFF		;SPACE 2 FILES FWD.	(20)
5219	025620	000002		2			
5220	025622	000001		1			
5221	025624	000000		0			
5222	025626	105410		SFR		;SPACE 2 FILES REV.	(21)

5223	025630	000001		1		
5224	025632	000002		2		
5225	025634	000000		0		
5226	025636	105010		SFF	;SPACE 2 FILES FWD.	(22)
5227	025640	000001		1		
5228	025642	000002		2		
5229	025644	000000		0		
5230	025646	177777	.WORD	END		
5231						
5232	025650	102010	BFSEQ5:	RWD	;REWIND.	(23)
5233	025652	000001		1		
5234	025654	000001		1		
5235	025656	000000		0		
5236	025660	104010		SRF	;SPACE 7 RECORDS FWD.	(24)
5237	025662	000007		7		
5238	025664	000001		1		
5239	025666	000000		0		
5240	025670	104410		SRR	;SPACE 7 RECORDS REV.	(25)
5241	025672	000007		7		
5242	025674	000001		1		
5243	025676	000000		0		
5244	025700	104010		SRF	;SPACE 7 RECORDS FWD.	(26)
5245	025702	000001		1		
5246	025704	000007		7		
5247	025706	000000		0		
5248	025710	104410		SRR	;SPACE 7 RECORDS REV.	(27)
5249	025712	000001		1		
5250	025714	000007		7		
5251	025716	000000		0		
5252	025720	177777	.WORD	END		
5253						
5254	025722	102010	BFSEQ6:	RWD	;REWIND.	(28)
5255	025724	000001		1		
5256	025726	000001		1		
5257	025730	000000		0		
5258	025732	104005		WRT	;WRITE.	(29)
5259	025734	010000		DATCNT		
5260	025736	000001		1		
5261	025740	000001		1		
5262	025742	105005		WRR	;WRITE RETRY.	(30)
5263	025744	010000		DATCNT		
5264	025746	000001		1		
5265	025750	000001		1		
5266	025752	177777	.WORD	END		
5267						
5268	025754	104401	BFSEQ7:	RDR	;READ REV.	(31)
5269	025756	010000		DATCNT		
5270	025760	000001		1		
5271	025762	000001		1		
5272	025764	105401		RNR	;READ NEXT REV.	(32)
5273	025766	010000		DATCNT		
5274	025770	000001		1		
5275	025772	000001		1		
5276	025774	125401		RNF	;READ NEXT FWD.	(33)
5277	025776	010000		DATCNT		
5278	026000	000001		1		
5279	026002	000001		1		

```

5280 026004 177777          .WORD  END
5281
5282 026006 104001          BFSEQ8:  RDF          ;READ FWD.          (34)
5283 026010 010000          DATCNT
5284 026012 000001          1
5285 026014 000001          1
5286 026016 105001          RPF          ;READ PREVIOUS FWD. (35)
5287 026020 010000          DATCNT
5288 026022 000001          1
5289 026024 000001          1
5290 026026 125001          RPR          ;READ PREVIOUS REV. (36)
5291 026030 010000          DATCNT
5292 026032 000001          1
5293 026034 000001          1
5294 026036 177777          .WORD  END
5295
5296 026040 101012          BFSEQ9: .WORD  CLN          ;CLEAN.            (37)
5297 026042 000001          1
5298 026044 000001          1
5299 026046 000000          0
5300 026050 102010          RWD          ;REWIND             (38)
5301 026052 000001          1
5302 026054 000001          1
5303 026056 000000          0
5304 026060 177777          .WORD  END          ;END OF SEQUENCE.
5305
5306 026062 104105          BFSE10: WTV          ;WRITE/VERIFY EVEN LENGTH. (39)
5307 026064 000012          12
5308 026066 000001          1
5309 026070 000000          0
5310 026072 104105          WTV          ;WRITE/VERIFY ODD LENGTH. (40)
5311 026074 000011          11
5312 026076 000001          1
5313 026100 000000          0
5314 026102 177777          .WORD  END
5315          .EVEN
5316
5317 026104          ENDTST
      026104          L10020:
      026104 104401          TRAP  C$ETST

```

```

5319          .SBTTL TEST 2: DATA RELIABILITY.
5320
5321          ;**
5322          ; TEST TO CHECK THE DATA RELIABILITY OF THE TK25.
5323          ;--
5324 026106    BGNTST
          T2::
5325
5326 026106    LET RANDOM :B= #1          ;SET THE RANDOM OPERATIONS FLAG.
          026106 112737 000001 003533    MOVB #1,RANDOM
5327 026114    LET EXPBOT :B= #0        ;CLEAR EXPECT BOT FLAG.
          026114 105037 003532    CLRB EXPBOT
5328 026120    LET R2 := #DATCNT - #1    ;SET UP THE RECORD LENGTH MASK.
          026120 012702 010000    MOV #DATCNT,R2
          026124 005302    DEC R2
5329 026126    LET LENMSK := COMP R2    ;ALLOW MAXIMUM BUFFER.
          026126 010237 003436    MOV R2,LENMSK
          026132 005137 003436    COM LENMSK
5330 026136    JSR PC,SETCH              ;CMD 1 = SET CHARACTERISTIC.
5331 026142    IFB STAFLG NE #0 THEN    ;IF STARTING THEN:
          026142 105737 003546    TSTB STAFLG
          026146 001404    BEQ 50475$
5332 026150    JSR PC,SETRW              ;CMD2=REWIND
5333 026154    LET STAFLG :B= #0        ;CLR START FLAG.
          026154 105037 003546    CLRB STAFLG
5334 026160    ENDIF
          026160
5335 026160    LET (R1)+ := #WRT        ;CMD3 = WRITE.
          026160 012721 104005    MOV #WRT,(R1)+
5336 026164    LET (R1)+ := #DATCNT    ;SET BRF TO MAX FOR PATTERN GENERATION.
          026164 012721 010000    MOV #DATCNT,(R1)+
5337 026170    LET R2 := COMP #RNOPSC
          026170 012702 177740    MOV #RNOPSC,R2
          026174 005102    COM R2
5338 026176    LET (R1)+ := R2        ;31 OPERATIONS.
          026176 010221    MOV R2,(R1)+
5339 026200    LET (R1)+ := #RANP      ;RANDOM PATTERN.
          026200 012721 000007    MOV #RANP,(R1)+
5340 026204    REPEAT
          026204
5341 026204    WHILE R1 LT #SEQEND DO  ;FILL SEQ TBL WITH RANDOM CMDS.
          026204
          026204 020127 003644    50476$:
          026210 002012    50477$:
          026210    LET RANS := RANS + RANB
          026210 063737 003440 003442    CMP R1,#SEQEND
          026212    LET R2 := RANS CLR.BY #177741 ;R2 = RANDOM # (0 - 36).
          026212 013702 003442    BGE 50500$
          026224 042702 177741    ADD RANB,RANS
          026224 004772 026366    MOV RANS,R2
          026234    JSR PC,@RANCMD(R2)    ;SET UP A RANDOM CMD + BRF.
          026234 000763    BIC #177741,R2
          026236    ENDDO
          026236
          026236    BR 50477$
          026236    LET (R1) := #END      ;STORE END OF SEQUENCE CODE IN TABLE.
          026236 012711 177777    MOV #END,(R1)
5347 026242    JSR PC,EXALL            ;GO EXECUTE ALL CMDS IN SEQUENCE TABLE.
          026242 004737 007036
5348 026246    LET R1 := #CMDSEQ      ;INIT CMD SEQ TBL POINTER.

```

5349	026246	012701	003554						
	026252			UNTIL R2 NE #0				MOV #CMDSEQ,R1	
	026252	005702						;REPEAT UNTIL EOT IS REACHED	
	026254	001753						TST R2	
5350	026256			LET ALLEOT :B= ALLEOT + #1				BEQ 504764	
	026256	105237	003543					INCB ALLEOT	
5351	026262	000240		NOP					
5352	026264	000240		NOP					
5353	026266	000240		NOP					
5354	026270	004737	030364	JSR PC,TSWEOT					
5355									
5356									
5357									
5358	026274	004737	026426	JSR PC,RANRD					
5359	026300			LET CMDSEQ+4 := COMP #RNOPSC					
	026300	012737	177740						
	026306	005137	003560						
5360	026312			LET CMDSEQ+14 := CMDSEQ+4					
	026312	013737	003560						
5361	026320			LET (R1) := #END					
	026320	012711	177777						
5362	026324	004737	007036	JSR PC,EXALL					
5363	026330			LET ALLEOT :B= #0					
	026330	105037	003543						
5364	026334			LET RPTFLG :B= #1					
	026334	112737	000001						
5365	026342			LET R1 := #CMDSEQ					
	026342	012701	003554						
5366	026346	004737	007016	JSR PC,SETRW					
5367	026352			LET (R1) := #END					
	026352	012711	177777						
5368	026356	004737	007036	JSR PC,EXALL					
5369									
5370	026362			EXIT TST					
	026362	104432							
	026364	000174							
5371									

```

TRAP C$EXIT
.WORD L10035-.

```

```

5373      :      ADDRESSES OF SUBROUTINES USED TO SET UP RANDOM OPERATIONS IN
5374      :      THE DATA RELIABILITY TEST.
5375
5376 026366 026502      RANCMD: RANWR      ;WRITE.
5377 026370 026502      RANWR      ;WRITE.
5378 026372 026502      RANWR      ;WRITE.
5379 026374 026502      RANWR      ;WRITE.
5380 026376 026502      RANWR      ;WRITE.
5381 026400 026502      RANWR      ;WRITE.
5382 026402 026502      RANWR      ;WRITE.
5383 026404 026502      RANWR      ;WRITE.
5384 026406 026426      RANRD      ;READ.
5385 026410 026426      RANRD      ;READ.
5386 026412 026426      RANRD      ;READ.
5387 026414 026426      RANRD      ;READ.
5388 026416 026426      RANRD      ;READ.
5389 026420 026426      RANRD      ;READ.
5390 026422 026426      RANRD      ;READ.
5391 026424 026426      RANRD      ;READ.
5392
5393
5394
5395
5396
5397      :      SUBROUTINE TO SET UP READ COMMANDS IN SEQUENCE TABLE.
5398      :      INPUTS:
5399      :      OUTPUTS:
5400      :      REGISTERS:      R2
5401      :      CALLS:
5402
5403 026426      RANRD: LET (R1)+ := #SRR      ;STORE SPACE RECORD REVERSE CMD
      026426 012721 104410      MOV      #SRR,(R1)+
5404 026432      LET RANB := RANB + RANS      MOV      RANS,RANB
      026432 063737 003442 003440      ADD      RANS,RANB
5405 026440      LET R2 := RANB CLR.BY #RNOPSC      MOV      RANB,R2
      026440 013702 003440      BIC      #RNOPSC,R2
      026444 042702 177740
5406 026450      LET (R1)+ := R2      ;SET REPOSITION COUNT
      026450 010221      MOV      R2,(R1)+
5407 026452      LET (R1)+ := #1      ;DO ONCE
      026452 012721 000001      MOV      #1,(R1)+
5408 026456      LET (R1)+ := #RANP      ;RANDOM PATTERN.
      026456 012721 000007      MOV      #RANP,(R1)+
5409 026462      LET (R1)+ := #RDF      ;STORE READ FWD CMD.
      026462 012721 104001      MOV      #RDF,(R1)+
5410 026466      LET (R1)+ := #DATCNT      ;SET BRF TO MAX TO READ RANDOM LENGTHS.
      026466 012721 010000      MOV      #DATCNT,(R1)+
5411 026472      LET (R1)+ := R2      ;SET RANDOM # OF OPERATIONS.
      026472 010221      MOV      R2,(R1)+
5412 026474      LET (R1)+ := #RANP      ;RANDOM PATTERN.
      026474 012721 000007      MOV      #RANP,(R1)+
5413 026500      RTS PC

```

```

5415 ; SUBROUTINE TO SET UP A WRITE COMMAND IN THE SEQUENCE TABLE.
5416 ; INPUTS:
5417 ; OUTPUTS:
5418 ; REGISTERS:
5419 ; CALLS:
5420
5421 026502 RANWR: LET (R1)+ := #WRT ;STORE WRITE CMD.
      026502 012721 104005 ;MOV #WRT,(R1)+
5422 026506 004737 026526 JSR PC,RANW ;STORE BRF, # OF OPERATIONS, PATTERN.
5423 026512 000207 RTS PC
5424
5425
5426
5427
5428
5429 ; SUBROUTINE TO SET UP A WRITE/VERIFY COMMAND IN THE SEQUENCE TABLE.
5430 ; INPUTS:
5431 ; OUTPUTS:
5432 ; REGISTERS:
5433 ; CALLS:
5434
5435 026514 RANWV: LET (R1)+ := #WTV ;STORE WRITE/VERIFY CMD.
      026514 012721 104105 ;MOV #WTV,(R1)+
5436 026520 004737 026526 JSR PC,RANW ;STORE BRF, # OF OPERATIONS, PATTERN.
5437 026524 000207 RTS PC
5438
5439
5440
5441
5442
5443 ; SUBROUTINE TO STORE BRF, # OF OPERATIONS, PATTERN IN COMMAND
5444 ; SEQUENCE TABLE FOR WRITE AND WRITE/VERIFY COMMANDS.
5445 ; INPUTS:
5446 ; OUTPUTS:
5447 ; REGISTERS: R2
5448 ; CALLS:
5449
5450 026526 RANW: LET (R1)+ := #DATCNT ;SET BRF TO MAX FOR PATTERN GENERATION.
      026526 012721 010000 ;MOV #DATCNT,(R1)+
5451 ;RANDOM BRF WILL BE GENERATED FOR EACH RECORD.
5452 026532 LET RANB := RANB + RANS ;RANDOM BRF WILL BE GENERATED FOR EACH RECORD.
      026532 063737 003442 003440 ADD RANS,RANB
5453 026540 LET R2 := RANB CLR.BY #RNOPSC ;RANDOM BRF WILL BE GENERATED FOR EACH RECORD.
      026540 013702 003440 MOV RANB,R2
      026544 042702 177740 BIC #RNOPSC,R2
5454 026550 LET (R1)+ := R2 ;SET RANDOM # OF OPERATIONS.
      026550 010221 MOV R2,(R1)+
5455 026552 LET (R1)+ := #RANP ;RANDOM PATTERN.
      026552 012721 000007 MOV #RANP,(R1)+
5456 026556 000207 RTS PC ;RETURN.
5457
5458 .EVEN
5459
5460 026560 L10035:
      026560 104401 TRAP C$ETST
5461

```

```

5463      .SBTTL TEST 3: WRITE AND READ STREAMING TEST.
5464
5465      ;**
5466      ;
5467      ; THIS TEST STREAM WRITES 9000 RECORDS OF 4096 BYTES EACH.
5468      ; DATA IS THEN VERIFIED BY PERFORMING A REWIND OPERATION, FOLLOWED
5469      ; BY A READ FORWARD OPERATON FOR THE 9000 RECORDS.
5470      ;
5471      ;--
5472
5473      026562      BGNTST
5474      026562      T3::
5475      026562      IFB HERE EQ #0 THEN
5476      026562      105737      003340      TSTB      HERE
5476      026566      001012      BNE      50501$
5476      026570      PRINTF #TAPCAP
5476      026570      012746      022157      MOV      #TAPCAP, -(SP)
5476      026574      012746      000001      MOV      #1, -(SP)
5476      026600      010600      MOV      SP, R0
5476      026602      104417      TRAP     C$PNTF
5476      026604      062706      000004      ADD      #4, SP
5477      026610      LET HERE :B= HERE + #1
5477      026610      105237      003340      INCB     HERE
5478      026614      ENDF
5479      026614      LET RANDOM :B= #0      ;CLEAR THE RANDOM OPERATIONS FLAG.
5479      026614      105037      003533      CLRB    RANDOM
5480      026620      LET EXPBOT :B= #0      ;CLEAR THE EXPECT BOT FLAG.
5480      026620      105037      003532      CLRB    EXPBOT
5481      026624      004737      006772      JSR PC, SETCH      ;SET CHARACTERISTICS.
5482      026630      004737      007016      JSR PC, SETRW      ;SET REWIND COMMAND IN BUFFER.
5483      026634      LET STAF LG :B= #0      ;CLEAR THE START FLAG.
5483      026634      105037      003546      CLRB    STAF LG
5484      026640      LET (R1) := #END      ;PLACE END FLAG IN SEQUENCE TABLE.
5484      026640      012711      177777      MOV     #END, (R1)
5485      026644      004737      007036      JSR PC, EXALL      ;REWIND ALL UNITS.
5486      026650      004737      017666      JSR PC, FIRSTU     ;FIND THE FIRST UNIT TO TEST (UUT)
5487
5488      ; *****
5489      ;WRITE AND READ EACH UNIT IN TURN BEFORE GOING ON TO THE NEXT.
5490
5491      ; *****
5492
5493
5494      026654      WHILE DEVTBL(R5) NE #END DO      ;WHILE THERE ARE MORE DEVICES:
5494      026654      026527      002536      177777      50502$:
5494      026662      001550      CMP     DEVTBL(R5), #END
5495
5496      026664      LET BTPT := BTADDR(R5)      ;CLEAR BAD SPOT COUNTER
5496      026664      016537      002550      003526      MOV     BTADDR(R5), BTPT
5497      026672      LET @BTPT := #0      ;START FROM BOT
5497      026672      005077      154630      CLR     @BTPT
5498      026676      LET STREAM :B= #255.      ;SET FLAG - WE'RE GOING TO STREAM
5498      026676      112737      000377      003544      MOVB   #255., STREAM
5499      026704      LET R1 := #CMDSEQ      ;SETUP SEQUENCE TABLE ADDRESS
    
```


5500	026704	012701	003554	LET (R1)+ := #WRT	;WRITE COMMAND	MOV	#CMDSEQ,R1
5501	026710	012721	104005	LET (R1)+ := #DATCNT	;4096-BYTE RECORD LENGTH.	MOV	#WRT,(R1)+
5502	026714	012721	010000	LET (R1)+ := #9000.	;WRITE 9000 RECORDS.	MOV	#DATCNT,(R1)+
5503	026720	012721	021450	LET (R1)+ := #5	;GENERATE AND WRITE PATTERN 5.	MOV	#9000.,(R1)+
5504	026724	012721	000005	LET (R1) := #END	;SET END OF SEQUENCE TABLE.	MOV	#5,(R1)+
5505	026730	012711	177777	LET R1 := #CMDSEQ	;SEQ. TABLE ADDRESS FOR SUBR. 'SETUP'.	MOV	#END,(R1)
5506	026734	012701	003554	JSR PC, SETUP	;SETUP THE COMMAND TABLE	MOV	#CMDSEQ,R1
5507	026740	004737	010000	LET R5SAVE := R5	;SAVE R5, EH?	MOV	R5,R5SAVE
5508	026744	010537	003460				
5509	026750			WHILE NCNT LT NCNT1 DO	;WHILE MORE RECORDS SHOULD BE WRITTEN:		
	026750						50504\$:
	026750	023737	003420			CMP	NCNT,NCNT1
	026756	002022	003422			BGE	50505\$
5510							
5511	026760	004737	007672	JSR PC, CMDAC	;SAVE ASCII COMMAND FOR ERROR MESSAGE		
5512	026764	004737	010774	JSR PC, EXECUTE	;ISSUE COMMAND TO UNIT.		
5513	026770	004737	011470	JSR PC, GOWAIT	;GO WAIT FOR DONE TO SET		
5514	026774			IF DEVTBL(R5) EQ #NINUSE THEN			
	026774	026527	002536			CMP	DEVTBL(R5),#NINUSE
	027002	001005	177774			BNE	50506\$
5515	027004			LET NCNT := NCNT1 - #1			
	027004	013737	003422			MOV	NCNT1,NCNT
	027012	005337	003420			DEC	NCNT
5516	027016			ENDIF			
	027016						50506\$:
5517	027016			LET NCNT := NCNT + #1	;UPDATE THE RECORD COUNT	INC	NCNT
5518	027022	005237	003420	ENDDO	;END OF RECORD 'DO' LOOP	BR	50504\$
	027022	000752					50505\$:
5519	027024			LET R1 := #CMDSEQ	;RESET R1 TO TOP OF TABLE	MOV	#CMDSEQ,R1
	027024	012701	003554				
5520	027030	004737	007016	JSR PC, SETRW	;ISSUE REWIND		
5521	027034			LET (R1) := #END	;PLACE END FLAG IN SEQUENCE TABLE	MOV	#END,(R1)
5522	027040	012711	177777	LET -(SP) := L\$LUN	;SAVE THE CURRENT LUN	MOV	L\$LUN,-(SP)
	027040	013746	002074				
5523	027044	004737	007036	JSR PC, EXALL	;DO REWIND, NOW		
5524	027050			LET R5 := R5SAVE	;RESTORE R5	MOV	R5SAVE,R5
	027050	013705	003460				
5525	027054			LET L\$LUN := (SP)+	;RESTORE THE CURRENT LUN	MOV	(SP)+,L\$LUN
	027054	012637	002074				
5526	027060			IF DEVTBL(R5) NE #NINUSE THEN			
	027060	026527	002536			CMP	DEVTBL(R5),#NINUSE
	027066	001431	177774			BEQ	50507\$
5527	027070			LET R1 := #CMDSEQ	;TOP OF COMMAND TABLE	MOV	#CMDSEQ,R1
	027070	012701	003554				
5528	027074			LET (R1)+ := #RDF	;READ FORWARD COMMAND	MOV	#RDF,(R1)+
	027074	012721	104001				

5529	027100			LET (R1)+ := #DATCNT	;4096 BYTE RECORDS
	027100	012721	010000		MOV #DATCNT,(R1)+
5530	027104			LET (R1)+ := #9000.	;9000 ITERATIONS
	027104	012721	021450		MOV #9000,(R1)+
5531	027110			LET (R1)+ := #5	;READ PATTERN NUMBER 5
	027110	012721	000005		MOV #5,(R1)+
5532	027114			LET (R1)+ := #END	;TABLE TERMINATOR
	027114	012721	177777		MOV #END,(R1)+
5533	027120			LET R1 := #CMDSEQ	;TOP OF TABLE, AGAIN!
	027120	012701	003554		MOV #CMDSEQ,R1
5534	027124	004737	010000	JSR PC, SETUP	;SET UP THE COMMAND TABLE
5535	027130			LET VFYFLG :B= #1	;ALLOW THE DATA VERIFY
	027130	112737	000001 003534		MOVB #1,VFYFLG
5536	027136			LET R5SAVE := R5	;SAVE R5
	027136	010537	003460		MOV R5,R5SAVE
5537	027142	004737	016530	JSR PC, VFYDAT	;GO OFF AND CHECK READ OPERATIONS
5538	027146			LET R5 := R5SAVE	;RESTORE R5
	027146	013705	003460		MOV R5SAVE,R5
5539	027152			ENDIF	
	027152				50507\$:
5540	027152	005037	003420	LET NCNT := #0	;CLEAR RECORD COUNT
	027152				CLR NCNT
5541	027156			LET R3 := #ENDERF	;END OF BUFFER ADDRESS
	027156	012703	003506		MOV #ENDERF,R3
5542	027162	004737	012110	JSR PC, CLRERR	;RESET THE ERROR BUFFER
5543	027166			LET VFYFLG :B= #0	;CLEAR VERIFY FLAG
	027166	105037	003534		CLRB VFYFLG
5544	027172			LET EXPBOT :B= #0	;CLEAR EXPECT BOT FLAG.
	027172	105037	003532		CLRB EXPBOT
5545	027176	004737	017734	JSR PC, NEXTU	;GET NEXT UNIT TO TEST (UUT).
5546					
5547	027202			ENDDO	;END OF UUT LOOP
	027202	000624			BR 50502\$
	027204				50503\$:
5548					
5549	027204			LET STREAM :B= #0	;CLEAR STREAMING FLAG FOR OTHER TESTS.
	027204	105037	003544		CLRB STREAM
5550	027210			LET ALLEOT :B= #0	;RESET THE UNITS @ EOT STATUS
	027210	105037	003543		CLRB ALLEOT
5551	027214			LET RPTFLG :B= #1	;REQUEST A REPORT
	027214	112737	000001 003535		MOVB #1,RPTFLG
5552	027222			LET R1 := #CMDSEQ	;TOP OF TABLE
	027222	012701	003554		MOV #CMDSEQ,R1
5553	027226	004737	007016	JSR PC, SETRW	;STORE THE REWIND COMMAND
5554	027232			LET (R1) := #END	;TERMINATOR
	027232	012711	177777		MOV #END,(R1)
5555	027236	004737	007036	JSR PC, EXALL	;REWIND AND REPORT STATUS FOR ALL UNITS
5556					
5557	027242			EXIT TST	;EXIT TEST
	027242	104432			TRAP C\$EXIT
	027244	000002			.WORD L10036-
5558				.EVEN	
5559					
5560	027246			ENDTST	;JUST IN CASE.
	027246				
	027246	104401		L10036:	TRAP C\$ETST

5637
5638 027524
027524
027524 104401

ENDTST
L10040:

TRAP C#ETST

```

5640
5641          .SBTTL TEST 6: EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.
5642
5643          ;++
5644          ; TEST TO EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.
5645          ;--
5646
5647 027526          BGNTST
5648          T6::
5649 027526          LET RANDOM :B= #0          ;CLEAR RANDOM MODE FLAG.
5650 027532          LET EXPBOT :B= #1          ;SET EXPECT BOT FLAG.
5651 027540          LET IRE :B= PIRE          ;MOVE INHIBIT RFC ERROR REPORT FLAG.
5652 027546          JSR PC,SETCH          ;CMD 1 = SET CHARACTERISTIC.
5653 027552          LET CMDSEQ+2 := CHAR          ;MOVE CHAR CODE FROM P TBL TO SEQ TBL.
5654 027560          LET R2 := #CMDD          ;R2 POINTS TO CMD2 IN SOFT P TABLE.
5655 027564          JSR PC,PTCMDS          ;MOVE CMD 2 FROM P TBL TO SEQ TBL.
5656 027570          JSR PC,PTCMDS          ;MOVE CMD 3 FROM P TBL TO SEQ TBL.
5657 027574          JSR PC,PTCMDS          ;MOVE CMD 4 FROM P TBL TO SEQ TBL.
5658 027600          JSR PC,PTCMDS          ;MOVE CMD 5 FROM P TBL TO SEQ TBL.
5659 027604          JSR PC,PTCMDS          ;MOVE CMD 6 FROM P TBL TO SEQ TBL.
5660 027610          JSR PC,PTCMDS          ;MOVE CMD 7 FROM P TBL TO SEQ TBL.
5661 027614          JSR PC,PTCMDS          ;MOVE END CMD FROM P TBL TO SEQ TBL.
5662 027620          LET JLOOP := #0          ;CLEAR JMP CMD LOOP COUNT.
5663 027624          LET STAFLG :B= #0          ;CLEAR START FLAG
5664 027630          LET R1 := #CMDSEQ          ;INIT SEQUENCE TABLE POINTER.
5665          000000          $BRJMP=0          ;ENABLE JMP SUBSTITUTION FOR BR, IF NECESSARY.
5666
5667
5668 027634          3$: WHILE (R1) NE #END DO          ;WHILE THERE ARE CMDS LEFT IN SEQUENCE TBL:
5669 027646          CMP #JMP.C,(R1)          ;IS THIS A JUMP CMD?
5670 027652          BNE 6$          ;BR IF NOT.
5671 027654          LET R1 := R1 + #2          ;POINT TO BRF.
5672 027660          MOV (R1)+,JLOC          ;SAVE BRF (LOCATION).
5673 027664          CMP (R1)+,JLOOP          ;HAS LOOP COUNT BE SATISFIED?
5674 027670          BNE 1$          ;IF NOT, JMP AGAIN.
5675 027672          LET R1 := R1 + #2          ;IF SO, ADJUST SEQ POUNTER
5676 027676          BR 3$          ;AND GO TO NEXT COMMAND.
5677 027700          1$: LET JLOOP := JLOOP + #1          ;UPDATE THE LOOP COUNT.
5678 027704          LET R1 := #CMDSEQ          ;INIT CMD SEQ TABLE POINTER.
5679 027710          2$: DEC JLOC          ;DECR LOCATION COUNTER.

```



```

030136 005002
5706 030140
030140
5707 030140
030140 000137 030172
030144
5708 030144
030144 032765 000001 003516
030152 001406
030154 032737 000001 003426
030162 001402
030164 000137 030172
030170
5709
5710 030170
030170 005002
5711 030172
030172
5712 030172
030172
5713 030172 004737 017734
5714 030176
030176 000735
030200
5715 030200
030200 020227 000001
030204 001402
030206 000137 030254
5716 030212
030212 013737 003420 003422
030220 005237 003422
5717 030224
030224 105237 003543
5718 030230
030230 023727 003434 000002
030236 001402
030240 000137 030250
5719 030244 004737 030364
5720 030250
030250
5721 030250
030250 000137 030260
030254
5722 030254
030254 105037 003543
5723 030260
030260
5724 030260
030260 005237 003420
5725 030264
030264 013737 003426 003432
5726 030272
030272 000647
030274
5727 030274 004737 016530
5728
5729

```

```

ENDIF
ELSE
IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN
LET R2 := #0
ENDIF
ENDIF
JSR PC,NEXTU
ENDDO
IF R2 EQ #1 THEN
LET NCNT1 := NCNT + #1
LET ALLEOT :B= ALLEOT + #1
IF CMDLG EQ #2 THEN
JSR PC,TSWEOT
ENDIF
ELSE
LET ALLEOT :B= #0 ;WHEN NOT ALL @EOT, CLEAR FLAG
ENDIF
LET NCNT := NCNT + #1
LET PCMDWD := CMDWRD
ENDDO
JSR PC,VFYDAT

```

```

CLR R2
50523$:
NOT REVERSE:
JMP 50524$
50522$:
IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN
BIT #X0.EOT,EOTFLG(R5)
BEQ 50525$
BIT #CMD.CO,CMDWRD
BEQ .+6
JMP 50526$
50525$:
;IF NOT AT EOT OR NOT A MOTION CMD THEN:
;CLEAR EOT/BOT FLAG.
CLR R2
50526$:
50524$:
;FIND NEXT UNIT
;
BR 50520$
50521$:
;IF ALL UNIT ARE AT EOT/BOT THEN:
CMP R2,#1
BEQ .+6
JMP 50527$
MOV NCNT,NCNT1
INC NCNT1
INCB ALLEOT
CMP CMDLG,#2
BEQ .+6
JMP 50530$
50530$:
JMP 50531$
50527$:
CLR B ALLEOT
50531$:
;UPDATE RECORD COUNT.
INC NCNT
;SAVE PREVIOUS COMMAND WORD.
MOV CMDWRD,PCMDWD
BR 50515$
50516$:
;IF LAST CMD WAS A WRITE VERIFY, THEN GO
;VERIFY THE LAST N RECORDS OF DATA.

```


HARDWARE TESTS MACRO M1200 28-MAR-85 15:28 PAGE 108-3
 TEST 6: EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.

SEQ 0168

```

5730 030300          ENDDO
      030300 000137 027634
      030304
                    JMP      50513$
                    50514$:
5731
5732          177777
5733 030304          $BRJMP=-1 ;TURN OFF JMP SUBSTITUTION (SPMACJ CONTROL, ONLY).
      030304 105037 003543      LET ALLEOT :B= #0 ;NOT AT EOT
5734 030310          LET RPTFLG :B= #1 ;REQUEST A REPORT      CLRB      ALLEOT
      030310 112737 000001 003535      LET RPTFLG :B= #1
5735 030316          LET R1 := #CMDSEQ ;SET UP THE COMMAND TABLE ADDRESS
      030316 012701 003554      LET R1 := #CMDSEQ      MOV      #1,RPTFLG
5736 030322 004737 007016      JSR PC,SETRW ;STORE REWIND COMMAND      MOV      #CMDSEQ,R1
5737 030326          LET (R1) := #END ;TERMINATE THE TABLE
      030326 012711 177777      LET (R1) := #END
5738 030332 004737 007036      JSR PC,EXALL ;ISSUE THE REWINDS      MOV      #END,(R1)
5739
5740
5741 030336          EXIT      TST
      030336 104432
      030340 000130
                    TRAP      C$EXIT
                    .WORD      L10041-.
5742

```

```

5744
5745
5746      ;      SUBROUTINE TO MOVE A COMMAND FROM THE SOFTWARE P ^ABLE TO
5747      ;      THE COMMAND SEQUENCE TABLE.
5748      ;      INPUTS:          R2 = POINTER TO SOFT "P" TABLE
5749      ;      OUTPUTS:
5750      ;      REGISTERS:      R3.
5751      ;      CALLS:
5752      PTCMDS: LET R3 := (R2)+ - #1 SHIFT +1 ;R3 = COMMAND TABLE INDEX.
5753      030342 012203      MOV      (R2)+,R3
5754      030344 005303      DEC      R3
5755      030346 006303      ASL      R3
5756      030350 016321 003656 LET (R1)+ := CMDTBL(R3) ;MOVE COMMAND WORD.
5757      030354 012221      MOV      CMDTBL(R3),(R1)+
5758      030356 012221      LET (R1)+ := (R2)+ ;MOVE # OF BYTES.
5759      030360 012221      LET (R1)+ := (R2)+ ;MOVE # OF OPERATIONS.
5760      030362 000207      LET (R1)+ := (R2)+ ;MOVE PATTERN CODE.
5761      030364 000207      RTS PC ;MOVE (R2)+,(R1)+
5762
5763      ;      SUBROUTINE TO WRITE THEN READ REVERSE ONE RECORD BEYOND EOT
5764      ;      INPUTS:
5765      ;      OUTPUTS:
5766      ;      REGISTERS:
5767      ;      CALLS:          CMDAC,EXSUB,CKHAE
5768      T5WEOT: NOP
5769      030364 000240      NOP
5770      030366 000240      JSR PC,EXSUB ;WRITE ONE RECORD BEYOND EOT
5771      030370 004737 007332 JSR PC,CKHAE ;SO THAT READ SHORTER STOP DISTANCE
5772      030374 004737 020274 ;SHALL POSITION HEAD IN CLEAN IRG GAP
5773      030400 013737 003426 003432 LET PCMDWD := CMDWRD ;REPOSITION TAPE
5774      030406 012737 104401 003426 LET CMDWRD := #RDR ;BEFORE EXTRA RECORD
5775      030414 012737 000004 003434 LET CMDLG := #4 ;BY READING REVERSE
5776      030422 013737 003426 002314 LET CMDPKT := CMDWRD CLR.BY #BRF.C ;MOV      CMDWRD,PCMDWD
5777      030430 042737 004000 002314 ;BIC      #BRF.C,CMDPKT
5778      030436 013737 002314 003430 LET CMDSAV := CMDPKT ;THAT RECORD TO ALLOW
5779      030444 013737 003416 002316 LET CMDPKT+CP.ADL := DATARD ;NEXT COMMAND IN THE
5780      030452 004737 007672 ;MOV      CMDPKT,CMDSAV
5781      030456 004737 007332 ;MOV      DATARD,CMDPKT+CP.ADL
5782      030462 004737 020274 JSR PC,CMDAC ;TABLE TO BE EXECUTED
5783      030466 000207 JSR PC,EXSUB
5784      030470 030470 104401 JSR PC,CKHAE
5785      030472 030472 104401 RTS PC
5786      030474 030474 104401 .EVEN
5787      030476 030476 104401 ENDTST
5788      030478 030478 104401 L10041:
5789      030480 030480 104401 TRAP      C#ETST
5790      030482 030482 104401 ENDMOD
    
```

```

5785          .TITLE PARAMETER CODING
5796
5797          .SBTTL  HARDWARE PARAMETER CODING SECTION
5806
5807 030472          BGNMOD
5808
5809
5810          ;**
5811          ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
5812          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
5813          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
5814          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
5815          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
5816          ; WITH THE OPERATOR.
5817          ;--
5818 030472          BGNHRD
5819 030472 000024
5820 030474          L$HARD::          .WORD L10042-L$HARD/2
5821
5822 030474          GPRMA  TS4ADR,0,0,160002,177564,YES
5823 030476 000031          .WORD  T$CODE
5824 030500 030520          .WORD  TS4ADR
5825 030502 160002          .WORD  T$LLOLIM
5826 030504 177564          .WORD  T$HILIM
5827 030504          GPRMD  TS4VCT,2,0,777,60,776,YES
5828 030504 001032          .WORD  T$CODE
5829 030506 030535          .WORD  TS4VCT
5830 030510 000777          .WORD  777
5831 030512 000060          .WORD  T$LLOLIM
5832 030514 000776          .WORD  T$HILIM
5833
5834 030516          EXIT HRD
5835 030516 013004          .WORD  T$CODE
5836
5837 030520          .NLIST  BEX
5838 030535          .ASCIZ  /TSSR ADDRESS/
5839          .ASCIZ  /VECTOR/
5840          .LIST  BEX
5841          .EVEN
5842 030544          ENDHRD
5843
5844          L10042:          .EVEN

```

```

5845 .SBTTL SOFTWARE PARAMETER CODING SECTION
5846
5847
5848 ;**
5849 ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
5850 ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES THE
5851 ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
5852 ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
5853 ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
5854 ; WITH THE OPERATOR.
5855 ;--
5856 030544          BGNSFT
      030544 000531
      030546
5857 L$SOFT::
5864
5865 030546          GPRML  CLRM,0,1,YES
      030546 000130
      030550 031324
      030552 000001
5866 030554          GPRML  RRVM,0,400,YES
      030554 000130
      030556 031343
      030560 000400
5867 030562          GPRML  HAEM,2,1,YES
      030562 001130
      030564 031372
      030566 000001
5868 030570          GPRML  RCVERM,2,400,YES
      030570 001130
      030572 031416
      030574 000400
5869 030576          GPRML  IRECM,4,1,YES
      030576 002130
      030600 031440
      030602 000001
5870 030604          XFERT  NEXTSP
      030604 004024
5871 030606          GPRML  BADTM,4,400,YES
      030606 002130
      030610 031461
      030612 000400
5872 030614          NEXTSP: GPRML  DINTM,6,1,YES
      030614 003130
      030616 031506
      030620 000001
5873 030622          GPRML  IREM,6,400,YES
      030622 003130
      030624 031531
      030626 000400
5874 030630          GPRML  RAMM,10,1,YES
      030630 004130
      030632 031564
      030634 000001
5875
5876 030636          GPRML  EWPT,12,1,YES
      030636 005130

```

.WORD L10043-L\$SOFT/2

.WORD T\$CODE
.WORD CLRM
.WORD 1

.WORD T\$CODE
.WORD RRVM
.WORD 400

.WORD T\$CODE
.WORD HAEM
.WORD 1

.WORD T\$CODE
.WORD RCVERM
.WORD 400

.WORD T\$CODE
.WORD IRECM
.WORD 1

.WORD T\$CODE
.WORD BADTM
.WORD 400

.WORD T\$CODE
.WORD DINTM
.WORD 1

.WORD T\$CODE
.WORD IREM
.WORD 400

.WORD T\$CODE
.WORD RAMM
.WORD 1

.WORD T\$CODE

	030640	031610			.WORD	EWPT
	030642	000001			.WORD	1
5877	030644		GPRML	CHGM,12,400,YES		
	030644	005130			.WORD	T\$CODE
	030646	031646			.WORD	CHGM
	030650	000400			.WORD	400
5878	030652		XFERF	ENDSP1		
	030652	127044			.WORD	T\$CODE
5879	030654		GPRMD	CHARM,14,0,377,0,777,YES		
	030654	006032			.WORD	T\$CODE
	030656	031665			.WORD	CHARM
	030660	000377			.WORD	377
	030662	000000			.WORD	T\$LLOLIM
	030664	000777			.WORD	T\$HILIM
5880	030666		GPRMD	CMD2M,16,D,37,1,33,YES		
	030666	007052			.WORD	T\$CODE
	030670	031712			.WORD	CMD2M
	030672	000037			.WORD	37
	030674	000001			.WORD	T\$LLOLIM
	030676	000033			.WORD	T\$HILIM
5881	030700		GPRMD	BPCRM,20,D,-1,1,DATCNT,YES		
	030700	010052			.WORD	T\$CODE
	030702	031720			.WORD	BPCRM
	030704	177777			.WORD	-1
	030706	000001			.WORD	T\$LLOLIM
	030710	010000			.WORD	T\$HILIM
5882	030712		GPRMD	NUMBM,22,D,-1,1,77777,YES		
	030712	011052			.WORD	T\$CODE
	030714	031732			.WORD	NUMBM
	030716	177777			.WORD	-1
	030720	000001			.WORD	T\$LLOLIM
	030722	077777			.WORD	T\$HILIM
5883	030724		GPRMD	PATTM,24,D,17,0,10,YES		
	030724	012052			.WORD	T\$CODE
	030726	031752			.WORD	PATTM
	030730	000017			.WORD	17
	030732	000000			.WORD	T\$LLOLIM
	030734	000010			.WORD	T\$HILIM
5884	030736		GPRMD	CMD3M,26,D,37,1,33,YES		
	030736	013052			.WORD	T\$CODE
	030740	031764			.WORD	CMD3M
	030742	000037			.WORD	37
	030744	000001			.WORD	T\$LLOLIM
	030746	000033			.WORD	T\$HILIM
5885	030750		GPRMD	BPCRM,30,D,-1,1,DATCNT,YES		
	030750	014052			.WORD	T\$CODE
	030752	031720			.WORD	BPCRM
	030754	177777			.WORD	-1
	030756	000001			.WORD	T\$LLOLIM
	030760	010000			.WORD	T\$HILIM
5886	030762		GPRMD	NUMBM,32,D,-1,1,77777,YES		
	030762	015052			.WORD	T\$CODE
	030764	031732			.WORD	NUMBM
	030766	177777			.WORD	-1
	030770	000001			.WORD	T\$LLOLIM
	030772	077777			.WORD	T\$HILIM
5887	030774		GPRMD	PATTM,34,D,17,0,10,YES		

	030774	016052			.WORD	T\$CODE
	030776	031752			.WORD	PATM
	031000	000017			.WORD	17
	031002	000000			.WORD	T\$LLOLIM
	031004	000010			.WORD	T\$HILIM
5888	031006		GPRMD	CMD4M,36,D,37,1,33,YES		
	031006	017052			.WORD	T\$CODE
	031010	031772			.WORD	CMD4M
	031012	000037			.WORD	37
	031014	000001			.WORD	T\$LLOLIM
	031016	000033			.WORD	T\$HILIM
5889	031020		GPRMD	BPCRM,40,D,-1,1,DATCNT,YES		
	031020	020052			.WORD	T\$CODE
	031022	031720			.WORD	BPCRM
	031024	177777			.WORD	-1
	031026	000001			.WORD	T\$LLOLIM
	031030	010000			.WORD	T\$HILIM
5890	031032		GPRMD	NUMBM,42,D,-1,1,77777,YES		
	031032	021052			.WORD	T\$CODE
	031034	031732			.WORD	NUMBM
	031036	177777			.WORD	-1
	031040	000001			.WORD	T\$LLOLIM
	031042	077777			.WORD	T\$HILIM
5891	031044		GPRMD	PATM,44,D,17,0,10,YES		
	031044	022052			.WORD	T\$CODE
	031046	031752			.WORD	PATM
	031050	000017			.WORD	17
	031052	000000			.WORD	T\$LLOLIM
	031054	000010			.WORD	T\$HILIM
5892	031056		GPRMD	CMD5M,46,D,37,1,33,YES		
	031056	023052			.WORD	T\$CODE
	031060	032000			.WORD	CMD5M
	031062	000037			.WORD	37
	031064	000001			.WORD	T\$LLOLIM
	031066	000033			.WORD	T\$HILIM
5893	031070		GPRMD	BPCRM,50,D,-1,1,DATCNT,YES		
	031070	024052			.WORD	T\$CODE
	031072	031720			.WORD	BPCRM
	031074	177777			.WORD	-1
	031076	000001			.WORD	T\$LLOLIM
	031100	010000			.WORD	T\$HILIM
5894	031102		GPRMD	NUMBM,52,D,-1,1,77777,YES		
	031102	025052			.WORD	T\$CODE
	031104	031732			.WORD	NUMBM
	031106	177777			.WORD	-1
	031110	000001			.WORD	T\$LLOLIM
	031112	077777			.WORD	T\$HILIM
5895	031114		GPRMD	PATM,54,D,17,0,10,YES		
	031114	026052			.WORD	T\$CODE
	031116	031752			.WORD	PATM
	031120	000017			.WORD	17
	031122	000000			.WORD	T\$LLOLIM
	031124	000010			.WORD	T\$HILIM
5896	031126		XFER	ENDSP2		
	031126	002004			.WORD	T\$CODE
5897	031130		ENDSP1: XFER	ENDSP		
	031130	075004			.WORD	T\$CODE

5898	031132		ENDSP2: GPRMD	CMD6M,56,D,37,1,33,YES		
	031132	027052			.WORD	T\$CODE
	031134	032006			.WORD	CMD6M
	031136	000037			.WORD	37
	031140	000001			.WORD	T\$LLOLIM
	031142	000033			.WORD	T\$HILIM
5899	031144		GPRMD	BPCRM,60,D,-1,1,DATCNT,YES		
	031144	030052			.WORD	T\$CODE
	031146	031720			.WORD	BPCRM
	031150	177777			.WORD	-1
	031152	000001			.WORD	T\$LLOLIM
	031154	010000			.WORD	T\$HILIM
5900	031156		GPRMD	NUMBM,62,D,-1,1,77777,YES		
	031156	031052			.WORD	T\$CODE
	031160	031732			.WORD	NUMBM
	031162	177777			.WORD	-1
	031164	000001			.WORD	T\$LLOLIM
	031166	077777			.WORD	T\$HILIM
5901	031170		GPRMD	PATTM,64,D,17,0,10,YES		
	031170	032052			.WORD	T\$CODE
	031172	031752			.WORD	PATTM
	031174	000017			.WORD	17
	031176	000000			.WORD	T\$LLOLIM
	031200	000010			.WORD	T\$HILIM
5902	031202		GPRMD	CMD7M,66,D,37,1,33,YES		
	031202	033052			.WORD	T\$CODE
	031204	032014			.WORD	CMD7M
	031206	000037			.WORD	37
	031210	000001			.WORD	T\$LLOLIM
	031212	000033			.WORD	T\$HILIM
5903	031214		GPRMD	BPCRM,70,D,-1,1,DATCNT,YES		
	031214	034052			.WORD	T\$CODE
	031216	031720			.WORD	BPCRM
	031220	177777			.WORD	-1
	031222	000001			.WORD	T\$LLOLIM
	031224	010000			.WORD	T\$HILIM
5904	031226		GPRMD	NUMBM,72,D,-1,1,77777,YES		
	031226	035052			.WORD	T\$CODE
	031230	031732			.WORD	NUMBM
	031232	177777			.WORD	-1
	031234	000001			.WORD	T\$LLOLIM
	031236	077777			.WORD	T\$HILIM
5905	031240		GPRMD	PATTM,74,D,17,0,10,YES		
	031240	036052			.WORD	T\$CODE
	031242	031752			.WORD	PATTM
	031244	000017			.WORD	17
	031246	000000			.WORD	T\$LLOLIM
	031250	000010			.WORD	T\$HILIM
5906	031252		GPRMD	CMD8M,76,D,37,1,33,YES		
	031252	037052			.WORD	T\$CODE
	031254	032022			.WORD	CMD8M
	031256	000037			.WORD	37
	031260	000001			.WORD	T\$LLOLIM
	031262	000033			.WORD	T\$HILIM
5907	031264		GPRMD	BPCRM,100,D,-1,1,DATCNT,YES		
	031264	040052			.WORD	T\$CODE
	031266	031720			.WORD	BPCRM


```

5913
5920 .NLIST BEX
5921 031324 103 114 105 CLRM: .ASCIZ /CLEAR COUNTERS/
5922 031343 122 105 123 RRVN: .ASCIZ /RESET RANDOM VARIABLES/
5923 031372 110 101 114 HAEM: .ASCIZ /HALT AFTER EACH CMD/
5924 031416 120 122 111 RCVERM: .ASCIZ /PRINT SOFT ERRORS/
5925 031440 111 116 110 IRECM: .ASCIZ /INHIBIT RECOVERY/
5926 031461 102 101 104 BADTM: .ASCIZ /BAD TAPE SPOT DETECT/
5927 031506 104 111 123 DINTM: .ASCIZ /DISABLE INTERRUPTS/
5928 031531 111 116 110 IREM: .ASCIZ /INHIBIT RFC ERROR REPORT/
5929 .LIST BEX
5930 .EVEN
5931 031562 JMPMSG:
5932 031562 XFER JMPMS2
5933 031564 100004 .WORD T$CODE
5933 031564 103 117 116 RAMM: .ASCIZ /CONTROLLER RAM DUMP/
5933 031567 124 122 117
5933 031572 114 114 105
5933 031575 122 040 122
5933 031600 101 115 040
5933 031603 104 125 115
5933 031606 120 000
5934 031610 105 116 101 EWPT: .ASCIZ /ENABLE EARLY WARNING MESSAGES/
5934 031613 102 114 105
5934 031616 040 105 101
5934 031621 122 114 131
5934 031624 040 127 101
5934 031627 122 116 111
5934 031632 116 107 040
5934 031635 115 105 123
5934 031640 123 101 107
5934 031643 105 123 000
5935 031646 103 110 101 CHGM: .ASCIZ /CHANGE CMD SEQ/
5935 031651 116 107 105
5935 031654 040 103 115
5935 031657 104 040 123
5935 031662 105 121 000
5936 031665 103 110 101 CHARM: .ASCIZ /CHARACTERISTICS CODE/
5936 031670 122 101 103
5936 031673 124 105 122
5936 031676 111 123 124
5936 031701 111 103 123
5936 031704 040 103 117
5936 031707 104 105 000
5937 031712 103 115 104 CMD2M: .ASCIZ "CMD/2"
5937 031715 057 062 000
5938 031720 102 122 106 BPCRM: .ASCIZ /BRF COUNT/
5938 031723 040 103 117
5938 031726 125 116 124
5938 031731 000
5939 031732 043 040 117 NUMBM: .ASCIZ /# OF OPERATIONS/
5939 031735 106 040 117
5939 031740 120 105 122
5939 031743 101 124 111
5939 031746 117 116 123
5939 031751 000
5940 031752 120 101 124 PATTM: .ASCIZ /PATTERN/

```

```
031755 124 105 122
031760 116 000
5941
5942 .LIST BEX
5943 031762 JMPMS2: .EVEN
5944 031762 023004 EXIT SFT
5945 .WORD T$CODE
5946
5947 031764 103 115 104 CMD3M: .NLIST BEX
5948 031772 103 115 104 CMD4M: .ASCIZ "CMD/3"
5949 032000 103 115 104 CMD5M: .ASCIZ "CMD/4"
5950 032006 103 115 104 CMD6M: .ASCIZ "CMD/5"
5951 032014 103 115 104 CMD7M: .ASCIZ "CMD/6"
5952 032022 103 115 104 CMD8M: .ASCIZ "CMD/7"
5953 .LIST BEX
5954 032030 ENDSFT
032030 L10043: .EVEN
5955
5956 ;*****
5957 ;*****
5958 ; PATCH AREA
5959
5960 032030 PATCH:: .BLKW 64.
5961 ;*****
5962 ;*****
5963
5964 032230 LASTAD
032230 032244 .EVEN
032232 000004 .WORD T$FREE
032234 .WORD T$SIZE
5965 032234 L$LAST:: ENDMOD
```

5967
5968
5969
5970
5971
5972
5973 032234
5974 032234
032234 000000
032236 000002
032240
5975 032240 172522
5976 032242 000224
5977 032244
032244
5978 032244
5979
5980 000001

.SBTTL HARD CODED P-TBL
;+
;DIAG IS PRE-PARAMETERIZED PER TBL
;--
BGNSETUP 1
BGNPTAB
L10044:
172522
224
ENDPTAB
L10046:
ENDSETUP
.END

.WORD 0
.WORD L10046-./2-1

PARAMETER CODING
SYMBOL TABLE

MACRO M1200 28-MAR-85 15:28 PAGE 114-1

SEQ 0179

ACK.C = 100000 G	BTMSG2 015407	CP.ADL = 000002 G	C\$SPRI = 000041	EXPBOT 003532 G
ADR = 000020 G	BTMSG3 015464	CP.CMD = 000000 G	C\$SVEC = 000037	EXSUB 007332 G
ALLEOT 003543 G	BTPPT 003526 G	CP.CNT = 000006 G	C\$TPRI = 000013	E\$END = 002100
ASSEMB = 000010	BTRPT 021034	CRLF 005307 G	DATARD 003416 G	E\$LOAD = 000035
ATTNM 004431 G	BTO 003000 G	CRLFSP 005312 G	DATAWT 003414 G	FATSM 004467 G
AUDRPM 004741 G	BT1 003052 G	CTCC 003456 G	DATCNT = 010000 G	FIRSTU 017666 G
AUTODM 024150	BT2 003124 G	CVC.C = 040000 G	DATRAT 003412 G	FMT.CO = 000040 G
BADTM 031461	BT3 003176 G	C\$AU = 000052	DEVTBL 002536 G	FMT.C1 = 000100 G
BADTSW 002211 G	BYTECO = 000403	C\$AUTO = 000061	DFPTBL 002176 G	FTLCNT 003320 G
BFSEQ 025332	CHAR 002220 G	C\$BRK = 000022	DFTSCH = 000040 G	FUNRM 004447 G
BFSEQ0 025356	CHARM 031665	C\$BSEG = 000004	DIA = 100006 G	F\$AU = 000015
BFSEQ1 025430	CHGFLG 002217 G	C\$BSUB = 000002	DIABLK = 003414 G	F\$AUTO = 000020
BFSEQ2 025442	CHGM 031646	C\$CEFG = 000045	DIACNT = 000020 G	F\$BGN = 000040
BFSEQ3 025534	CHKERR 012456 G	C\$CLCK = 000062	DIAGMC = 000000	F\$CLEA = 000007
BFSEQ4 025606	CH.EAI = 000040 G	C\$CLEA = 000012	DINT 002212 G	F\$DU = 000016
BFSEQ5 025650	CH.ERI = 000020 G	C\$CLOS = 000035	DINTM 031506	F\$END = 000041
BFSEQ6 025722	CH.ESS = 000200 G	C\$CLP1 = 000006	DLY = 000020 G	F\$HARD = 000004
BFSEQ7 025754	CKDATA 017252 G	C\$CVEC = 000036	DLY.C = 000020 G	F\$HW = 000013
BFSEQ8 026006	CKDCNT 017662	C\$DCLN = 000044	DRI = 100013 G	F\$INIT = 000006
BFSEQ9 026040	CKDFF 017664	C\$DODU = 000051	DROPDN 020210	F\$JMP = 000050
BFSE10 026062	CKHAE 020274 G	C\$DRPT = 000024	DROPEP 003541 G	F\$MOD = 000000
BGNFLG = 003474	CKHRTN 020362	C\$DU = 000053	DROPN 020210	F\$MSG = 000011
BINC 016310	CKKFF 017664	C\$EDIT = 000003	DROPU 017770 G	F\$PROT = 000021
BIT0 = 000001 G	CLN = 101012 G	C\$ERDF = 000055	DROPUA 020120	F\$PWR = 000017
BIT00 = 000001 G	CLRERR 012110 G	C\$ERHR = 000056	DRORTN 020176	F\$RPT = 000012
BIT01 = 000002 G	CLRFLG 002204 G	C\$ERRO = 000060	DTAERM 005472 G	F\$SEG = 000003
BIT02 = 000004 G	CLRM 031324	C\$ERSF = 000054	DTAER2 004773 G	F\$SOFT = 000005
BIT03 = 000010 G	CMDAC 007672 G	C\$ERSO = 000057	DTAER3 005042 G	F\$SRV = 000010
BIT04 = 000020 G	CMDASC 003744 G	C\$ESCA = 000010	DTAER4 005104 G	F\$SUB = 000002
BIT05 = 000040 G	CMDDD 002222 G	C\$ESEG = 000005	DTAER5 005125 G	F\$SW = 000014
BIT06 = 000100 G	CMDLG 003434 G	C\$ESUB = 000003	EF.CON = 000036 G	F\$TEST = 000001
BIT07 = 000200 G	CMDPKM 004176 G	C\$ETST = 000001	EF.NEW = 000035 G	GCMDA 007746 G
BIT08 = 000400 G	CMDPKT 002314 G	C\$EXIT = 000032	EF.PWR = 000034 G	GENPAT 010356 G
BIT09 = 001000 G	CMDSAV 003430 G	C\$GETB = 000026	EF.RES = 000037 G	GES = 100017 G
BIT1 = 000002 G	CMDSEQ 003554 G	C\$GETW = 000027	EF.STA = 000040 G	GETSTM 005253 G
BIT10 = 002000 G	CMDSE2 003564 G	C\$GMAN = 000043	EINC 016316	GIT 010734
BIT11 = 004000 G	CMDTBL 003656 G	C\$GPHR = 000042	END = 177777 G	GOWAIT 011470 G
BIT12 = 010000 G	CMDWRD 003426 G	C\$GPLO = 000030	ENDERF = 003506	GSCPK 00324 G
BIT13 = 020000 G	CMD.CO = 000001 G	C\$GPRI = 000040	ENDFLG = 003546	G\$CNT0 = 000200
BIT14 = 040000 G	CMD.C1 = 000002 G	C\$INIT = 000011	ENDSP 031322	G\$DELM = 000372
BIT15 = 100000 G	CMD.C2 = 000004 G	C\$INLP = 000020	ENDSP1 031130	G\$DISP = 000003
BIT2 = 000004 G	CMD.C3 = 000010 G	C\$MANI = 000050	ENDSP2 031132	G\$EXCP = 000400
BIT3 = 000010 G	CMD.C4 = 000020 G	C\$MEM = 000031	EOTFLG 003516 G	G\$HILI = 000002
BIT4 = 000020 G	CMD2M 031712	C\$MSG = 000023	ERCVER 002207 G	G\$LOLI = 000001
BIT5 = 000040 G	CMD3M 031764	C\$OPEN = 000034	ERLOG 003502 G	G\$NO = 000000
BIT6 = 000100 G	CMD4M 031772	C\$PNTB = 000014	ERRREC 003505 G	G\$OFFS = 000400
BIT7 = 000200 G	CMD5M 032000	C\$PNTF = 000017	ERS = 100411 G	G\$OFSI = 000376
BIT8 = 000400 G	CMD6M 032006	C\$PNTS = 000016	ERSFLG 003545 G	G\$PRMA = 000001
BIT9 = 001000 G	CMD7M 032014	C\$PNTX = 000015	EVL = 000004 G	G\$PRMD = 000002
BOE = 000400 G	CMD8M 032022	C\$QIO = 000377	EWMMSG 013050	G\$PRML = 000000
BORERS 015534 G	CMPDAT 003410 G	C\$RDBU = 000007	EWPRNT 002216 G	G\$RADA = 000140
BPCRM 031720	CNTBGN = 002560	C\$REFG = 000047	EWPT 031610	G\$RADB = 000000
BRCPK 002330 G	CNTEND = 003330	C\$RESE = 000033	EWTRY 013174 G	G\$RADD = 000040
BRFCNT 003424 G	CNTLEN = 000550 G	C\$REVI = 000003	EXALL 007036 G	G\$RADL = 000120
BRF.C = 004000 G	CODELM 004066 G	C\$RFLA = 000021	EXARTN 007330	G\$RADO = 000020
BTADDR 002550 G	COUNTE = 050436	C\$RPT = 000025	EXCRTN 011466	G\$XFER = 000004
BTMSG1 015322	CP.ADH = 000004 G	C\$SEFG = 000046	EXCUTE 010774 G	G\$YES = 000010

PARAMETER CODING
SYMBOL TABLE

MACRO M1200 28-MAR-85 15:28 PAGE 114-2

SEQ 0180

HAE	002206	G	L\$APT	002036	G	L10005	006612	NINUSE=	177774	G	RAMR5H	003344	
HAEM	031372		L\$AU	024424	G	L10006	006634	NOINTM	004515	G	RAMSIZ	003406	G
HALTM	004136	G	L\$AUT	002070	G	L10007	006656	NRDYM	024244		RAMWRT	002214	G
HELP	= 000000		L\$AUTO	023660	G	L10010	010756	NSSRM	004365	G	RANB	003440	G
HERE	003340		L\$CCP	002106	G	L10011	022242	NUMBM	031732		RANBC =	153624	G
HOE	= 100000	G	L\$CLEA	024302	G	L10013	023656	NURTY1	005167	G	RANCMD	026366	
HRDCNT	003310	G	L\$CO	002032	G	L10014	024146	OFLINM	005223	G	RANDOM	003533	G
IBE	= 010000	G	L\$DEPO	002011	G	L10015	024356	ONEFIL=	000001		RANP =	000007	G
IDU	= 000040	G	L\$DESC	002140	G	L10016	024422	OPFLAG	003552	G	RANRD	026426	
IER	= 020000	G	L\$DESP	002076	G	L10017	024516	OPP.C =	020000	G	RANS	003442	G
IE.C	= 000200	G	L\$DEVP	002060	G	L10020	026104	O\$APTS=	000000		RANSC =	032561	G
INIT10	022252		L\$DISP	002124	G	L10021	024652	O\$AU =	000001		RANW	026526	
INIT15	022604		L\$DLY	002116	G	L10022	024676	O\$BGNR=	000001		RANWR	026502	
INIT16	022624		L\$DTP	002040	G	L10023	024716	O\$BGNS=	000001		RANWV	026514	
INTFLG	003506	G	L\$DTP	002034	G	L10024	024736	O\$DU =	000001		RAN1	010736	G
INTPRI=	000340	G	L\$DU	024360	G	L10025	024756	O\$ERRT=	000000		RAN2	010740	G
IRE	003540	G	L\$DUT	002072	G	L10026	024776	O\$GNSW=	000001		RAN3	010742	G
IREC	002210	G	L\$DVTY	002166	G	L10027	025016	O\$POIN=	000001		RCVERM	031416	
IREC	031440		L\$EF	002052	G	L10030	025036	O\$SETU=	000001		RDF =	104001	G
IREM	031531		L\$ENVI	002044	G	L10031	025056	PASCNT	003260	G	RDR =	104401	G
ISR	= 000100	G	L\$ETP	002102	G	L10032	025076	PATCH	032030	G	RECCNT	003330	G
IXE	= 004000	G	L\$EXP1	002046	G	L10033	025134	PATERN	003454	G	RECLOG	003501	G
I\$AU	= 000041		L\$EXP4	002064	G	L10034	025320	PATRO	010442	G	RECREG	006560	
I\$AUTO=	000041		L\$EXP5	002066	G	L10035	026560	PATR1	010500	G	RECTAP	006674	G
I\$CLN =	000041		L\$HARD	030474	G	L10036	027246	PATR2	010520	G	RECUD	012310	G
I\$DU =	000041		L\$HIME	002120	G	L10037	027412	PATR3	010530	G	REPEAT=	050256	
I\$HRD =	000041		L\$HPCP	002016	G	L10040	027524	PATR4	010554	G	RERM	004644	G
I\$INIT=	000041		L\$HPTP	002022	G	L10041	030470	PATR5	010566	G	RETRY =	050254	
I\$MOD =	000041		L\$HW	002176	G	L10042	030544	PATR6	010600	G	RETRYC	003474	G
I\$MSG =	000041		L\$ICP	002104	G	L10043	032030	PATR7	010620	G	REWRT	015710	
I\$PROT=	000040		L\$INIT	022252	G	L10044	032240	PATR8	010744	G	RFBC	002660	G
I\$PTAB=	000041		L\$LADP	002026	G	L10046	032244	PATBL	010420		RFCERM	004350	G
I\$PWR =	000041		L\$LAST	032234	G	MBR =	100012	PATM	031752		RFREC	002760	G
I\$RPT =	000041		L\$LOAD	002100	G	MEMOM	023554	PCMDWD	003432	G	RFUNR	002770	G
I\$SEG =	000041		L\$LUN	002074	G	MISCFG	003551	PIRE	002213	G	RLEXM	004404	G
I\$SETU=	000041		L\$MREV	002050	G	MOD.CO=	000400	PNT =	001000	G	RNF =	125401	G
I\$SFT =	000041		L\$NAME	002000	G	MOD.C1=	001000	PRI =	002000	G	RNOPSC=	177740	G
I\$SRV =	000041		L\$PRIO	002042	G	MOD.C2=	002000	PRI00 =	000000	G	RNR =	105401	G
I\$SUB =	000041		L\$PROT	022244	G	MOD.C3=	004000	PRI01 =	000040	G	RNYM	004600	G
I\$TST =	000041		L\$PRT	002112	G	MOVMSG	012224	PRI02 =	000100	G	RPF =	105001	G
JLOC	003452	G	L\$REPP	002062	G	MSGCNT=	000016	PRI03 =	000140	G	RPR =	125001	G
JLOOP	003450	G	L\$REV	002010	G	MSGPKA	002506	PRI04 =	000200	G	RPTCNT	003476	G
JMP	= 000040	G	L\$RPT	020440	G	MSGPKT	002340	PRI05 =	000240	G	RPTFLG	003535	G
JMPMSG	031562		L\$SOFT	030546	G	MSGPK0	002356	PRI06 =	000300	G	RPT1A	021302	
JMPMS2	031762		L\$SPC	002056	G	MSGPK1	002374	PRI07 =	000340	G	RPT1B	021357	
JMP.C =	000040	G	L\$SPCP	002020	G	MSGPK2	002412	PRXST	020212	G	RPT1C	021430	
J\$JMP =	000167		L\$SPTP	002024	G	MSGPK3	002430	PTCMDS	030342		RPT1D	021501	
LBRVEC	003466	G	L\$STA	002030	G	MS.RFC=	000004	PWRFLG	003547	G	RPT1E	021727	
LCSR	003464	G	L\$SW	002204	G	MS.XS0=	000006	RAMDAT	003346	G	RPT1F	021605	
LENMSK	003436	G	L\$TEST	002114	G	MS.XS1=	000010	RAMDUM	014436	G	RPT1G	021656	
LHERTZ	003472	G	L\$TIML	002014	G	MS.XS2=	000012	RAMER	016324	G	RPT1I	022046	
LOE	= 040000	G	L\$UNIT	002012	G	MS.XS3=	000014	RAMFHR	005320		RPT1J	021757	
LOG	016024	G	L10000	002202		NCMD.C=	177740	RAMHLD	003342		RPT1K	022037	
LOOPCT	003537	G	L10001	002312		NCNT	003420	RAMIOP	005377		RRANV	002205	G
LOT	= 000010	G	L10002	005636		NCNT1	003422	RAMLIN	005462		RRBC	002620	G
LVECT	003470	G	L10003	006562		NEXTSP	030614	RAMM	031564		RRECL =	000020	G
L\$ACP	002110	G	L10004	006570		NEXTU	017734	RAMPD	005450		RRREC	002740	G

PARAMETER CODING
SYMBOL TABLE

MACRO M1200 28-MAR-85 15:28 PAGE 114-3

SEQ 0181

RRUNR	002750	G	TIME1	003444	G	T\$TAGL=	177777	WRITEM	020364	G	\$F\$TRU=	000404
RRVM	031343		TIME2	003446	G	T\$TAGN=	010047	WRR	= 105005	G	\$F\$UNT=	000130
RS1	= 001233	G	TOERM	004303	G	T\$TEMP=	000000	WRREC	002720	G	\$F\$WHI=	000120
RS2	= 007622	G	TOOMM	004554	G	T\$TEST=	000006	WRT	= 104005	G	\$F\$YES=	000402
RS3	= 000000	G	TRAPD4	003550	G	T\$TSTM=	177777	WRTY	014736	G	\$IFLEV=	177777
RTLE	014600	G	TRAP4	024274	G	T\$TSTS=	000001	WRTYCT	003250	G	\$ISK0 =	000001
RTLRTN	014734		TRKMSG	013133		T\$AU =	010017	WRTYER	003500	G	\$ISK1 =	000001
RWCPK	002334	G	TSAM	004532	G	T\$AUT=	010014	WRTYFG	003477	G	\$ISK2 =	000001
RWD	= 102010	G	TSBA	= 002456	G	T\$CLE=	010015	WRUNR	002730	G	\$ISK3 =	000001
RWERR	003503	G	TSC.FC=	177717	G	T\$DAT=	010046	WSSR	012124	G	\$ISK4 =	000001
RSSAVE	003460	G	TSC.TC=	177761	G	T\$DU =	010016	WTM	= 100011	G	\$ISK5 =	000001
SCCNT	003270	G	TSDB	002456	G	T\$HAR=	010042	WTR	= 101011	G	\$ISK6 =	000001
SCERM	004324	G	TSSR	002466	G	T\$HW =	010000	WTV	= 104105	G	\$LOCTA=	177777
SCH	= 140004	G	TSSREG	003462	G	T\$INI=	010013	WTVERM	004260	G	\$LSTIN=	000001
SCHBK	002446	G	TSVCT	002476	G	T\$MSG=	010003	WTYBRF	015320		\$LSTTA=	000001
SCHCNT=	000010	G	TS.A16=	000400	G	T\$PC =	000001	WTYCMD	015314		\$LVTAG=	050436
SEGEND	003644	G	TS.A17=	001000	G	T\$PRO=	010012	WTYWRD	015316		\$NESTL=	177777
SETCH	006772	G	TS.NBA=	002000	G	T\$PTA=	010045	X\$ALWA=	000000		\$NSKO =	000120
SETRW	007016	G	TS.NXM=	004000	G	T\$RPT=	010011	X\$FALS=	000040		\$NSK1 =	000120
SETUP	010000	G	TS.OFL=	000100	G	T\$SOF=	010043	X\$OFFS=	000400		\$NSK2 =	000110
SFF	= 105010	G	TS.RMR=	010000	G	T\$SRV=	010010	X\$TRUE=	000020		\$NSK3 =	000110
SFPTBL	002204	G	TS.SC =	100000	G	T\$SUB=	010034	XO.BOT=	000002	G	\$NSK4 =	000110
SFR	= 105410	G	TS.SPE=	020000	G	T\$SW =	010001	XO.EOT=	000001	G	\$NSK5 =	000110
SRF	= 104010	G	TS.SSR=	000200	G	T\$TES=	010041	XO.LET=	020000	G	\$NSK6 =	000110
SRR	= 104410	G	TS.UPE=	040000	G	T1	024520	XO.ONL=	000100	G	\$SAVE =	000001
STAERM	005640	G	TS4ADR	030520		T1SWB	003542	XO.RLL=	010000	G	\$SAVLE=	177777
STAER1	006152		TS4CL	002526	G	T1.1	024530	XO.RLS=	040000	G	\$SELLE=	177777
STAER2	006330		TS4INT	002516	G	T1.10	025060	XO.TMK=	100000	G	\$SSKO =	050514
STAER3	006407		TS4INO	006564	G	T1.11	025100	X1.EWN=	000010	G	\$TAGLE=	177777
STAER4	006445		TS4IN1	006606	G	T1.12	025164	X2.OPM=	100000	G	\$TAGNU=	050532
STAER5	006465		TS4IN2	006630	G	T1.2	024654	X3.DCK=	000010	G	\$TEMP =	000402
STAER6	006274		TS4IN3	006652	G	T1.3	024700	X3.RNY=	157400	G	\$TSKO =	050513
STAER7	006244		TS4VCT	030535		T1.4	024720	ZROPAT	010504		\$TSK1 =	050514
STAF LG	003546	G	T\$ARGC=	000001		T1.5	024740	\$BGNLE=	000002		\$TSK2 =	050515
STREAM	003544	G	T\$CODE=	023004		T1.6	024760	\$BRJMP=	177777		\$TSK3 =	050516
SVCGBL =	000000		T\$ERRN=	000001		T1.7	025000	\$BSKO =	050254		\$TSK4 =	050531
SVCINS=	000001		T\$EXCP=	000000		T1.8	025020	\$BSK1 =	050256		\$TSK5 =	050530
SVCSUB=	000000		T\$FLAG=	000041		T1.9	025040	\$BSK2 =	050436		\$TSK6 =	050524
SVCTAG=	000000		T\$FREE=	032244		T2	026106	\$ERFLG=	000400		\$TSK7 =	050526
SVCTST=	000000		T\$GMAN=	000000		T3	026562	\$F\$AND=	000310		\$U	= 000403
SWBFLG	003536	G	T\$HILI=	000010		T4	027250	\$F\$BAD=	000401		\$ARGC=	000000
SWB.C =	010000	G	T\$LAST=	000001		T5	027414	\$F\$BLA=	000170		\$BYTE=	000403
SYMD =	000007		T\$LOLI=	000000		TSWEOT	030364	\$F\$CAS=	000150		\$CASE=	000000
SYMS =	000007		T\$LSYM=	010000		T6	027526	\$F\$DEC=	000220		\$DST =	000000
S\$LSYM=	010000		T\$LTNO=	000006		UAM	= 000200	\$F\$DO =	000340		\$ELOC=	000402
TAPCAP	022157		T\$NEST=	177777		UNL	= 100412	\$F\$FAL=	000405		\$ERFL=	000000
TCCRA	013154		T\$NS0 =	000000		UNREC	003504	\$F\$G00=	000400		\$FLAG=	000001
TCC0	013306	G	T\$NS1 =	000005		URERM	004666	\$F\$IF =	000110		\$FROM=	000000
TCC1	013330	G	T\$NS2 =	000002		VFEXC	016642	\$F\$INC=	000210		\$INH =	000403
TCC2	013346	G	T\$PCNT=	000000		VFISU	017126	\$F\$L00=	000200		\$LOC =	027701
TCC3	013514	G	T\$PTAB=	010045		VFYCNT	003300	\$F\$NAM=	000160		\$LOCN=	000000
TCC4	013536	G	T\$PTHV=	000001		VFYDAT	016530	\$F\$NO =	000403		\$REG =	177777
TCC5	014256	G	T\$PTNU=	000001		VFYFLG	003534	\$F\$OR =	000320		\$RETU=	000000
TCC6	014360	G	T\$SAVL=	177777		VFY.C =	000100	\$F\$RTI=	000350		\$RTN1=	000000
TCC7	014414	G	T\$SEGL=	177777		WLKZRO	010534	\$F\$RTN=	000300		\$RTN2=	000000
TC2RTN	013512		T\$SIZE=	000004		WRBC	002560	\$F\$SEL=	000140		\$SRC =	000000
TC4STO	003530	G	T\$SUBN=	000000		WRECL =	000020	\$F\$THE=	000330		\$TGSV=	000000

PARAMETER CODING
SYMBOL TABLE

MACRO M1200 28-MAR-85 15:28 PAGE 114-4

SEQ 0182

\$\$TGS1= 000000

\$\$TGS2= 000000

\$\$TO = 000000

\$\$\$TAG= 050000

. ABS. 032244 000
000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 58622 WORDS (229 PAGES)

DYNAMIC MEMORY: 19748 WORDS (75 PAGES)

ELAPSED TIME: 00:46:34

CZTKIB.BIN,CZTKIB/-SP/CR=SVC/ML,SPMACJ/ML,CZTKIB