

TK25

TK25 DATA RELIAB TST
CZTKIAR0

COPYRIGHT (c) 1984
AH-T284A-MC
FICHE 01 OF 01

JUL 1984
digital
Made In USA

1001 0001 1001
1001 0001 1001
1001 0001 1001
1001 0001 1001

.REM &

IDENTIFICATION

PRODUCT CODE: AC-T783A-MC

PRODUCT NAME: CZTKIAO TK25 DATA RELIABILITY TEST

PRODUCT DATE: 15 MARCH 1984

MAINTAINER: MAGTAPE DIAGNOSTIC ENGINEERING

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

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

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

COPYRIGHT (C) 1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBIS
DEC	DECUS	DECTAPE	

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

USER'S GUIDE

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

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

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

F1

PARAMETER CODING

MACRO M1200 21 MAR 84 16:45 PAGE 7

SEQ 0005

208
209
210
211
212
213
214
215
216

XST3

XXDP.

EXTENDED STATUS REGISTER 3

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

218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

1.1.1 FUNCTIONAL DESCRIPTION -

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

1.1.2 STRUCTURE OF PROGRAM

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

1.1.3 MEMORY MAP -

0	VECTOR AREA
400	
	STACK
2000	
	DIAGNOSTIC
32000	
	FREE MEMORY
41000	
	DIAGNOSTIC SUPERVISOR
72000	
	XXDP.
77777	

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

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

1.1.4 DIAGNOSTIC INFORMATION -

1.1.4.1 SCOPE -

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

THERE ARE 6 TESTS IN THIS PROGRAM:

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

1.1.4.2 ERROR RECOVERY -

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

1.1.4.3 WRITE ERROR RECOVERY -

THERE ARE 2 DISTINCT, SELECTABLE WRITE ERROR RECOVERY ALGORITHMS:

1. MEDIA/OPERATIONAL SELECTIVE ALGORITHM
2. OPERATIONAL ALGORITHM

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

TO SELECT THE SECOND ALGORITHM:

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

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

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

1.1.4.3.1 MEDIA/OPERATIONAL SELECTIVE WRITE ERROR RECOVERY ALGORITHM

SCOPE

THE ALGORITHM DISCERNS MEDIA RELATED WRITE ERRORS FROM
OPERATIONAL ONES.

ALGORITHM

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

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

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

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

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

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

BAD SPOTS REPORTS

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

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

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

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

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

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

IF TEST 2 IS HALTED, THEN RESTARTED OR CONTINUED, THE RECORD COUNT IS RESET TO ZERO AND THE BAD SPOT ID SHALL FOLLOW THAT RESET COUNT.

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

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

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

EXAMPLE OF A TAPE PASS PRINT:

CZTKIA SFT ERR 00009 ON UNIT 00 TST 002 SUB 000 PC: 012100
RECOVERABLE ERROR

WRT CMD FAILED - UNIT 0 PASS: 1 RECORD: 6

PREVIOUS CMD WAS WRT

CMDPKT TSBA RFC TSSR TCC

100205 002406 000000 100210 4

026600

000000

003107

XST0 XST1 XST2 XST3

000350 000002 100400 000000

SUSPECT BAD SPOT AFTER 1 RETRY, 2 REPEAT

SUSPECT BAD SPOT AFTER 2 RETRY, 1 REPEAT

SUSPECT BAD SPOT AFTER 3 RETRY, 1 REPEAT

SUSPECT BAD SPOT AFTER 4 RETRY, 3 REPEAT

RETRY FAILED ON BAD SPOT...ERASED!

SUSPECT BAD SPOT AFTER 1 RETRY, 1 REPEAT

CZTKIA SFT ERR 00009 ON UNIT 00 TST 002 SUB 000 PC: 012100
RECOVERABLE ERROR

WRT CMD FAILED - UNIT 0 PASS: 1 RECORD:10210

PREVIOUS CMD WAS WRT

CMDPKT TSBA RFC TSSR TCC

100205 002406 000000 100210 4

026600

000000

004000

XST0 XST1 XST2 XST3

000350 000002 100010 000000

RECOVERED ON RETRY # 1

↑C

DR>PRI

UNIT 0 PASS: 1 RECORD:10210

BYTES WRITTEN 0,272,279,691

BYTES READ REV 0,301,123,654

BYTES READ REV 0,301,120,381

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

2 BAD SPOTS THIS TAPE PASS PRECEDING RECORD #:

SPEC COND	HARD	FATAL	COMPARE
2	0	0	0

DR>

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

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

1.1.4.3.2 OPERATIONAL WRITE-ERROR-RECOVERY ALGORITHM

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

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

1.1.4.4 EARLY WARNING WRITE ERRORS -

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

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

1.1.4.5 DIAGNOSTIC TIMING ADJUSTMENT

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

532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586

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

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

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

1.2.2 SOFTWARE REQUIREMENTS -

- 0 DIAGNOSTIC SUPERVISOR

1.3 RELATED DOCUMENTS AND STANDARDS

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

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

ORDER OF HOST CPU DIAGNOSTIC USAGE:

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

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

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

1.5 ASSUMPTIONS

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

1.6 DIAGNOSTIC HISTORY

REVISION A 15 MAR 84 ORIGINAL RELEASE

2.0 OPERATING INSTRUCTIONS

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

2.1 HARDWARE PARAMETERS

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

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

TSSR ADDRESS (172522) ?

VECTOR (224) ?

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

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

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

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

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

2.2 SOFTWARE PARAMETERS

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

CLEAR COUNTERS (L) Y ?

RESET RANDOM VARIABLES (L) N ?

HALT AFTER EACH CMD (L) N ?

PRINT SOFT ERRORS (L) N ?

INHIBIT RECOVERY (L) N ?

BAD TAPE SPOT DETECT (L) Y ?

DISABLE INTERRUPTS (L) N ?

INHIBIT RFC ERROR REPORTS (L) N ?

CONTROL FR RAM DUMP (L) N ?

ENABLE EARLY WARNING MESSAGES (L) N ?

CHANGE CMD SEQUENCE (L) N ?

2.2.1 CLEAR COUNTERS

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

695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742

2.2.2 RESET RANDOM VARIABLES

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

2.2.3 PRINT SOFT ERRORS -

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

2.2.4 INHIBIT RECOVERY

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

2.2.5 BAD TAPE SPOT DETECTION -

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

2.2.6 DISABLE INTERRUPTS

YOU CAN DISABLE OR ENABLE CONTROLLER INTERRUPTS WITH THIS QUESTION.

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

744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771

2.2.8 CONTROLLER RAM DUMP .

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

2.2.9 ENABLE EARLY WARNING MESSAGES

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

2.2.10 CHANGE CMD SEQUENCE

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

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

773	
774	
775	
776	
777	CHARACTERISTICS CODE (0) 40 ? (0,20,40,200) (OCTAL)
778	CMD/2 (D) 13 ? (1 27) (DECIMAL)
779	BRF COUNT (D) 1 ? (1-4K) (DECIMAL)
780	# OF OPERATIONS (D) 1 ? (1 32K) (DECIMAL)
781	PATTERN (D) 7 ? (0-8) (DECIMAL)
782	CMD/3 (D) 4 ? (1-27) (DECIMAL)
783	BRF COUNT (D) 4096 ? (1-4K) (DECIMAL)
784	# OF OPERATIONS (D) 11719 ? (1-32K) (DECIMAL)
785	PATTERN (D) 7 ? (0 8) (DECIMAL)
786	CMD/4 (D) 7 ? (1 27) (DECIMAL)
787	BRF COUNT (D) 4096 ? (1-4K) (DECIMAL)
788	# OF OPERATIONS (D) 11719 ? (1-32K) (DECIMAL)
789	PATTERN (D) 7 ? (0-8) (DECIMAL)
790	CMD/5 (D) 2 ? (1-27) (DECIMAL)
791	BRF COUNT (D) 4096 ? (1-4K) (DECIMAL)
792	# OF OPERATIONS (D) 11719 ? (1-32K) (DECIMAL)
793	PATTERN (D) 7 ? (0-8) (DECIMAL)
794	CMD/6 (D) 13 ? (1-27) (DECIMAL)
795	BRF COUNT (D) 1 ? (1 4K) (DECIMAL)
796	# OF OPERATIONS (D) 1 ? (1 32K) (DECIMAL)
797	PATTERN (D) 7 ? (0 8) (DECIMAL)
798	CMD/7 (D) 27 ? (1-27) (DECIMAL)
799	BRF COUNT (D) 4096 ? (1-4K) (DECIMAL)
800	# OF OPERATIONS (D) 11719 ? (1-32K) (DECIMAL)
801	PATTERN (D) 7 ? (0-8) (DECIMAL)
802	CMD/8 (D) 27 ? (1-27) (DECIMAL)
803	BRF COUNT (D) 4096 ? (1 4K) (DECIMAL)
804	# OF OPERATIONS (D) 11719 ? (1-32K) (DECIMAL)
805	PATTERN (D) 7 ? (0 8) (DECIMAL)
806	
807	
808	
809	
810	
811	
812	
813	
814	
815	
816	

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.

2.2.11 COMMAND LIST FOR USE IN SOFTWARE DIALOGUE.

CODE	COMMAND	DESCRIPTION
1	DRI	DRIVE INITIATE.
2	RUF	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	DLT	DELAY 'N' MILLISECONDS WHERE N IS DEFINED IN THE NUMBER OF OPERATIONS. THIS DELAY IS USED BETWEEN EACH EXECUTABLE COMMAND.
27	END	END OF COMMAND SEQUENCE.

2.2.12 DATA PATTERN LIST FOR USE IN SOFTWARE DIALOGUE

PATTERN #	DESCRIPTION.
0	INCREMENTING PATTERN. 0 - 377.
1	ALL 1'S PATTERN.
2	ALL 0'S PATTERN.
3	1 BIT WALKING FROM R TO L IN A FIELD OF 0'S.
4	0 BIT WALKING FROM R TO L IF A FIELD OF 1'S.
5	ALTERNATING 1 AND 0 BITS WITH ALTERNATE BYTES COMPLIMENTED. (125/25
6	ALTERNATING BYTES OF 000 AND 377.
7	RANDOM DATA PATTERN.
8	NO PATTERN GENERATION.

2.3 EXAMPLES OF SOFTWARE DIALOGUE

2.3.1 BASIC FUNCTION AND DATA RELIABILITY WITH ALL ERROR REPORTING ENABLED

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

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

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

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

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

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

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

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

CHANGE SW (L) ?	Y<CR>	
CLEAR COUNTERS (L) N ?	Y<CR>	
RESET RANDOM VARIABLES (L) N ?	N<CR>	
HALT AFTER EACH CMD (L) N ?	N<CR>	
PRINT SOFT ERRORS (L) N ?	N<CR>	
INHIBIT RECOVERY (L) N ?	N<CR>	
BAD TAPE SPOT DETECT (L) Y ?	N<CR>	
DISABLE INTERRUPTS (L) N ?	Y<CR>	
INHIBIT RFC ERROR REPORT (L) N ?	Y<CR>	
CONTROLLER RAM DUMP (L) N ?	N<CR>	
ENABLE EARLY WARNING MESSAGES (L) N ?	N<CR>	
CHANGE CMD SEQUENCE (L) N ?	Y<CR>	
CHARACTERISTICS CODE (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 ?	<+2>	

960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007

2.4 EXECUTION TIMES

2.4.1 SYSTEM CONFIGURATION

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

2.4.2 TEST EXECUTION TIMES

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

NOTE

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

3.0 ERROR INFORMATION

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

3.1 ERROR REPORTING

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

ERROR #, TEST #, SUBTEST #, PROGRAM COUNTER, UNIT #, COMMAND, PREVIOUS COMMAND, PASS COUNT, NUMBER OF RECORDS FROM BOT, RECORD READ COUNT, THE COMMAND PACKET, TSSR, TCC, TSBA, RFC, AND THE EXTENDED STATUS REGISTERS.

STANDARD ERROR REPORT FORMAT:

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

* CAUTION *

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

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

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

EXAMPLE OF AN ERROR REPORT:

```

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

```

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

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

3.1.2 ERROR 2 - TK25 NOT READY:

BEFORE ANY COMMAND IS ISSUED TO THE TK25, THE SUBSYSTEM READY BIT IN THE TSSR4 IS CHECKED. IF THE SSR IS NOT SET, THE PROGRAM REPORTS THE NOT READY ERROR. THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST SEQUENCE UNLESS THE IDU OPTION IS USED.

3.1.3 ERROR 3 - NO RESPONSE ERROR:

ONCE THE TSDB IS LOADED, THE TK25 HAS ONE MILLISECOND TO RESPOND OR THE PROGRAM REPORTS A "NO RESPONSE" ERROR. THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST SEQUENCE UNLESS THE IDU OPTION IS USED.

3.1.4 ERROR 4 - NO INTERRUPT ERROR:

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

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

3.1.5 SPECIAL CONDITION ERRORS: -

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

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

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

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

3.1.5.2 ERROR 6 TERMINATION CLASS CODE 1, ATTENTION CONDITION

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

3.1.5.3 ERROR 7 TERMINATION CLASS CODE 2, TAPE STATUS ALERT

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

3.1.5.4 ERROR 8 TERMINATION CLASS CODE 3, FUNCTION REJECT

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

1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205

3.1.5.5 ERROR 9 - TERMINATION CLASS CODE 4, RECOVERABLE ERROR

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

3.1.5.6 ERROR 10 - TERMINATION CLASS CODE 5, RECOVERABLE ERROR

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

3.1.5.7 ERROR 11 TERMINATION CLASS CODE 6, UNRECOVERABLE ERROR

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

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

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

3.1.6 ERROR 13 - RFC NON-ZERO ERROR:

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

1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252

3.1.7 ERROR 14 - RETRY LIMIT EXCEEDED: -

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

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

3.1.8 ERROR 15 - TOO MANY INTERRUPTS:

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

3.1.9 ERROR 16 - CAPSTAN RUNAWAY:

CAPSTAN DID NOT STOP WITHIN ACCEPTABLE WINDOW AFTER LAST COMMAND. THIS IS A FATAL DEVICE ERROR AND THE DEVICE WILL BE DROPPED FROM THE TEST CYCLE UNLESS THE IDU OPTION IS USED.

3.1.10 ERROR 17 - DATA COMPARE ERROR:

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

3.2 ERROR HALTS

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

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

4.0 PERFORMANCE REPORT

```

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

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

SPEC COND   HARD   FATAL   COMPARE
          XXXXX   XXXXX   XXXXX   XXXXX

```

5.0 TEST SUMMARIES

5.1 TEST 1 - BASIC FUNCTIONS.

EXECUTES AND VERIFIES CORRECT COMPLETION OF ALL TK25 FUNCTIONS.

```

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

SUBTEST 2   REWIND.
            • REWIND.
            • REWIND AT BOT.

SUBTEST 3   WRITE/VERIFY.
            • WRITE/VERIFY PATTERN 1.
            • WRITE/VERIFY PATTERN 2.
            • WRITE/VERIFY PATTERN 3.
            • WRITE/VERIFY PATTERN 4.
            • WRITE/VERIFY PATTERN 5.
            • WRITE/VERIFY PATTERN 6.
            • WRITE/VERIFY PATTERN 0.

SUBTEST 4   WRITE TAPE MARK, ERASE.
            • WRITE TAPE MARK.
            • WRITE 10 RECORDS
            • ERASE 10 TIMES
            • WRITE TAPE MARK.
            • WRITE TAPE MARK RETRY.

SUBTEST 5   SPACE FILES.
            • SPACE 2 FILES REVERSE.

```

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

- SPACE 2 FILES FORWARD.
- SPACE 2 FILES REVERSE.
- SPACE 2 FILES FORWARD.

SUBTEST 6 SPACE RECORDS.

- REWIND.
- SPACE 7 RECORDS FORWARD.
- SPACE 7 RECORDS REVERSE.
- SPACE 7 RECORDS FORWARD.
- SPACE 7 RECORDS REVERSE.

SUBTEST 7 - WRITE RETRY.

- REWIND.
- WRITE DATA.
- WRITE RETRY.

SUBTEST 8 - READ REV RETRY.

- READ REVERSE.
- READ NEXT REVERSE.
- READ NEXT FORWARD.

SUBTEST 9 READ FWD RETRY.

- READ FORWARD.
- READ PREVIOUS FORWARD.
- READ PREVIOUS REVERSE.

SUBTEST 10 CLEAN.

- CLEAN.
- REWIND.

SUBTEST 11 - WRITE/VERIFY SWAPPED DATA BYTES.

- WRITE/VERIFY EVEN LENGTH (RECORD 1).
- WRITE/VERIFY ODD LENGTH (RECORD 2).
- SET DATA BYTE SWAP.
- WRITE/VERIFY EVEN LENGTH (RECORD 3).
- WRITE/VERIFY ODD LENGTH (RECORD 4).
- CLEAR DATA BYTE SWAP.

SUBTEST 12 - READ SWAPPED DATA BYTES.

- READ REV RECORD 4.
- READ REV RECORD 3.
- SET DATA BYTE SWAP.
- READ REV RECORD 2.
- READ REV RECORD 1.
- READ FWD RECORD 1.
- READ FWD RECORD 2.
- CLEAR DATA BYTE SWAP.
- READ FWD RECORD 3.
- READ FWD RECORD 4.

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

5.2 TEST 2 - DATA RELIABILITY.

1. THE TAPE IS INITIATED WITH THE FOLLOWING COMMANDS:
SET CHARACTERISTIC 40
REWIND
WRITE 31 RECORDS OF RANDOM LENGTH AND DATA
2. WRITE, REPOSITION RECORD REVERSE, AND READ COMMANDS ARE
SELECTED AT RANDOM AND ARE EXECUTED A RANDOM NUMBER OF
TIMES WITH RANDOM LENGTHS AND RANDOM PATTERN UNTIL END
OF TAPE IS REACHED.
3. AT THE END OF EACH PASS, A REWIND COMMAND IS ISSUED AND
A PERFORMANCE REPORT IS PRINTED.

NOTE

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

5.3 TEST 3 - WRITE AND READ STREAMING TEST.

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

1. REWINDS ALL UNITS, THEN ON EACH UNIT:

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

1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462

NOTE

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

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

5.4 TEST 4 - WRITE COMPATABILITY/WRITE UTILITY.

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

5.5 TEST 5 READ COMPATABILITY/READ UTILITY.

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

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

5.6 TEST 6 - EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.

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

6.0 DEVICE INFORMATION TABLES

6.1 GENERAL

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

COMMANDS ARE NOT WRITTEN TO THE DRIVE; RATHER, COMMAND POINTERS ARE WRITTEN WHICH POINT TO COMMAND PACKETS SOMEWHERE IN CPU MEMORY. THE COMMAND POINTER IS USED BY THE TK25 SUBSYSTEM TO FETCH THE WORD(S) WITHIN THE COMMAND PACKET. THE WORDS WITHIN THE COMMAND PACKET ARE:

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

THE TSSR CONTAINS ALL THE INFORMATION WHICH WILL BE NECESSARY TO DETERMINE WHETHER:

1. THE DRIVE IS READY TO ACCEPT ANOTHER COMMAND.
2. THE PREVIOUS COMMAND WAS EXECUTED WITHOUT ERROR.

IF EITHER OF THE ABOVE CONDITIONS IS UNTRUE AT "JOB DONE" OR "COMMAND INITIATION" TIME, IT MAY BE NECESSARY TO GET THE EXTENDED STATUS REGISTERS TO DETERMINE WHAT ACTION IS TO BE TAKEN AND/OR LOG THE ERROR INFORMATION.

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

THE TSDB MUST BE WRITTEN WITH A DATO INSTRUCTION TO PROPERLY WRITE THE COMMAND POINTER. A DATOB WILL CAUSE A MAINTENANCE FUNCTION. A DATO TO THE TSSR WILL CAUSE SUBSYSTEM INIT.

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

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

6.2 BUS INTERFACE SPECIFICATIONS

THE STANDARD ADDRESSES AND VECTORS ARE ASSIGNED AS FOLLOWS:

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

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

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)	)OML)	)IE)	)VCK)	)	)WLK)	)BOT)	)EOT)
XST1	)DLT)	)	)	)CRS)	)NER)	)	)	)TN3)	)TN2)	)TN1)	)TN0)	)EW)	)	)UNC)	)	)
XST2	)OPM)	)DCF)	)DMF)	)SPD)	)0)	)1)	)	)	)	)	)	)	)	)	)	)
XST3	)	)	)	)	)	)	)	)	)	)	)	)	)	)	)	)

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

6.3.2 TK25 STATUS REGISTER (TSSR) -

6.3.2.1 TSSR READ ONLY -

```

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

```

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

1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677

BIT	NAME	TERMINATION CLASS	DEFINITION
07	SSR	S	SUBSYSTEM READY: THE SUBSYSTEM IS NOT BUSY AND IS READY TO ACCEPT A NEW COMMAND POINTER WHENEVER THIS BIT IS SET.
06	OFL	S	OFF LINE: INDICATES THE TAPE TRANSPORT IS OFF LINE AND UNAVAILABLE FOR ANY MOTION COMMANDS.
05	FC1		FATAL TERMINATION CLASS 1 - NOT USED
04	FC0	-	FATAL TERMINATION CLASS 0 - NOT USED
03	TC2	S	TERMINATION CLASS BIT 2: THIS BIT, ALONG WITH TC1 AND 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

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

6.3.2.2 TSSR WRITE ONLY

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

1. A WRITE WORD (DAT0) TO THE TSSR WILL INITIALIZE THE TK25 SUBSYSTEM AND REWIND THE TAPE ON THE CARTRIDGE.
2. A WRITE BYTE (DAT0B) 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):
 - O REWIND THE TAPE
 - O SPACE FORWARD ONE RECORD
 - O READ THE FIRST 256 BITS OF THE SECOND RECORD INTO CPU MEMORY STARTING AT ADDRESS 0.

1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767

6.3.3 EXTENDED STATUS REGISTER 0 (XSTAT0) -

```

      15   14   13   12   11   10   09   08   07   06   05   04   03   02   01   00
XST0)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

```

1769
1770
1771
1772
1773 }
1774 }
1775 }
1776 }
1777 }
1778 }
1779 }
1780 } 09 ILC 3
1781 }
1782 }
1783 }
1784 }
1785 } 08 ILA 3
1786 }
1787 }
1788 } 07 MOT S
1789 }
1790 }
1791 } 06 ONL 5/1/3
1792 }
1793 }
1794 }
1795 }
1796 }
1797 } 05 IE S
1798 }
1799 } 04 VCK S/3
1800 }
1801 }
1802 }
1803 }
1804 } 03 PED S
1805 }
1806 }
1807 } 02 WLK S/3
1808 }
1809 }
1810 } 01 BOT S/2/3
1811 }
1812 }
1813 }
1814 }
1815 } 00 EOT S/2
1816 }
1817 }
1818 }
1819 }
1820 }
1821 }

```

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

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

6.3.4 EXTENDED STATUS REGISTER 1 (XSTAT1) -

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

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 0 IS A 1, THE CONTROLLER ASSUMES A 22 BIT Q BUS, AND IF 0, AN 18 BIT Q BUS OR A UNIBUS. IN THE LATTER CASE, COMMAND PACKETS CONTAINING ADDRESSES GREATER THAN 18 BITS WILL GENERATE ERRORS.

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

6.3.6 EXTENDED STATUS REGISTER 3 (XSTAT3)

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

BIT	NAME	TERMINATION CLASS	DEFINITION
15 08 EAD 15 08		5	ERROR ADDRESS MOST SIGNIFICANT BYTE. IF XSTAT 1 BIT 14 IS 1, THE ADDRESS IS THE PROGRAM COUNTER IN THE SIA, AND IF 0 IN THE TCS.
07			NOT USED.
06	OPI	6	OPERATION INCOMPLETE: SETS IF 16 FEET OF TAPE WITHOUT DATA PASSES THE READ HEAD ON READ, SPACE, OR SKIP OPERATIONS OR 4 FEET OF TAPE IN WRITE OPERATIONS.
05	REV	5	REVERSE: INDICATES REVERSE IS THE DIRECTION OF CURRENT TAPE OPERATION.
04	TCH	7	NO TACHS: INDICATES THAT CAPSTAN MOTOR DID NOT START OR THAT TACHS ARE NOT BEING GENERATED OR DETECTED.
03	STP	5/6	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.

```

1986      .TITLE PROGRAM HEADER AND TABLES
1987      .SBTTL PROGRAM HEADER
2018
2020 000000      .ENABL ABS,AMA
2021      .      2000
2023 002000      BGNMOD
2024
2025      ***
2026      ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
2027      ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
2028      ;
2029
2030 002000      POINTER BGNRPT,BGNSW,BGNSFT,BGNAU,BGNDU,BGNSETUP
2031
2039
2040 002000      HEADER CZTKI,A,0,5000,1,INTPRI
002000      L$NAME::      ;DIAGNOSTIC NAME
002000      103
002001      132
002002      124
002003      113
002004      111
002005      000
002006      000
002007      000
002010      L$REV::      ;REVISION LEVEL
002010      101
002011      L$DEPO::      ;0
002011      060
002012      L$UNIT::      ;NUMBER OF UNITS
002012 000001
002014      L$TIML::      ;LONGEST TEST TIME
002014 005000
002016      L$MPCP::      ;POINTER TO H.W. QUES.
002016 027750
002020      L$SPCP::      ;POINTER TO S.W. QUES.
002020 030022
002022      L$MPTP::      ;PTR. TO DEF. H.W. PTABLE
002022 002176
002024      L$SPTP::      ;PTR. TO S.W. PTABLE
002024 002204
002026      L$LADP::      ;DIAG. END ADDRESS
002026 031510
002030      L$STA::      ;RESERVED FOR APT STATS
002030 000000
002032      L$CO::
002032 000000
002034      L$DTYP::      ;DIAGNOSTIC TYPE
002034 000001
002036      L$APT::      ;APT EXPANSION
002036 000000
002040      L$DTP::      ;PTR. TO DISPATCH TABLE
002040 002124
002042      L$PRIO::      ;DIAGNOSTIC RUN PRIORITY
002042 000340
002044      L$ENVI::      ;FLAGS DESCRIBE HOW IT WAS SETUP
002044 000000

```

```

.ASCII /C/
.ASCII /Z/
.ASCII /T/
.ASCII /K/
.ASCII /I/
.BYTE 0
.BYTE 0
.BYTE 0
.ASCII /A/
.ASCII /O/
.WORD T$PTHV
.WORD 5000
.WORD L$HARD
.WORD L$SOFT
.WORD L$HW
.WORD L$SW
.WORD L$LAST
.WORD 0
.WORD 0
.WORD 1
.WORD 0
.WORD L$DISPATCH
.WORD INTPRI
.WORD 0

```

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

```

2049      .SBTTL DISPATCH TABLE
2050
2051      ;**
2052      ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
2053      ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
2054      ;
2055
2056      DISPATCH 6          ; SIX TESTS
2057      002122      000006          .WORD 6
2058      002122      000006
2059      002124      024004          .WORD T1
2060      002124      024004
2061      002126      025372          .WORD T2
2062      002126      025372
2063      002130      026046          .WORD T3
2064      002130      026046
2065      002132      026524          .WORD T4
2066      002132      026524
2067      002134      026670          .WORD T5
2068      002134      026670
2069      002136      027002          .WORD T6
2070      002136      027002
2071
2072      .SBTTL DESCRIPTIVE TEXT
2073
2074      ;**
2075      ; 2 LINES OF TEXT PRINTED TO THE OPERATOR TO IDENTIFY THE DIAGNOSTIC AND THE DEVICE UNDER TEST
2076      ;**
2077
2078      DESCRIPT      <DATA RELIABILITY TEST>
2079      L$DESC::
2080      002140      104      101      124          .ASCIZ /DATA RELIABILITY TE
2081      002140      104      101      124
2082      002143      101      040      122
2083      002143      101      040      122
2084      002146      105      114      111
2085      002146      105      114      111
2086      002151      101      102      111
2087      002151      101      102      111
2088      002154      114      111      124
2089      002154      114      111      124
2090      002157      131      040      124
2091      002157      131      040      124
2092      002162      105      123      124
2093      002162      105      123      124
2094      002165      000
2095
2096      DEVTYP      <TK25>
2097      L$DVTYP::
2098      002166      124      113      062          .ASCIZ /TK25/
2099      002166      124      113      062
2100      002171      065      000
2101
2102      .EVEN
2103
2104      .ASCIZ /TK25/
2105
2106      .EVEN

```

PROGRAM HEADER AND TABLES
DEFAULT HARDWARE P TABLE

MACRO M1200 21 MAR 84 16:45 PAGE 48

SEQ 0045

```
2075      .SBTTL  DEFAULT HARDWARE P-TABLE
2076
2077      ;**
2078      ; THE DEFAULT HARDWARE P TABLE CONTAINS DEFAULT VALUES OF
2079      ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
2080      ; IS IDENTICAL TO THE STRUCTURE OF THE RUN TIME P TABLE.
2081      ; -
2082
2083      BGNHW  DFPTBL
2084
2085      L$HW::
2086      DFPTBL::
2087
2088      172522      ;TSSR ADDRESS.
2089      224         ;VECTOR ADDRESS.
2090
2091      002176      172522
2092      002200      000224
2093
2094      002202      ENDNHW
2095      002202
2096
2097      .WORD  L10000-L$HW/2
```

```

2097          .SBTTL  SOFTWARE P-TABLE
2098
2099          ;**
2100          ; THE SOFTWARE P TABLE CONTAINS THE VALUES OF THE PROGRAM
2101          ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
2102          ;
2103          BGNSW  SFPTBL
2104          000043          .WORD  L10001-L$SW/2
2105          L$SW::
2106          SFPTBL::
2107          CLRFLG::.BYTE  1          ;CLEAR COUNTERS FLAG.
2108          RRANV::.BYTE  0          ;RESET RANDOM VARIABLES EACH PASS FLAG.
2109          HAE::.BYTE  0          ;HALT AFTER EACH COMMAND FLAG.
2110          ERCVER::.BYTE  0          ;ENABLE RECOVERABLE ERROR PRINTS FLAG.
2111          IREC::.BYTE  0          ;INHIBIT ERROR RECOVERY FLAG.
2112          BADTSW::.BYTE  1          ;BAD TAPE SWITCH TO REWRITE ON SAME SPOT & DETECT BAD TAPE
2113          DINT::.BYTE  0          ;DISABLE INTERRUPTS FLAG.
2114          PIRE::.BYTE  0          ;INHIBIT RESIDUAL FRAMECOUNT ERROR REPORT FLAG.
2115          RAMWRT::.BYTE  0          ;ENABLE OPTIONAL RAM DUMP
2116          .BYTE  0          ;SPARE
2117          EWPRT::.BYTE  0          ;ENABLE EARLY WARNING END OF TRACK STATUS PRINTS
2118          CHGFLG::.BYTE  0          ;CHANGE CMD SEQ TABLE FLAG.
2119          .EVEN
2120          CHAR::  CH.EAI          ;CHARACTERISTICS CODE (DEFAULT = 40).
2121          CMDD::.WORD  13.          ;COMMAND 2 (DEFAULT = REWIND).
2122          .WORD  1          ;BYTE COUNT
2123          .WORD  1          ;NUMBER OF OPERATIONS
2124          .WORD  RANP          ;RANDOM DATA
2125          .WORD  4          ;COMMAND 3 (DEFAULT = WRITE)
2126          .WORD  DATCNT          ;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
2127          .WORD  15000.          ;NUMBER OF OPERATIONS (DEFAULT = 15000).
2128          .WORD  RANP          ;RANDOM DATA
2129          .WORD  13.          ;COMMAND 4 (DEFAULT = REWIND)
2130          .WORD  1          ;BYTE COUNT
2131          .WORD  1          ;ONE OPERATION
2132          .WORD  RANP          ;RANDOM DATA
2133          .WORD  2          ;COMMAND 5 (DEFAULT = READ FWD).
2134          .WORD  DATCNT          ;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
2135          .WORD  15000.          ;NUMBER OF OPERATIONS (DEFAULT = 15000)
2136          .WORD  RANP          ;RANDOM DATA
2137          .WORD  27.          ;TERMINATOR
2138          .WORD  1          ;BYTE COUNT
2139          .WORD  1          ;NUMBER OF OPERATIONS
2140          .WORD  RANP          ;PATTERN
2141          .WORD  27.          ;END OF CMD SEQ TABLE CODE (DEF) OR CMD 7
2142          .WORD  DATCNT          ;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
2143          .WORD  15000.          ;NUMBER OF OPERATIONS (DEFAULT = 15000).
2144          .WORD  RANP          ;RANDOM DATA
2145          .WORD  27.          ;END OF CMD SEQ TABLE CODE (DEF) OR CMD 8
2146          .WORD  DATCNT          ;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
2147          .WORD  15000.          ;NUMBER OF OPERATIONS (DEFAULT = 15000).
2148          .WORD  RANP          ;RANDOM DATA
2149          .WORD  27.          ;END OF CMD SEQ TABLE CODE (DEF) OR CMD 8
2150          .WORD  DATCNT          ;BYTE COUNT (DEFAULT = MAX BUFFER SIZE).
2151          .WORD  15000.          ;NUMBER OF OPERATIONS (DEFAULT = 15000).
2152          .WORD  RANP          ;RANDOM DATA
2153          ENDSW
2154          L10001:
2155          ENDMOD

```

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

.TITLE GLOBAL AREAS
.SBTTL GLOBAL EQUATES SECTION

BGNMOD

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

EQUALS

; BIT DIFINITIONS

100000	BIT15== 100000
040000	BIT14== 40000
020000	BIT13== 20000
010000	BIT12== 10000
004000	BIT11== 4000
002000	BIT10== 2000
001000	BIT09== 1000
000400	BIT08== 400
000200	BIT07== 200
000100	BIT06== 100
000040	BIT05== 40
000020	BIT04== 20
000010	BIT03== 10
000004	BIT02== 4
000002	BIT01== 2
000001	BIT00== 1

001000	BIT9== BIT09
000400	BIT8== BIT08
000200	BIT7== BIT07
000100	BIT6== BIT06
000040	BIT5== BIT05
000020	BIT4== BIT04
000010	BIT3== BIT03
000004	BIT2== BIT02
000002	BIT1== BIT01
000001	BIT0== BIT00

; EVENT FLAG DEFINITIONS

; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

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

; PRIORITY LEVEL DEFINITIONS

000340	PRI07== 340
000300	PRI06== 300

GLOBAL AREAS MACRO M1200 21 MAR 84 16:45 PAGE 5: 1
GLOBAL EQUATES SECTION

SEQ 0048

```

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      MOE== 100000

```

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

```

; REGISTER USAGE.
;
; R0 - PASSES PARAMETERS TO/FROM DIAGNOSTIC SUPERVISOR.
; R1 - COMMAND SEQUENCE TABLE POINTER.
; R2 - GENERAL PURPOSE REGISTER.
; R3 - GENERAL PURPOSE REGISTER.
; R4 - GENERAL PURPOSE REGISTER.
; R5 - CURRENT LOGICAL DEVICE NUMBER x 2.
; R6 - STACK POINTER.
; R7 - PROGRAM COUNTER.
;
; THE FOLLOWING ARE BIT DEFINITIONS FOR THE TSSR REGISTERS.
;
TS.SC==100000      ;SPECIAL CONDITION BIT.
TS.UPE==40000      ;UNIBUS PARITY ERROR
TS.SPE==20000      ;SERIAL BUS PARITY ERROR.
TS.RMR==10000      ;REGISTER MODIFICATION REFUSED.
TS.NXM==4000       ;NON EXISTENT MEMORY.
TS.NBA==2000       ;NEED BUFFER ADDRESS.
TS.A17==1000       ;BUS ADDRESS BIT 17.
TS.A16==400        ;BUS ADDRESS BIT 16.
TS.SSR==200        ;UNIT READY BIT.
TS.OFL==100        ;OFF LINE.
TSC.FCC==177717    ;FATAL CLASS CODE MASK.
TSC.TCC==177761    ;TERMINATION CLASS CODE MASK.

```

```

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
177717
177761

```

```

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

```

```

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

```

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

```

2393 .SBTTL GLOBAL DATA SECTION
2394
2395 ***
2396 ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
2397 ; IN MORE THAN ONE TEST.
2398 ;
2399
2400 ; COMMAND PACKET.
2401
2402 .          *          .+3E177774      ;MUST BE ON MOD 4 BOUNDARY.
2403 CMDPKT:: 0          ;1ST WORD IS TK25 COMMAND.
2404          0          ;2ND WORD IS THE BUFFER LOW ADDRESS.
2405          0          ;3RD WORD IS THE BUFFER HIGH ADDRESS.
2406          0          ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.
2407
2408 ; GET STATUS COMMAND PACKET.
2409
2410 .          *          .+3E177774      ;MUST BE ON MOD 4 BOUNDARY.
2411 GSCPK:: .WORD      GES
2412
2413 ; MESSAGE BUFFER RELEASE COMMAND PACKET.
2414
2415 .          *          .+3E177774      ;MUST BE ON MOD 4 BOUNDARY.
2416 BRCPK:: .WORD      MBR
2417
2418 ; REWIND COMMAND PACKET (USED IN ERROR RECOVERY ONLY)
2419
2420 .          *          .+3E177774      ;MUST BE ON A MODULE 4 BOUNDARY.
2421 RWCpk:: .WORD      RWD
2422          .WORD      1
2423
2424 ; WORK AREA FOR ANALYSIS OF MESSAGE PACKET CONTENTS.
2425
2426 MSGPKT:: .BLKW      7          ;1ST WORD:: MESSAGE TYPE.
2427          ;2ND WORD:: DATA FIELD LENGTH.
2428          ;3RD WORD:: RESIDUAL FRAME COUNT.
2429          ;4TH WORD:: XSTAT0
2430          ;5TH WORD:: XSTAT1
2431          ;6TH WORD:: XSTAT2
2432          ;7TH WORD:: XSTAT3
2433
2434 ; MESSAGE PACKETS.
2435
2436 MSGPK0:: .BLKW      7          ;MESSAGE PACKET FOR DEVICE #0
2437 MSGPK1:: .BLKW      7          ;MESSAGE PACKET FOR DEVICE #1
2438 MSGPK2:: .BLKW      7          ;MESSAGE PACKET FOR DEVICE #2
2439 MSGPK3:: .BLKW      7          ;MESSAGE PACKET FOR DEVICE #3
2440

```

```

2442      ; SET CHARACTERISTIC BLOCK.
2443
2444 002446 002356  SCHBK:: MSGPK0      ;1ST WORD:: MSGPKT ADDR LO(SET UP BY EXECUTE ROUTINE).
2445 002450 000000      0      ;2ND WORD:: MSGPKT ADDR HI.
2446 002452 000016      MSGCNT      ;3RD WORD:: MSG BUFFER LENGTH (BYTES)
2447 002454 000040      CM.EAI      ;4TH WORD:: CHARACTERISTICS WORD(SET BY SETUP ROUTINE).
2448
2449      ; TK25 REGISTER ADDRESSES.
2450
2451 002456      TSDB:: .BLKW 4      ;TK25 DATA BUFFER ADDRESSES.
2452 002466      TSSR:: .BLKW 4      ;TK25 STATUS REGISTER ADDRESSES.
2453 002476      TSVCT:: .BLKW 4      ;TK25 VECTOR ADDRESSES.
2454      002456      TSBA=TSDB      ;DATA BUFFER ADDRESS REGISTER.
2455
2456      ; ADDRESSES OF MESSAGE PACKETS.
2457
2458 002506 002356  MSGPKA:: MSGPK0      ;DEVICE 0.
2459 002510 002374      MSGPK1      ;DEVICE 1.
2460 002512 002416      MSGPK2      ;DEVICE 2.
2461 002514 002430      MSGPK3      ;DEVICE 3.
2462
2463      ; ADDRESSES OF INTERRUPT HANDLING ROUTINES.
2464
2465 002516 006552  TS4INT:: TS4IN0      ;DEVICE 0.
2466 002520 006560      TS4IN1      ;DEVICE 1.
2467 002522 006566      TS4IN2      ;DEVICE 2.
2468 002524 006574      TS4IN3      ;DEVICE 3.
2469
2470      ; TK25 CODE LEVELS, WILL BE STORED AFTER SCH CMD IN BASIC FUNCTION TEST
2471
2472 002526 000000  TS4CL:: 0      ;DEVICE 0
2473 002530 000000      0      ;DEVICE 1
2474 002532 000000      0      ;DEVICE 2
2475 002534 000000      0      ;DEVICE 3
2476
2477      ; UNIT NUMBERS OF ALL DEVICES BEING TESTED(1 4).
2478      ; WHEN DEVICE IS NOT IN USE, IT'S LOCATION WILL = 3.
2479      ; R5 WILL ALWAYS CONTAIN THE PRESENT LOGICAL UNIT NUMBER x 2.
2480
2481 002536 177774  DEVTBL:: .WORD NINUSE
2482 002540 177774      .WORD NINUSE
2483 002542 177774      .WORD NINUSE
2484 002544 177774      .WORD NINUSE
2485 002546 177777      .WORD END
2486
2487      ; BAD TAPE TABLE POINTER: USED BY WRITE RETRY ROUTINE
2488      ; 'WRTY' TO LOG BAD TAPE SPOTS ON UNITS UNDER TEST
2489
2490 002550 003000  BTADDR:: BT0
2491 002552 003052      BT1
2492 002554 003124      BT2
2493 002556 003176      BT3

```

C5

GLOBAL AREAS MACRO M1200 21 MAR 84 16:45 PAGE 55
GLOBAL DATA SECTION

SEQ 0054

```

2495      1      COUNTER AREA.
2496
2497      002560      CNTBGN=.
2498      002560      WRBC:: .BLKW 20      ;BYTES WRITTEN.
2499      002620      RRBC:: .BLKW 20      ;BYTES READ REV.
2500      002660      RFBC:: .BLKW 20      ;BYTES READ FWD.
2501      002720      WRREC:: .BLKW 4      ;RECOVERABLE WRITE ERRORS.
2502      002730      WRUNR:: .BLKW 4      ;UNRECOVERABLE WRITE ERRORS.
2503      002740      RRREC:: .BLKW 4      ;RECOVERABLE READ REV ERRORS.
2504      002750      RRUNR:: .BLKW 4      ;UNRECOVERABLE READ REV ERRORS.
2505      002760      RFREC:: .BLKW 4      ;RECOVERABLE READ FWD ERRORS.
2506      002770      RFUNR:: .BLKW 4      ;UNRECOVERABLE READ FWD ERRORS.
2507      003000      BT0:: .BLKW 21.      ;UNIT 0 BAT TAPE SPOTS LOG
2508      003052      BT1:: .BLKW 21.      ;UNIT 1 BAT TAPE SPOTS LOG
2509      003124      BT2:: .BLKW 21.      ;UNIT 2 BAT TAPE SPOTS LOG
2510      003176      BT3:: .BLKW 21.      ;UNIT 3 BAT TAPE SPOTS LOG
2511      003250      WRTYCT:: .BLKW 4      ;WRITE RETRY COUNTER
2512      003260      PASCNT:: .BLKW 4      ;PASS COUNT.
2513      003270      SCCNT:: .BLKW 4      ;SPECIAL CONDITION COUNT.
2514      003300      VFYCNT:: .BLKW 4      ;COUNT OF TK25 DATA COMPARE ERRORS.
2515      003310      HRDCNT:: .BLKW 4      ;COUNT OF HARD ERRORS.
2516      003320      FTLCNT:: .BLKW 4      ;COUNT OF FATAL ERRORS.
2517      003330      CNTEND=.      ;END OF STATISTICAL COUNTERS.
2518      003330      RECcnt:: .BLKW 4      ;NUMBER OF RECORDS FROM BOT: CLEARED ON REWIND
2519      000550      CNTLEN= CNTEND-CNTBGN      ;AND WHEN RESTARTING OR CONTINUING TEST 2.
2520      000550      ;LENGTH OF STATISTICAL COUNTER AREA.
2521
2522      003340      000      HERE: .BYTE 0      ;TEST 3 ASCII SEMAPHORE
2523      ;      THE FOLLOWING ARE THE DEFINITIONS OF VARIABLES
2524      ;      USED BY THE PROGRAM.
2525
2526      .EVEN
2527      003342      000000      RAMHLD: .WORD 0      ;RAM ADDR HOLDER 1ST ADDRESS
2528      003344      000000      RAMRSH: .WORD 0      ;HOLDS RS FOR LATER
2529      003346      000000      RAMDATA:: .BLKW 16.      ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
2530      003406      000000      RAMSIZ:: .WORD 0      ;RAM DATA SIZE FOR PRAMPKT ROUTINE
2531      003410      000000      CMPDAT:: .WORD 0      ;COUNTS # OF READS (TEST 3) BEFORE ALLOWING A DATA COMPARE
2532      003412      000003      DATRAT:: .WORD 3      ;CONTROLS THE DATA COMPARE RATIO
2533      003414      000000      DATAWT:: .WORD 0      ;WRITE BUFFER ADDRESS.
2534      003416      000000      DATARD:: .WORD 0      ;READ BUFFER ADDRESS.
2535      003420      000000      NCNT:: .WORD 0      ;STORAGE FOR VALUE OF N.
2536      003422      000000      NCNT1:: .WORD 0      ;TEMP STORAGE FOR VALUE OF N.
2537      003424      000000      BRFCNT:: .WORD 0      ;STORAGE FOR BPCR VALUE.
2538      003426      177777      CMDWRD:: .WORD END      ;CONTAINS COMMAND WORD BEING EXECUTED PRESENTLY.
2539      003430      177777      CMDSAV:: .WORD END      ;SAVE LOCATION FOR CMD WORD DURING ERROR RECOVERY
2540      003432      177777      PCMDWO:: .WORD END      ;CONTAINS PREVIOUS COMMAND WORD.
2541      003434      000000      CMDLG:: .WORD 0      ;CURRENT COMMAND LOGGING CODE.
2542      003436      000000      LENMSK:: .WORD 0      ;RANDOM WRITE LENGTH MASK, TO BE SET UP BY TESTS
2543      003440      153624      RANB:: .WORD 153624      ;RANDOM # GENERATOR BASE.
2544      003442      032561      RANS:: .WORD 32561      ;RANDOM # SAVE LOCATION.
2545      003444      000000      TIME1:: .WORD 0      ;TIME COUNT 1.
2546      003446      000000      TIME2:: .WORD 0      ;TIME COUNT 2.
2547      003450      000000      JLOOP:: .WORD 0      ;JMP COMMAND LOOP COUNT.
2548      003452      000000      JLOC:: .WORD 0      ;JMP COMMAND LOCATION COUNT.
2549      003454      000000      PATERN:: .WORD 0      ;PATTERN SELECT CODE.

```

```

2551 003456 000000 CTCC:: .WORD 0 ;CURRENT TERMINATION CLASS CODE.
2552 003460 000000 R5SAVE:: .WORD 0 ;LOCATION FOR SAVING CURRENT DEVICE POINTER.
2553 003462 000000 TSSREG:: .WORD 0 ;CURRENT STATUS REGISTER.
2554 003414 DIABLK=DATAWT ;WRITE BUFFER ALSO USED FOR DIAG CMD.
2555
2556 ; ERROR FLAG AREA, THESE FLAGS ARE CLEARED DURING INITIALIZATION AND
2557 ; AFTER EACH COMMAND IS COMPLETED.
2558
2559 003464 BGNFLG=.
2560 003464 000000 RETRYC:: .WORD 0 ;# OF RECOVERY ATTEMPTS EXECUTED.
2561 003466 000 RPTCNT:: .BYTE 0 ;WRITE REPEAT ON SAME SPOT C/IR: 4 PER WRITE RETRY
2562 003467 000 WRTYFG:: .BYTE 0 ;WRITE RETRY ON SAME SPOT IN PROGRESS FLAG
2563 003470 000 WRTYER:: .BYTE 0 ;WRITE RETRY ON SAME SPOT ERROR FLAG
2564 003471 000 RECLOG:: .BYTE 0 ;RECORD COUNT HAS BEEN UPDATED FOR THIS RECORD.
2565 003472 000 ERLOG:: .BYTE 0 ;DATA BYTES AND ERRORS HAVE BEEN LOGGED FOR THIS RECORD.
2566 003473 000 RWERR:: .BYTE 0 ;READ/WRITE ERROR HAS OCCURED.
2567 003474 000 UNREC:: .BYTE 0 ;UNRECOVERABLE ERPOR HAS OCCURED.
2568 003475 000 ERRREC:: .BYTE 0 ;ERROR RECOVERY MODE.
2569 .EVEN
2570 003476 ENDERF=.
2571
2572 ; ADDITIONAL FLAGS, THESE FLAGS ARE CLEARED DURING INITIALIZATION.
2573
2574 003476 INTFLG:: .BLKW 4 ;INTERRUPT OCCURRED FLAGS FOR EACH DEVICE.
2575 003506 EOTFLG:: .BLKW 4 ;EOT/BOT FLAGS FOR EACH DEVICE (XSTATO).
2576 003516 000000 BTPT:: .WORD 0 ;BAD TAPE SPOT POINTER TO BT0-BT3 VIA BTADDR
2577 003520 000 EXPBOT:: .BYTE 0 ;BOT IS EXPECTED, DO NOT ABORT ON BOT/FUNC R'I.
2578 003521 000 RANDOM:: .BYTE 0 ;RANDOM EVERYTHING FLAG.
2579 003522 000 VFYFLG:: .BYTE 0 ;SET DURING WRITE/VERIFY COMMAND.
2580 003523 000 RPTFLG:: .BYTE 0 ;PERFORMANCE REPORT HAS BEEN REQUESTED.
2581 003524 000 SWBFLG:: .BYTE 0 ;ENABLES SWAP BYTE FUNCTION WHEN NOT EQUAL TO ZERO.
2582 003525 000 LOOPCT:: .BYTE 0 ;WRITE LOOP CONTROL (TEST 3)
2583 003526 000 IRE:: .BYTE 0 ;INHIBIT RESIDUAL FRAME COUNT ERROR REPORT.
2584 003527 000 DROPED:: .BYTE 0 ;CURRENT UNIT HAS BEEN DROPPED
2585 003530 000 T1SWB:: .BYTE 0 ;TEST1 SWAP BYTES FLAG
2586 003531 000 ALLEOT:: .BYTE 0 ;ALL UNITS @ EOT FLAG
2587 003532 000 STREAM:: .BYTE 0 ;INDICATES TEST ONE UNIT AT A TIME, COMPLETELY.
2588 003533 000 ERSFLG:: .BYTE 0 ;ERASE FLAG: DO ERASE AFTER A SPACE REV TO DELETE
2589 ;BADLY WRITTEN RECORD. 1 TO 4 ERASES LEAVING
2590 ;A 3 TO 12 INCH GAP MAY RESULT.
2591 .EVEN
2592 003534 ENDFLG=.
2593
2594 ; ADDITIONAL FLAGS, THESE FLAGS ARE CLEARED ONLY AFTER BEING CHECKED.
2595
2596 003534 000 STAFLG:: .BYTE 0 ;START FLAG - SET BY INIT CODE IF STARTING.
2597 003535 000 PWRFLG:: .BYTE 0 ;POWER FAILURE FLAG - SET ONLY DURING INIT.
2598 003536 000 TRAPD4:: .BYTE 0 ;TRAPED AT 4 FLAG
2599 003537 000 MISCFG:: .BYTE 0 ;MISCELLANEOUS FLAG
2600

```

```

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

```

```

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

```

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

```

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

```

```

2777 .SBTTL GLOBAL ERROR REPORT SECTION
2778
2779
2780 ;**
2781 ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX CALLS
2782 ; THAT ARE USED IN MORE THAN ONE TEST. IT ALSO INCLUDES THE ASCII MESSAGES
2783 ; THAT ARE USED BY THE PRINTB AND PRINTX CALLS..
2784 ;*
2785
2786 005460 BGNMSG DTAERM
005460
2792 005460 DTAERM::
005460 016546 003330 PRINTB #STAER1,DEVTBL(R5),PASCNT(R5),RECCNT(R5)
005464 016546 003260
005470 016546 002536
005474 012746 006140
005500 012746 000004
005504 010600
005506 104414
005510 062706 000012
2793 005514 PRINTB #STAER7
005514 012746 006232
005520 012746 000001
005524 010600
005526 104414
005530 062706 000004
2794 005534 LET RECRED := R2 ;SAVE R2
005534 010237 006546 MOV R2,RECRED
2795 005540 LET TIME1 := R3 ;SAVE R3
005540 010337 003444 MOV R3,TIME1
2796 005544 LET TIME2 := R4 ;SAVE R4
005544 010437 003446 MOV R4,TIME2
2797 005550 JSR PC,RECTAP ;RETRIEVE RECORD READ
2798 005554 LET R2 := RECRED ;RESTORE R2
005554 013702 006546 MOV RECRED,R2
2799 005560 LET RECRED := R3 ;SAVE RECORD READ
005560 010337 006546 MOV R3,RECRED
2800 005564 LET R3 := TIME1 ;RESTORE R3
005564 013703 003444 MOV TIME1,R3
2801 005570 LET R4 := TIME2 ;RESTORE R4
005570 013704 003446 MOV TIME2,R4
2802 005574 PRINTB #STAER6,RECRED ;PRINT RECORD READ
005574 013746 006546 MOV RECRED,(SP)
005600 012746 006262 MOV #STAER6,-(SP)
005604 012746 000002 MOV #2,(SP)
005610 010600
005612 104414
005614 062706 000006 MOV SP,R0
2803 005620 EXIT MSG TRAP C$PNTB
005620 000167 ADD #6,SP
005622 000000 .WORD JSJMP
2804 .EVEN .WORD L10002 2
2805
2806 005624 ENDMMSG
005624 L10002:
005624 104423 TRAP C$MSG
2807

```

2808	005626		BGNMSG STAERM		
	005626		STAERM::		
2809	005626		PRINTB #STAER1,DEVTBL(R5),PASCNT(R5),RECCNT(R5)		
	005626	016546			MOV RECCNT(R5),-(SP)
	005632	016546			MOV PASCNT(R5),-(SP)
	005636	016546			MOV DEVTBL(R5),-(SP)
	005642	012746			MOV #STAER1, -(SP)
	005646	012746			MOV #4, -(SP)
	005652	010600			MOV SP, R0
	005654	104414			TRAP C\$PNTB
	005656	062706			ADD #12, SP
2810	005662		PRINTB #STAER7		
	005662	012746			MOV #STAER7, (SP)
	005666	012746			MOV #1, (SP)
	005672	010600			MOV SP, R0
	005674	104414			TRAP C\$PNTB
	005676	062706			ADD #4, SP
2811	005702		LET R2 := CNDPKT CLR.BY #177740		
	005702	013702			MOV CNDPKT, R2
	005706	042702			BIC #177740, R2
2812	005712		LET R2 := R2 #1		
	005712	005302			DEC R2
2813	005714		IF R2 EQ #0 THEN	; IF CMD IS A READ	
	005714	005702			TST R2
	005716	001016			BNE 50000\$
2814	005720	004737	JSR PC, RECTAP	; THEN RETRIEVE	
2815	005724		LET RECD := R3	; AND	
	005724	010337			MOV R3, RECD
2816	005730		PRINTB #STAER6, RECD	; TYPE RECORD READ	
	005730	013746			MOV RECD, -(SP)
	005734	012746			MOV #STAER6, -(SP)
	005740	012746			MOV #2, (SP)
	005744	010600			MOV SP, R0
	005746	104414			TRAP C\$PNTB
	005750	062706			ADD #6, SP
2817	005754		ENDIF		
	005754			50000\$:	
2818	005754		PRINTX #STAER2		
	005754	012746			MOV #STAER2, -(SP)
	005760	012746			MOV #1, (SP)
	005764	010600			MOV SP, R0
	005766	104415			TRAP C\$PNTX
	005770	062706			ADD #4, SP
2819	005774		PRINTX #STAER3, CNDPKT, @TSDB(R5), MSGPKT+MS.RFC, TSSREG, CTCC		
	005774	013746			MOV CTCC, -(SP)
	006000	013746			MOV TSSREG, (SP)
	006004	013746			MOV MSGPKT+MS.RFC, -(SP)
	006010	017546			MOV @TSDB(R5), (SP)
	006014	013746			MOV CNDPKT, (SP)
	006020	012746			MOV #STAER3, -(SP)
	006024	012746			MOV #6, (SP)
	006030	010600			MOV SP, R0
	006032	104415			TRAP C\$PNTX
	006034	062706			ADD #16, SP
2820	006040		PRINTX #STAER4, CNDPKT+2, CNDPKT+4, CNDPKT+6		
	006040	013746			MOV CNDPKT+6, (SP)
	006044	013746			MOV CNDPKT+4, (SP)

Address	Instruction	Op-Code	Register	Value	Comment
006050	013746	002316			
006054	012746	006433			
006060	012746	000004			
006064	010600				
006066	104415				
006070	062706	000012			
2821 006074	PRINTX				#STAER5,MSGPKT*MS.XS0,MSGPKT*MS.XS1,MSGPKT*MS.XS2,MSGPKT*MS.XS3
006074	013746	002354			
006100	013746	002352			
006104	013746	002350			
006110	013746	002346			
006114	012746	006453			
006120	012746	000005			
006124	010600				
006126	104415				
006130	062706	000014			
2822 006134	EXIT				MSG
006134		000167			
006136		000410			
2823					
2824					
2825 006140		045	101	130	STAER1: .MLIST BEX
2826					.ASCIZ /#AXXX CMD FAILED - UNIT #D1#S3#APASS:#D5#S3#ARECORD:#D5#N/
2827 006232		045	101	120	STAER7: .ASCIZ /#APREVIOUS CMD WAS XXX/
2828 006262		045	123	061	STAER6: .ASCIZ /#S11#A* RECORD READ:#D5#A */
2829 006316		045	116	045	STAER2: .ASCIZ /#N#ACMDPKT#S2#ATSBA#S4#ARFC#S5#ATSSR#S3#ATCC#N/
2830 006375		045	117	066	STAER3: .ASCIZ /#06#S2#06#S2#06#S2#D1#N/
2831 006433		045	117	066	STAER4: .ASCIZ /#06#N/
2832 006440		045	117	066	.ASCIZ /#06#N/
2833 006445		045	117	066	.ASCIZ /#06#N/
2834 006453		045	101	130	STAER5: .ASCIZ /#AXST0#S4#AXST1#S4#AXST2#S4#AXST3#N/
2835 006516		045	117	066	.ASCIZ /#06#S2#06#S2#06#S2#06#N/
2836					.LIST BEX
2837					.EVEN
2838 006546		000000			RECRED: .WORD 0 ;RECORD READ FROM TAPE
2839					
2840 006550					ENDMSG
006550					L10003:
006550	104423				
					TRAP C\$MSG

```

2842 .SBTTL GLOBAL SUBROUTINES SECTION
2843 ***
2844 ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
2845 ; THAT ARE USED IN MORE THAN ONE TEST.
2846 ; -
2847
2848 ; MODULES TO HANDLE TK25 INTERRUPTS.
2849
2850 006552 BGNSRV TS4IN0 ;DEVICE 0.
2851 006552 TS4IN0::
2852 006552 005237 003476 LET INTFLG := INTFLG + #1 ;SET INTERRUPT OCCURRED FLAG.
2853 006556 ENDSRV INC INTFLG
2854 006556 L10004: RTI
2855 006560
2856 006560 005237 003500 BGNSRV TS4IN1 ;DEVICE 1.
2857 006560 TS4IN1::
2858 006560 005237 003500 LET INTFLG+2 := INTFLG+2 + #1 ;SET INTERRUPT OCCURRED FLAG.
2859 006564 ENDSRV INC INTFLG+2
2860 006564 L10005: RTI
2861 006564 000002
2862 006566
2863 006566 005237 003502 BGNSRV TS4IN2 ;DEVICE 2.
2864 006572 TS4IN2::
2865 006572 005237 003502 LET INTFLG+4 := INTFLG+4 + #1 ;SET INTERRUPT OCCURRED FLAG.
2866 006572 000002 ENDSRV INC INTFLG+4
2867 006572 L10006: RTI
2868 006574
2869 006574 005237 003504 BGNSRV TS4IN3 ;DEVICE 3.
2870 006600 TS4IN3::
2871 006600 005237 003504 LET INTFLG+6 := INTFLG+6 + #1 ;SET INTEPRUPT OCCURRED FLAG.
2872 006600 000002 ENDSRV INC INTFLG+6
2873 006600 L10007: RTI

```

```

2866
2867      ; SUBROUTINE TO RETRIEVE RECORD COUNT READ FROM TAPE FOR ERROR
2868      ; PRINTS.
2869      ; INPUTS:
2870      ; OUTPUTS: R3 = RECORD COUNT READ
2871      ; REGISTERS: R2, R3, R4
2872      ; CALLS:
2873
2874 006602 RECTAP::IF #MOD.CO SETIN CMDWRD THEN ;READ REV FETCH
006602 032737 000400 003426 BIT #MOD.CO.CMDWRD
006610 001430 BEQ 50001$
2875 006612 LET R2 := MSGPKT.MS.RFC + DATARD ;FIND LAST READ AD.
006612 013702 002344 MOV MSGPKT.MS.RFC,R2
006616 063702 003416 ADD DATARD,R2
2876 006622 IF #BIT00 SETIN R2 THEN ;ODD AD.. REASSEMBLE
006622 032702 000001 BIT #BIT00,R2
006626 001417 BEQ 50002$
2877 006630 LET R2 := R2 + #1 ;REC COUNT STARTING
006630 005202 INC R2
2878 006632 LET R3 := (R2) CLR.BY #177400 ;WITH UPPER BYTE FETCH
006632 111203 MOVB (R2),R3
006634 142703 177400 BICB #177400,R3
2879 006640 LET R3 := SWAP R3 ;
006640 000303 SWAB R3
2880 006642 LET R2 := R2 + #1 ;LOWER BYTE AD.
006642 005302 DEC R2
2881 006644 IFB SWBFLG NE #0 THEN
006644 105737 003524 TSTB SWBFLG
006650 001401 BEQ 50003$
2882 006652 LET R2 := R2 + #1 ;LOWER BYTE AD.
006652 005302 ON SWAP DEC R2
2883 006654 ENDF
006654
2884 006654 LET R4 := (R2) CLR.BY #177400 ;FETCH LOWER
006654 111204 BYTE 50003$:
006656 142704 177400 MOVB (R2),R4
2885 006662 LET R3 := R3 OR R4 ;MERGE BYTES
006662 050403 BICB #177400,R4
2886 006664 ELSE
006664 000401 BIS R4,R3
006666 BR 50004$
2887 006666 LET R3 := (R2) ;EVEN AD. FETCH
006666 011203 BR 50002$:
2888 006670 ENDF
006670 MOV (R2),R3
2889 006670 ELSE
006670 000402 50004$:
006672 BR 50005$
2890 006672 LET R3 := @DATARD ;READ FWD FETCH
006672 177703 174520 50001$:
2891 006676 ENDF
006676 MOV @DATARD,R3
2892 50005$:
2893 006676 RTS PC

```

```

2896      ;      SUBROUTINE TO STORE A SET CHARACTERISTIC COMMAND AS
2897      ;      THE FIRST ENTRY IN THE SEQUENCE TABLE.
2898      ;      INPUTS:
2899      ;      OUTPUTS:
2900      ;      REGISTERS:
2901      ;      CALLS:
2902
2903 006700      SETCH:: LET R1 := #CMDSEQ      ;INIT COMMAND SEQUENCE TABLE POINTER.
2904 006700 012701 003542      MOV      #SCH,(R1)+      ;THIS CODE SETS UP A SET CHARACTERISTIC
2905 006704 012721 140004      MOV      #DFTSCH,(R1)+      ;COMMAND AS THE FIRST COMMAND IN THE
2906 006710 012721 000040      MOV      #1,(R1)+      ;SEQUENCE TABLE.
2907 006714 012721 000001      TST      (R1)+      ;SKIP PATTERN LOCATION.
2908 006720 005721      RTS      PC
2909 006722 000207
2910
2911
2912
2913      ;      SUBROUTINE TO STORE A REWIND COMMAND IN THE SEQUENCE TABLE
2914      ;      INPUTS:
2915      ;      OUTPUTS:
2916      ;      REGISTERS:
2917      ;      CALLS:
2918
2919 006724      SETRW:: LET (R1)+ := #RWD      ;CMD = REWIND.
2920 006724 012721 102010      LET (R1)+ := #1      ;BRF.
2921 006730 012721 000001      LET (R1)+ := #1      ;# OF OPERATIONS.
2922 006734 012721 000001      TST (R1)+      ;SKIP PATTERN.
2923 006740 005721      RTS      PC      ;RETURN
2924 006742 000207

```

```

2925      ; SUBROUTINE TO EXECUTE ALL COMMANDS IN THE SEQUENCE TABLE ON ALL
2926      ; DEVICES.
2927      ; INPUTS:
2928      ; OUTPUTS:      R2 = TERMINATION INDICATOR (0-END OF TABLE,1-EOT)
2929      ; REGISTERS:
2930      ; CALLS:      CMDAC,SETUP,EXSUB,CKHAE,NEXTU,FIRSTU,VFYDAT.
2931
2932      EXALL:: LET R1 := #CMDSEQ      ;INIT SEQUENCE TABLE POINTER.
2933      ;      MOV      #CMDSEQ,R1
2934      ;      WHILE (R1) NE #END DO    ;WHILE THERE ARE CMDs IN THE SEQUENCE TABLE.
2935      ;      50006::
2936      ;      CMP      (R1),#END
2937      ;      BEQ      50007::
2938      ;      JSR PC,SETUP      ;GO SETUP THE COMMAND BLOCK.
2939      ;      WHILE NCNT LT NCNT1 DO  ;WHILE THERE ARE RECORDS REMAINING:
2940      ;      50010::
2941      ;      CMP      NCNT,NCNT1
2942      ;      BGE      50011::
2943      ;      JSR PC,CMDAC      ;STORE CMD ASCII IN ERROR MESSAGE.
2944      ;      IFB RANDOM NE #0 THEN ;IF IN RANDOM MODE:
2945      ;      TSTB     RANDOM
2946      ;      BEQ      50012::
2947      ;      IF CMDWRD EQ #WRT THEN ;IF CMD IS A WRITE THEN:
2948      ;      CMP      CMDWRD,#WRT
2949      ;      BNE      50013::
2950      ;      IFB VFYFLG EQ #0 THEN  ;IF DATA IS NOT TO BE VERIFIED THEN:
2951      ;      TSTB     VFYFLG
2952      ;      BNE      50014::
2953      ;      LET RANB := RANB + RANS ;GENERATE
2954      ;      ADD      RANS,RANB
2955      ;      LET RANS := RANS + RANB ;RANDOM
2956      ;      ADD      RANB,RANS
2957      ;      LET BRFCNT := RANS      ;LENGTH
2958      ;      MOV      RANS,BRFCNT
2959      ;      LET BRFCNT := BRFCNT CLR.BY LENMSK ;MASK RANDOM LENGTH.
2960      ;      BIC      LENMSK,BRFCNT
2961      ;      IF BRFCNT LT #18. THEN  ;DO NOT ALLOW BYTE COUNT OF LESS THAN 18.
2962      ;      CMP      BRFCNT,#18.
2963      ;      BGE      50015::
2964      ;      LET BRFCNT := #18. ;CHANGE COUNT OF 0 17 TO 18.
2965      ;      MOV      #18.,BRFCNT
2966      ;      50015::
2967      ;      LET CMDPKT.CP.CNT := BRFCNT ;MOVE BRFCNT TO CMD PACKET.
2968      ;      MOV      BRFCNT,CMDPKT.CP.CNT
2969      ;      50014::
2970      ;      50013::
2971      ;      50012::
2972      ;      JSR PC,EXSUB      ;ISSUE CMD TO ALL,AWAIT INTS,CHECK STATUS.
2973      ;      JSR PC,CKHAE      ;CHECK HALT AFTER EACH CMD FLAG.
2974      ;      LET R2 := #1      ;SET ALL UNITS AT BOT/EOT.
2975      ;      MOV      #1,R2
2976      ;      JSR PC,FIRSTU      ;FIND FIRST UNIT.
2977      ;      WHILE DFVIBL(R5) NE #END DO ;WHILE THERE ARE MORE UNITS:

```

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

```

2982      ; SUBROUTINE TO ISSUE COMMAND TO ALL DEVICES, WAIT FOR
2983      ; ALL INTERRUPTS, AND CHECK ALL STATUS.
2984      ; INPUTS:
2985      ; OUTPUTS:
2986      ; REGISTERS:
2987      ; CALLS: EXCUTE,GOWAIT,NEXTU,FIRSTU.
2988
2989      EXSUB:: JSR PC,FIRSTU ;SET UP FOR FIRST UNIT.
2990      WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
2991      007240 004737 017300      50026:
2992      007244      CMP DEVTBL(R5),#END
2993      007244 026527 002536 177777      BEQ 50027:
2994      007252 001465      IF #MOD.CO SETIN CMDWRD THEN ;IF CMD IS REVERSE THEN:
2995      007254 032737 000400 003426      BIT #MOD.CO,CMDWRD
2996      007262 001421      BEQ 50030:
2997      007264      IF #X0.BOT NOTSETIN EOTFLG(R5) THEN ;IF NOT AT BOT
2998      007264 032765 000002 003506      BIT #X0.BOT,EOTFLG(R5)
2999      007272 001014      BNE 50031:
3000      007274      IF #X0.EOT SETIN EOTFLG(R5) THEN ;BUT IF AT EOT
3001      007274 032765 000001 003506      BIT #X0.EOT,EOTFLG(R5)
3002      007302 001406      BEQ 50032:
3003      007304      IFB ALLEOT NE #0 THEN ;AND ALL OTHERS AT EOT
3004      007304 105737 003531      TSTB ALLEOT
3005      007310 001402      BEQ 50033:
3006      007312 004737 010654      JSR PC,EXCUTE ;THEN EXECUTE REV CMD
3007      007316      ENDIF ;IF NOT ALL AT EOT, FREEZE UNIT(S) AT EOT
3008      007316      ELSE ;IF NOT AT BOT AND
3009      007316 000402      BR 50034:
3010      007320      JSR PC,EXCUTE
3011      007320 004737 010654      ENDIF ;NOT AT EOT, EXEC REV CMD
3012      007324      ENDIF
3013      007324      ELSE ;ELSE IF CMD IS NOT REVERSE:
3014      007324 000435      BR 50035:
3015      007326      IF CMDLG EQ #2 AND #X0.BOT SETIN EOTFLG(R5) THEN
3016      007326 023727 003434 000002      CMP CMDLG,#2
3017      007334 001011      BNE 50036:
3018      007336 032765 000002 003506      BIT #X0.BOT,EOTFLG(R5)
3019      007344 001405      BEQ 50036:
3020      007346      ;CLEAR BAD SPOT COUNTS WHEN WRITING FROM BOT
3021      007346 016537 002550 003516      LET BTPT := BTADDR(R5)
3022      007354 005077 174136      MOV BTADDR(R5),BTPT
3023      007360      CLR #BTPT
3024      007360      ENDIF
3025      007360      IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN
3026      007360 032765 000001 003506      BIT #X0.EOT,EOTFLG(R5)
3027      007366 001404      BEQ 50037:
3028      007370 032737 000001 003426      BIT #CMD.CO,CMDWRD
3029      007376 001003      BNE 50040:
3030      007400      50037:
3031      ;IF NOT AT EOT OR NOT A MOTION CMD THEN:

```

```

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

```

3035	007530				IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN	50046\$:
	007530	032765	000001	003506		BIT #X0.EOT,EOTFLG(R5)
	007536	001404				BEQ 50054\$
	007540	032737	000001	003426		BIT #CMD.CO,CMDWRD
	007546	001003				BNE 50055\$
	007550					50054\$:
3036					JSR PC,GOWAIT	;IF NOT AT EOT OR NOT A MOTION CMD THEN:
3037	007550	004737	011274		ELSE	;WAIT FOR INT,CHECK STATUS.
3038	007554					
	007554	000405				BR 50056\$
	007556					50055\$:
3039	007556				IFB ALLEOT NE #0 THEN	
	007556	105737	003531			TSTB ALLEOT
	007562	001402				BEQ 50057\$
3040	007564	004737	011274		JSR PC,GOWAIT	
3041	007570				ENDIF	50057\$:
	007570				ENDIF	50056\$:
3042	007570				ENDIF	50053\$:
3043	007570				JSR PC,NEXTU	;FIND NEXT UNIT IN TEST CYCLE.
3044	007570	004737	017346		ENDDO	
3045	007574	000724				BR 50044\$
	007576					50045\$:
3046	007576	000207			RTS PC	;RETURN.

```

3048      ;      THIS SUBROUTINE STORES THE ASCII FOR THE CURRENT COMMAND AND PREVIOUS
3049      ;      COMMAND IN THE STANDARD ERROR MESSAGE.  ON ENTRY LOCATION CMDWRD
3050      ;      CONTAINS CURRENT CMD AND LOCATION PCMDWD CONTAINS PREVIOUS CMD.
3051      ;      INPUTS:
3052      ;      OUTPUTS:
3053      ;      REGISTERS:      R3, R4.
3054      ;      CALLS:      GCMDA
3055
3056      CMDAC:: LET R4 := CMDWRD      ;R4 = CMD BINARY.
3057      007600      013704      003426      ;R4 = CMD BINARY.
3058      007604      004737      007654      ;GET CMD ASCII.
3059      007610      112337      006142      ;MOVE CMD ASCII
3060      007614      112337      006143      ;
3061      007620      111337      006144      ;INTO MSG.
3062      007624      013704      003432      ;R4 = PREVIOUS CMD BINARY.
3063      007630      004737      007654      ;R4 = PREVIOUS CMD BINARY.
3064      007634      000240      ;GET CMD ASCII.
3065      007636      112337      006256      ;MOVE CMD ASCII
3066      007642      112337      006257      ;
3067      007646      111337      006260      ;INTO MSG.
3068      007652      000207      ;R4 = PREVIOUS CMD BINARY.
3069      ;
3070      ;
3071      ;      SUBROUTINE TO FIND THE ASCII EQUIVILENT OF THE COMMAND IN R4.
3072      ;      ADDRESS OF ASCII 1ST WORD IS RETURNED IN R3.
3073      ;      INPUTS:      R4 = PRESENT COMMAND WORD.
3074      ;      OUTPUTS:      R3 = ADDRESS OF PRESENT COMMAND ASCII.
3075      ;      REGISTERS:
3076      ;      CALLS:
3077
3078      GCMDA:: LET R3 := #0      ;INIT CMD TBL POINTER.
3079      007654      005003      ;UNTIL CURRENT CMD IS FOUND:
3080      007656      026304      003644      ;UNTIL CURRENT CMD IS FOUND:
3081      007662      001403      ;
3082      007664      062703      000002      ;SEARCH CMD TABLE.
3083      007670      000772      ;
3084      007672      010304      ;
3085      007674      006203      ;
3086      007676      060403      ;
3087      007700      062703      003732      ;
3088      007704      000207      ;

```

```

3088      ;      THIS SUBROUTINE LOADS THE TK25 COMMAND PACKET FROM ONE
3089      ;      ENTRY IN THE SEQUENCE TABLE.
3090      ;      INPUTS:
3091      ;      OUTPUTS:
3092      ;      REGISTERS:      R2, R3.
3093      ;      CALLS:      GENPAT.
3094
3095      007706      SETUP:: LET CMDLG := #0      ;CLR CMD LOGGING CODE(DISABLES LOGGING)
3096      007706      005037      003434      MOV      (R1)+,CMDPKT      ;LOAD THE COMMAND WORD.
3097      007712      012137      002314      MOV      (R1),CMDPKT+CP.CNT      ;LOAD THE BYTE/RECORD/FILE COUNT.
3098      007716      011137      002322      MOV      (R1),BRFCNT      ;SAVE BRFCNT FOR THIS COMMAND.
3099      007722      011137      003424      MOV      CMDPKT,R2      ;GET CMD.
3100      007726      013702      002314      BIC      #NCHD.C,R2      ;CLR ALL BUT CMD BITS.
3101      007732      042702      177740      MOV      R2,R3      ;SAVE IT TWICE.
3102      007736      010203      000010      SUB      #CMD.C3,R3      ;POSITION COMMAND?
3103      007740      162703      000010      BNE      2$      ;BR IF NOT.
3104      007744      001003      002316      MOV      (R1),CMDPKT+2      ;MOVE BPCR IN 2ND PKT WORD FOR POSITION CMD.
3105      007746      011137      002316      BR      3$
3106      007752      000461      2$:      IF CMDPKT EQ #WTM THEN      ;IF CMD IS A WRITE TAPE MARK THEN:
3107      007754      023727      002314      100011      CMP      CMDPKT,#WTM
3108      007756      001003      50062$      BNE      50062$
3109      007762      007764      012737      000002      003434      ;WTM LOGGING CODE IS 2.
3110      007766      007764      012737      000002      003434      MOV      #2,CMDLG
3111      007772      007772      50062$:
3112      007772      010203      MOV      R2,R3
3113      007774      162703      000001      SUB      #CMD.CO,R3
3114      010000      001017      BNE      1$
3115      010002      013737      003416      002316      MOV      DATARD,CMDPKT+CP.ADL
3116      010010      032737      000400      002314      IF #MOD.CO SET IN CMDPKT THEN
3117      010016      001404      LET CMDLG := #4
3118      010020      012737      000004      003434      ;LOGGING CODE IS 4.
3119      010026      000403      ELSE
3120      010030      012737      000004      003434      ;ELSE IF CMD IS A READ FWD:
3121      010036      000403      BR      50064$
3122      010040      012737      000006      003434      ;LOGGING CODE IS 6.
3123      010046      012737      000006      003434      MOV      #6,CMDLG
3124      010050      012737      000006      003434      ;CONTINUE.
3125      010056      010203      1$:      MOV      R2,R3
3126      010062      162703      000004      SUB      #CMD.C2,R3
3127      010068      001011      BNE      4$
3128      010074      012737      002446      002316      LET CMDPKT+CP.ADL := #SCHBK
3129      010080      012737      000010      002322      MOV      #SCHCNT,CMDPKT+CP.CNT
3130      010086      011137      002454      LET SCHBK+6 := (R1)
3131      010092      000412      BR      3$
3132      010098      010203      4$:      MOV      R2,R3
3133      010104      162703      000006      SUB      #CMD.C1!CMD.C2,R3
3134      010110      001006      BNE      3$
3135      010116      012737      000020      002322      MOV      #DIACNT,CMDPKT+CP.CNT
3136      010122      012737      003414      002316      MOV      #DIABLK,CMDPKT+CP.ADL

```

GLOBAL AREAS MACRO M1200 21 MAR 84 16:45 PAGE 68 1
GLOBAL SUBROUTINES SECTION

SEQ 0073

3131	010116	005721		31:	TST (R1).		;POINT TO N (NUMBER OF TIMES TO EXECUTE THIS INSTRUJC
3132	010120				LET NCNT1 := (R1).		;SAVE NUMBER OF OPERATIONS
	010120	012137	003422				MOV (R1),NCNT1
3133	010124				LET NCNT := #0		;CLEAR OPERATION COUNTER.
	010124	005037	003420				CLR NCNT
3134	010130	012137	003454		MOV (R1),PATTERN		;SAVE PATTERN CODE FOR CURRENT CMD.
3135	010134	010203			MOV R2,R3		;IS IT
3136	010136	162703	000005		SUB #CMD.C0!CMD.C2,R3		;A WRITE?
3137	010142	001010			BNE 51		;BR IF NOT.
3138	010144	013737	003414 002316		MOV DATAWT,CMDPKT+CP,ADL		;LOAD WRITE BUFFER LO ORDER.
3139	010152	004737	010264		JSR PC,GENPAT		;GO GENERATE THE WRITE PATTERN.
3140	010156				LET CMDLG := #2		;WRITE LOGGING CODE IS 2.
	010156	012737	000002 003434				MOV #2,CMDLG
3141	010164			51:	IF #VFY.C SETIN CMDPKT THEN		;IF DATA VERIFICATION IS REQUIRED:
	010164	032737	000100 002314				BIT #VFY.C,CMDPKT
	010172	001407					BEQ 500651
3142	010174				LET VFYFLG := #1		;SET VERIFY FLAG.
	010174	112737	000001 003522				MOVB #1,VFYFLG
3143	010202	042737	000100 002314		BIC #VFY.C,CMDPKT		;CLEAR VERIFY BIT (NOT USED BY HARDWARE).
3144	010210				ELSE		;IF DATA VERIFICATION IS NOT REQUIRED:
	010210	000402					BR 500661
	010212						500651:
3145	010212				LET VFYFLG := #0		;CLR VERIFY FLAG.
	010212	105037	003522				CLRB VFYFLG
3146	010216				ENDIF		
	010216				LET PCMDWD := CMDWRD		500661:
3147	010216						;SAVE PREVIOUS CMD WORD.
	010216	013737	003426 003432		LET CMDWRD := CMDPKT		MOV CMDWRD,PCMDWD
3148	010224						;SAVE PRESENT CMD WORD.
	010224	013737	002314 003426		IFB SWBFLG NE #0 THEN		MOV CMDPKT,CMDWRD
3149	010232						;IF SWAP BYTES IS ENABLED:
	010232	105737	003524				TSTB SWBFLG
	010236	001403					BEQ 500671
3150	010240				LET CMDPKT := CMDPKT SET.BY #SWB.C ;SET SWAP		BIT IN COMMAND.
	010240	052737	010000 002314				BIS #SWB.C,CMDPKT
3151	010246				ENDIF		
	010246						500671:
3152	010246	042737	004000 002314		BIC #BRF.C,CMDPKT		;CLR BRF BIT (INTERNAL ONLY).
3153	010254				LET CMD5AV := CMDPKT		;SAVE 1ST WORD OF COMMAND PACKET.
	010254	013737	002314 003430				MOV CMDPKT,CMD5AV
3154	010262	000207			RTS PC		;RETURN.

```

3156      ;      THIS SUBROUTINE SETS UP AND CALLS THE APPROPRIATE SUBROUTINE TO GENERATE
3157      ;      THE DESIRED PATTERN FOR THE WRITE AND WRITE/VERIFY COMMANDS.
3158      ;      INPUTS:
3159      ;      OUTPUTS:
3160      ;      REGISTERS:      R2, R3, R4.
3161      ;      CALLS:          PATRO  PATR7
3162
3163      GENPAT:: LET R3 := PATERN SHIFT 1      ;SETUP PATTERN ROUTINE POINTER
3164      010264 013703 003454      MOV      PATERN,R3
3165      010270 006303      ASL      R3
3166      010272 013704 003424      LET R4 := BRCNT * #1      ;SET LENGTH OF WRITE BFR
3167      010276 005204      MOV      BRCNT,R4
3168      010300 042704 000001      INC      R4
3169      010304 162704 000002      LET R4 := R4 CLR.BY #1      ;ROUNDED UP TO NEXT WORD
3170      010310 013702 003414      BIC      #1,R4
3171      010314 062702 000002      LET R4 := R4 * #2      ;WITH FIRST WORD RESERVED
3172      010320 004773 010326      SUB      #2,R4
3173      010324 000207      LET R2 := DATAWT * #2      ;FOR RECORD COUNT
3174      JSR      PC, @PATIBL(R3)      ;GO GENERATE THE APPROPRIATE PATTERN.
3175      RTS      PC      ;RETURN TO SETUP SUBROUTINE.
3176
3177      ;TK25 WRITE PATTERN LOOKUP TABLE. USED TO JSR TO THE
3178      ;CORRECT DATA PATTERN GENERATING ROUTINE.
3179
3180      PATIBL: PATRO      ; INCREMENTING PATTERN, 0 377
3181      PATR1      ; ALL ONES PATTERN
3182      PATR2      ; ALL ZEROES PATTERN
3183      PATR3      ; '1' BIT SHIFT, RIGHT TO LEFT
3184      PATR4      ; '0' BIT SHIFT, RIGHT TO LEFT
3185      PATR5      ; ALTERNATE '0' & '1' WITH ALT. BYTES COMPL.
3186      PATR6      ; ALTERNATE BYTES OF 000 AND 377
3187      PATR7      ; RANDOM PATTERN.
3188      PATR8      ; DUMMY. NO PATTERN, JUST EXITS.
3189
3190      ;INCREMENTING PATTERN. 0 377.
3191
3192      PATRO:: LET R3 := #400
3193      010350 012703 000400      MOV      #400,R3
3194      010354 162704 000002      1$: LET R4 := R4 * #2      ;DECREMENT WORD COUNT.
3195      010360 100411      SUB      #2,R4
3196      010362 010322      BMI      2$      ;BR IF DONE.
3197      010364 062703 001002      LET (R2) := R3      ;STORE DATA WORD.
3198      010370 020327 001000      MOV      R3,(R2)
3199      010374 001002      LET R3 := R3 * #1002      ;UPDATE PATTERN.
3200      010376 012703 000400      IF R3 EQ #1000 THEN      ;IF PATTERN HAS WRAPPED AROUND THEN:
3201      010402 000764      ADD      #1002,R3
3202      010404 000207      CMP      R3,#1000
3203      BR      1$      BNE      50070$
3204      LET R3 := #400      ;INIT THE PATTERN AGAIN.
3205      MOV      #400,R3
3206      ENDIF
3207
3208      50070$:
3209      BR      1$      ;DO IT AGAIN.
3210      RTS      PC      ;RETURN.
3211
3212      2$:

```

16

GLOBAL AREAS MACRO M1200 21 MAR 84 16:45 PAGE 64 1
GLOBAL SUBROUTINES SECTION

SEQ 0075

```

3197
3198
3199
3200 010406 012703 177777 PATR1:: MOV #1,R3 ;ALL ONES PATTERN;
3201 010412 ZROPAT: LET R4 := R4 #2 ;DECREMENT BYTE COUNT.
      010412 162704 000002 SUB #2,R4
3202 010416 100402 BMI 1$ ;DONE?,BR IF YES.
3203 010420 010322 MOV R3,(R2)+ ;IF NOT LOAD NEXT BYTE WITH PATTERN.
3204 010422 000773 BR ZROPAT ;DO IT AGAIN.
3205
3206 010424 000207 1$: RTS PC ;RETURN.

```

```

3208                                ;ALL ZEROES PATTERN.
3209
3210 010426 005003                    PATR2:: CLR    R3                ;CLR PATTERN REGISTER.
3211 010430 004737 010412            JSR    PC,ZROPAT            ;GO GENERATE IT.
3212 010434 000207                    RTS    PC                ;RETURN.
3213
3214                                ;ONE BIT WALKING FROM R TO L IN A FIELD OF ZEROES.
3215
3216 010436 012703 000401            PATR3:: MOV    #401,R3        ;INIT PATTERN REGISTER.
3217 010442                    WLKZRO: LET R4 := R4 - #2        ;DECREMENT WORD COUNT.
3218 010446 162704 000002                    SUB    #2,R4
3219 010450 010322                    BMI    1$                ;BR IF DONE.
3220 010452 006303                    MOV    R3,(R2)+            ;LOAD DATA.
3221 010454 005503                    ASL    R3                ;SHIFT PATTERN.
3222 010456 000771                    ADC    R3                ;ADD CARRY BACK INTO PATTERN.
3223 010460 000207                    BR     WLKZRO            ;DO IT AGAIN.
3224                                1$: RTS    PC                ;RETURN.
3225                                ;ZERO BIT WALKING FROM R TO L IN A FIELD OF 1'S.
3226
3227 010462 012703 177376            PATR4:: MOV    #177376,R3    ;INIT PATTERN REGISTER.
3228 010466 004737 010442            JSR    PC,WLKZRO        ;GO GENERATE ;IT.
3229 010472 000207                    RTS    PC                ;RETURN.
3230
3231                                ;ALTERNATING ONE AND ZERO BITS WITH ALTERNATE BYTES
3232                                ;COMPLEMENTED.
3233
3234 010474 012703 125125            PATR5:: MOV    #125125,R3    ;INIT PATTERN REGISTER.
3235 010500 004737 010412            JSR    PC,ZROPAT        ;GO GENERATE IT.
3236 010504 000207                    RTS    PC                ;RETURN.
3237
3238                                ;ALTERNATING BYTES OF 000 AND 377.
3239
3240 010506 012703 177400            PATR6:: MOV    #177400,R3    ;INIT PATTERN REGISTER.
3241 010512                    1$: LET R4 := R4 - #2        ;DECREMENT WORD COUNT.
3242 010516 162704 000002                    SUB    #2,R4
3243 010520 010322                    BMI    2$                ;BR IF DONE.
3244 010522 000773                    MOV    R3,(R2)+            ;LOAD DATA.
3245 010524 000207                    BR     1$                ;DO IT AGAIN.
3246                                2$: RTS    PC                ;RETURN.
3247                                ;RANDOM PATTERN GENERATOR
3248
3249 010526                    PATR7:: MOV    #RS1,RAN1        ;SET UP THE SEED
3250 010526 012737 001233 010644        MOV    #RS2,RAN2        ;SET UP THE SEED
3251 010534 012737 007622 010646        MOV    #RS3,RAN3        ;SET UP THE SEED
3252 010542 012737 000000 010650        LET R4 := R4 - #2        ;DECREMENT WORD COUNT
3253 010550                    1$: SUB    #2,R4
3254 010554 162704 000002                    BMI    GIT                ;BR IF DONE.
3255 010556 010346                    MOV    R3,-(SP)            ;MOVE THE FIRST SEED INTO R3
3256 010560 013703 010644                    MOV    RAN1,R3        ;CLEAR THE CARRY FLAG
3257 010564 000241                    CLC
3258 010566 005337 010650                    DEC    RAN3        ;DECREMENT THE THIRD SEED
3259 010572 006103                    ROL    R3
3260 010574 006103                    ROL    R3
3261 010576 063703 010646                    ADD    RAN2,R3        ;ADD THE SECOND SEED TO R3

```

```

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

```

```

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

```

3316	011014	001405			LET (R3) := #1	BEQ 500778	
	011016						;INIT THE MSG PACKET WITH ALL 1'S
3317	011016	012723	177777		LET R2 := R2 + #2	MOV #1,(R3)	
	011022						;UPDATE COUNTER.
3318	011022	062702	000002		ENDDD	ADD #2,R2	
	011026						BR 500768
	011026	000770					500778:
3319	011030	105737	002212		TSTB DINT		;ARE INTERRUPTS DISABLED.
3320	011034	001023			BNE 18		;BR IF YES.
3321	011036				IFB INTFLG(R5) GT #1 THEN		;IF MORE THAN ONE INTERRUPT HAS OCCURED:
	011036	126527	003476	000001		CMPB INTFLG(R5),#1	
	011044	003412				BLE 501008	
3322	011046				LET TSSREG := @TSSR(R5)		;FREEZE THE CURRENT STATUS REG FOR PRINT
	011046	017537	002466	003462		MOV @TSSR(R5),TSSREG	
3323	011054				ERRDF 15,TOOMM,STAERM		;REPORT TOO MANY INTERRUPTS.
	011054	104455				TRAP C:ERDF	
	011056	000017				.WORD 15	
	011060	004542				.WORD TOOMM	
	011062	005626				.WORD STAERM	
3324	011064	004737	017402		JSR PC,DROPU		;DROP THE UNIT
3325	011070	000500			BR EXCRTN		;RETURN UNIT HAS BEEN DROPPED.
3326	011072				ENDIF		
3327	011072				LET INTFLG(R5) := #0	501008:	
	011072	005065	003476				;CLR INTERRUPT FLAG FOR THIS DEV.
3328	011076	052737	000200	002314	BIS #IE.C,CMDPKT	CLR INTFLG(R5)	
3329	011104				IFB ERRREC EQ #0 THEN		;SET INT ENABLE BIT.
	011104	105737	003475				;IF NOT RETRYING
	011110	001005				TSTB ERRREC	
3330	011112				LET RECCNT(R5) := RECCNT(R5) + #1	BNE 501018	
3331	011116	005265	003330		LET @DATAWT := RECCNT(R5)	INC RECCNT(R5)	
	011116	016577	003330	172270			;THEN UPDATE REC COUNT TO WRITE IT ON TAPE
3332	011124				ENDIF	MOV RECCNT(R5),@DATAWT	
	011124						
3333	011124				IF L8TEST EQ #3 AND CMDWRD EQ @WRT THEN	501018:	
	011124	023727	002114	000003		CMP L8TEST,#3	
	011132	001040				BNE 501028	
	011134	023727	003426	104005		CMP CMDWRD,@WRT	
	011142	001034				BNE 501028	
3334	011144				LET LOOPCT := LOOPCT + 1		
	011144	005046				CLR (SP)	
	011146	113716	003525			MOVB LOOPCT,(SP)	
	011152	063716	000001			ADD 1,(SP)	
	011156	111637	003525			MOVB (SP),LOOPCT	
	011162	062706	000002			ADD #2,SP	
3335	011166				IFB LOOPCT GT 3 THEN		
	011166	123737	003525	000003		CMPB LOOPCT,3	
	011174	003417				BLE 501038	
3336	011176				DELAY 21.		
	011176	012727	000025			MOV #21,(PC)	
	011202	000000				.WORD 0	
	011204	013727	002116			MOV L8DL,(PC)	
	011210	000000				.WORD 0	
	011212	005367	177772			DEC 6,PC	
	011216	001375				BNE 4	

011220	005367	177756				DEC	22(PC)
011224	001367					BNE	.-20
3337	011226				LET LOOPCT :B= 0		
3338	011226	113737	000000	003525		MOVB	0,LOOPCT
	011234				ENDIF		
3339	011234				ENDIF		50103:
	011234						50102:
3340	011234	012775	002314	002456	MOV @CMDPKT,@TSDB(R5)	;LOAD TSDB WITH CMDPKT ADDRESS	
3341						;THIS INITIATES COMMAND EXECUTION.	
3342	011242				IF @TS.SSR SETIN @TSSR(R5) THEN	;IF READY DID NOT DROP THEN:	
	011242	032775	000200	002466		BIT	@TS.SSR,@TSSR(R5)
	011250	001410				BEQ	50104:
3343	011252	004737	011726		JSR PC,MOVMSG	;MOVE CURRENT MESSAGE PACKET TO COMMON.	
3344	011256				ERRDF 3,TOERM,STAERM	;REPORT NO TK25 RESPONSE.	
	011256	104455				TRAP	C:ERDF
	011260	000003				.WORD	3
	011262	004271				.WORD	TOERM
	011264	005626				.WORD	STAERM
3345	011266	004737	017402		JSR PC,DROPU	;DROP THE UNIT	
3346	011272				ENDIF		
	011272						50104:
3347	011272	000207			EXCRIN: RTS PC	;RETURN.	

```

3349      ;      THIS SUBROUTINE WAITS FOR THE TK25 INERRUPT OR DONE BIT TO SET AND ALLOWS THE
3350      ;      OPEATOR TO TRANSFER CONROL TO THE SUPERVISOR.
3351      ;      UPON APPEARANCE OF THE INTERRUPT OR DONE, CHECK TSSR FOR STATUS ERRORS,
3352      ;      LOG BYTES AND ERRORS AND PERFORM ERROR RECOVERY IF NESSASARY.
3353      ;      INPUTS:
3354      ;      OUTPUTS:
3355      ;      REGISTERS:      R2, R3.
3356      ;      CALLS:      DROPU, MOVMSG, RECUD, CHKERR, LOG, CLRERR.
3357
3358      GOWAIT::IF DEVTBL(R5) EQ #NINUSE THEN
3359      011274 026527 002536 177774      CMP      DEVTBL(R5),#NINUSE
3360      011274 001001      BNE      50105$
3361      011302 000563      BR      1$
3362      011306      ENDIF
3363      011306      LET TIME1 := # 1      ;INIT TIME OUT COUNTER.
3364      011306 012737 177777 003444      MOV      # -1,TIME1
3365      011314      REPEAT      ;REPEAT UNTIL INTERRUPT OCCURES:
3366      011314      BREAK      ;HONOR SUPERVISOR BREAKS
3367      011314 104422      TRAP      C$BRK
3368      011316      IF PCMDWD EQ #RWD AND CMDWRD EQ #WRT THEN
3369      011316 023727 003432 102010      CMP      PCMDWD,#RWD
3370      011324 001020      BNE      50107$
3371      011326 023727 003426 104005      CMP      CMDWRD,#WRT
3372      011334 001014      BNE      50107$
3373      011336      ;POSSIBLE FIRST WRITE ON TAPE
3374      011336 012727 000050      MOV      #40.,(PC)+
3375      011342 000000      .WORD      0
3376      011344 013727 002116      MOV      L$DLY,(PC)+
3377      011350 000000      .WORD      0
3378      011352 005367 177772      DEC      -6(PC)
3379      011356 001375      BNE      -4
3380      011360 005367 177756      DEC      22(PC)
3381      011364 001367      BNE      -20
3382      011366      ENDIF      ;SO DELAY TO CALIBRATE TAPE
3383      011366      IF CMDWRD EQ #RWD THEN      ;IF COMMAND WAS REWIND THEN:
3384      011366 023727 003426 102010      CMP      CMDWRD,#RWD
3385      011374 001014      BNE      50110$
3386      011376      DELAY 10.      ;WAIT EXTRA 10 MSECS EACH LOOP.
3387      011376 012727 000012      MOV      #10.,(PC)+
3388      011402 000000      .WORD      0
3389      011404 013727 002116      MOV      L$DLY,(PC)+
3390      011410 000000      .WORD      0
3391      011412 005367 177772      DEC      6(PC)
3392      011416 001375      BNE      -4
3393      011420 005367 177756      DEC      22(PC)
3394      011424 001367      BNE      -20
3395      011426      ENDIF
3396      011426      IF CMDWRD EQ #SFF OR CMDWRD EQ #SFR THEN
3397      011426 023727 003426 105010      CMP      CMDWRD,#SFF
3398      011434 001404      BEQ      50111$
3399      011436 023727 003426 105410      CMP      CMDWRD,#SFR
3400      011444 001014      BNE      50112$

```

```

011446
3372 011446          DELAY 12.          ;ADD DELAY FOR SPACE TAPE MARK COMMANDS
011446 012727 000014
011452 000000
011454 013727 002116
011460 000000
011462 005367 177772
011466 001375
011470 005367 177756
011474 001367
3373 011476          ENDIF
011476
3374 011476          IFB DINT EQ 00 THEN ;IF INTERRUPTS ARE ENABLED.
011476 105737 002212
011502 001003
3375 011504          LET R2 := INTFLG(R5) ;FETCH INTERRUPT OCCURRED FLAG.
011504 016502 003476
3376 011510          ELSE                ;IF IN BRUTUS MODE:
011510 000406
011512
3377 011512          LET R3 := COMP 0TS.SSR ;SET UP A MASK FOR THE DONE BIT.
011512 012703 000200
011516 005103
3378 011520          LET R2 := 0TSSR(R5) CLR.BY R3 ;FETCH DONE BIT.
011520 017502 002466
011524 040302
3379 011526          ENDIF
011526
3380 011526          LET TIME1 := TIME1 01 ;UPDATE TIMEOUT COUNTER.
011526 005337 003444
3381 011532          UNTIL R2 NE 00 OR TIME1 EQ 00 ;REPEAT UNTIL INTERRUPT OR READY OCCURES.
011532 005702
011534 001003
011536 005737 003444
011542 001264
011544
3382 011544          IF TIME1 EQ 00 THEN ;IF TIME OUT HAS OCCURRED:
011544 005737 003444
011550 001022
3383 011552          LET 0DATAWT := RECCNT(R5) 01 ;RE ADJUST REC COUNT DOWN
011552 016577 003330 171634
011560 005377 171630
3384 011564          JSR PC,MOVMSG ;MOVE CURRENT MSG PACKET TO COMMON AREA.
3385 011570          ERDIF 4,NOINTM,STAERM ;REPORT NO INTERRUPT.
011570 104455
011572 000004
011574 004503
011576 005626
3386 011600          JSR PC,DROPU ;DROP THE UNIT.
3387 011604          LET R3 := 0ENDERF
011604 012703 003476
3388 011610          JSR PC,CLRERR ;CLEAR ALL ERROR FLAGS
3389 011614          ELSE
011614 000417
011616
3390 011616          JSR PC,MOVMSG ;MOVE CURRENT MSG. PACKET TO COMMON AREA.
3391 011622          JSR PC,RECUD ;UPDATE THE RECORD COUNT.

```

F7

GLOBAL AREAS MACRO M1200 21 MAR 84 16:45 PAGE 72 2
GLOBAL SUBROUTINES SECTION

SEQ 0083

```

3392 011626 004737 012160      JSR  PC,CHKERR
3393 011632      IFB WRTYFG EQ #0 THEN
      011632 105737 003467
      011636 001006
3394 011640 004737 015436      JSR  PC,LOG
3395 011644      LET R3 := #ENDERF
      011644 012703 003476
3396 011650 004737 011656      JSR  PC,CLRERR
3397 011654      ENDIF
      011654
3398 011654      ENDIF
      011654
3399 011654 000207      1$: RTS  PC

```

```

;CHECK FOR STATUS ERRORS.
;
      TSTB  WRTYFG
      BNE   50120$
;LOG BYTES AND ERRORS.
      MOV   #ENDERF,R3
;CLEAR ALL ERROR FLAGS
      50120$:
      50117$:
;RETURN IF DONE.

```

```

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

```

```

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

```

17

```

3451      ; SUBROUTINE TO ADJUST THE RECORD COUNT.
3452      ; INPUTS:
3453      ; OUTPUTS:
3454      ; REGISTERS:
3455      ; CALLS:
3456
3457 012012 RECUD:: IFB RECLOG EQ #0 THEN          ; IF RECORD HAS NOT BEEN LOGGED:
012012 105737 003471                                YSTB RECLOG
012016 001057                                BNE 501268
3458 012020 LET RECCNT(R5) := RECCNT(R5) + #1
012020 005365 003330                                DEC RECCNT(R5)
3459 012024 IF #BIT0 NOTSETIN CTCC AND #X2.OPM SETIN MSGPKT.MS.XS2 THEN ; IF TAPE MOVED THEN:
012024 032737 000001 003456                                BIT #BIT0,CTCC
012032 001046                                BNE 501278
012034 032737 100000 002352                                BIT #X2.OPM,MSGPKT.MS.XS2
012042 001442                                BEQ 501278
3460 012044 LET RECLOG := RECLOG + #1 ; SET RECORD LOGGED.
012044 105237 003471                                INCB RECLOG
3461 012050 IF CMDWRD EQ #RWD THEN ; IF THIS IS A REWIND CMD:
012050 023727 003426 102010                                CMP CMDWRD,#RWD
012056 001003                                BNE 501308
3462 012060 LET RECCNT(R5) := #0 ; CLEAR RECORD COUNT.
012060 005065 003330                                CLR RECCNT(R5)
3463 012064 ELSE
012064 000431                                BR 501318
012066 501308:
3464 012066 IF #BRF.C SETIN CMDWRD THEN ; IF BRF USED, UPDATE RECORD COUNT.
012066 032737 004000 003426                                BIT #BRF.C,CMDWRD
012074 001425                                BEQ 501328
3465 012076 IF #MOD.CO NOTSETIN CMDWRD THEN ; IF A FORWARD CMD:
012076 032737 000400 003426                                BIT #MOD.CO,CMDWRD
012104 001007                                BNE 501338
3466 012106 IF #MOD.CO NOTSETIN PCMDWD THEN ; IF PREV CMD WAS A FWD ALSO:
012106 032737 000400 003432                                BIT #MOD.CO,PCMDWD
012114 001002                                BNE 501348
3467 012116 LET RECCNT(R5) := RECCNT(R5) + #1 ; INCREMENT RECORD COUNT.
012116 005265 003330                                INC RECCNT(R5)
3468 012122 ENDF
012122
3469 012122 ELSE ; IF REVERSE CMD:
012122 000412                                BR 501358
012124 501338:
3470 012124 IF #MOD.CO SETIN PCMDWD THEN ; IF PREVIOUS CMD WAS A REV ALSO:
012124 032737 000400 003432                                BIT #MOD.CO,PCMDWD
012132 001406                                BEQ 501368
3471 012134 IF #X0.BOT NOTSETIN EOTFLG(R5) THEN ; WHEN NOT AT BOT THEN
012134 032765 000002 003506                                BIT #X0.BOT,EOTFLG(R5)
012142 001002                                BNE 501378
3472 012144 LET RECCNT(R5) := RECCNT(R5) - #1 ; DECREMENT RECORD COUNT.
012144 005365 003330                                DEC RECCNT(R5)
3473 012150 ENDF
012150
3474 012150 ENDF
012150
3475 012150 ENDF
012150
3476 012150 ENDF
012150

```

J7

GLOBAL AREAS MACRO M1200 21 MAR 84 16:45 PAGE 75 1
GLOBAL SUBROUTINES SECTION

SEQ 0087

3477	012150			ENDIF		501321:
	012150					501311:
3478	012150			ENDIF		501271:
	012150					MOV RECCNT(R5),B0DATAWT
3479	012150			LET B0DATAWT := RECCNT(R5)		501261:
3480	012156	016577	003330	ENDIF		
	012156					
3481	012156	000207		RTS PC	;RETURN.	

```

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

```

```

3509 012316 012316 032737 100000 003462 IF #TS.SC SETIN TSSREG THEN ;IF SPECIAL COND STATUS IS SET THEN:
012316 001441 BEQ 50147$
3510 012326 012326 023727 003456 000002 IF CTCC NE #2 THEN ;IF TCC IS NOT 2 THEN:
012334 001405 CMP CTCC,#2
3511 012336 012336 105737 003475 IFB ERRREC EQ #0 THEN ;IF NOT IN ERROR RECOVERY:
012342 001002 BEQ 50150$
3512 012344 005265 003270 INC SCCNT(R5) ;INC SC COUNTER.
3513 012350 012350 ENDIF
3514 012350 012350 ENDIF
3515 012350 012350 032737 004000 003462 IF #TS.NXM SETIN TSSREG OR #TS.UPE SETIN TSSREG THEN ;WHEN NON-EXISTANT MEMO
012356 001004 BIT #TS.NXM,TSSREG
012360 032737 040000 003462 BNE 50152$
012366 001412 BIT #TS.UPE,TSSREG
012370 BEQ 50153$
3516 012370 012370 032737 100000 002352 IF #X2.OPM NOTSETIN MSGPKT+MS.XS2 THEN ;AND TAPE NOT MOVED
012376 001003 BIT #X2.OPM,MSGPKT+MS.XS2
3517 012400 012400 012702 000005 LET R2 := #5 ;SET TCC5 INDEX
3518 012404 012404 000402 ELSE BR 50155$
012406 50154$:
3519 012406 012406 012702 000004 LET R2 := #4 ;TAPE MOVED, SET TCC4 INDEX
3520 012412 012412 000402 ENDIF MOV #4,R2
3521 012412 012412 000402 ELSE BR 50155$
012414 50153$:
3522 012414 012414 013702 003456 LET R2 := CTCC ;SET DETECTED TCC INDEX
3523 012420 012420 000430 ENDIF MOV CTCC,R2
3524 012420 012420 006302 LET R2 := R2 SHIFT 1 ;CURRENT TCC X 2.
3525 012422 004772 012612 JSR PC,@TCCRA(R2) ;GO TO THE TCC HANDLING SUBROUTINE.
3526 012426 000430 ELSE BR 50157$
012430 50147$:
3527 012430 012430 032737 004000 003426 IF #BRF.C SETIN CMDWRD THEN ;IF BRF IS USED IN THIS CMD THEN:
012436 001424 BIT #BRF.C,CMDWRD
3528 012440 012440 005737 002344 IF MSGPKT+MS.RFC NE #0 THEN ;IF THERE IS AN RFC THEN:
012444 001421 TST MSGPKT+MS.RFC
3529 012446 012446 105737 003521 IFB RANDOM EQ #0 ORB VFYFLG NE #0 THEN TSTB RANDOM
012452 001403 BEQ 50162$
012454 105737 003522 TSTB VFYFLG
012460 001413 BEQ 50163$

```

```

3530 012462
3531 012462          IFB IRE EQ #0 THEN          ;IF NOT IN RANDOM OR IF CMD IS WTV:
          012462 105737 003526                    ;IF RFC ERROR REPORTS ARE ALLOWED:
          012466 001010                    TSTB IRE
3532 012470          LET HRDCNT(R5) := HRDCNT(R5) + #1 ;UPDATE HARD ERROR COUNT
          012470 005265 003310                    BNE 50164$
3533 012474          ERRHRD 13,RFCERM,STAERM ;REPORT RFC ERROR          INC HRDCNT(R5)
          012474 104456                    TRAP C$ERRHRD
          012476 000015                    .WORD 13
          012500 004336                    .WORD RFCERM
          012502 005626                    .WORD STAERM
3534 012504 004737 014050          JSR PC,RAMDUM          ;GO DO RAM DUMP
3535 012510          ENDIF
          012510                    50164$:
3536 012510          ENDIF                    50163$:
          012510                    50161$:
3537 012510          ENDIF                    50160$:
          012510                    50157$:
3538 012510          ENDIF                    ;IF A READ/WRITE ERROR HAS OCCURRED THEN:
          012510                    TSTB RWERR
3539 012510          ENDIF                    BEQ 50165$
          012510                    ;RESTORE CMD PACKET AFTER ERROR RECOV.
3540 012510          IFB RWERR NE #0 THEN          MOV CMDSAV,CMDPKT
          012510 105737 003473          LET CMDPKT := CMDSAV
          012514 001403
3541 012516          LET CMDPKT := CMDSAV          ;RESTORE CMD PACKET AFTER ERROR RECOV.
          012516 013737 003430 002314          MOV CMDSAV,CMDPKT
3542 012524          ENDIF
          012524                    50165$:
3543 012524          ENDIF                    50140$:
          012524
3544 012524 000207          1$: RTS PC          ;RETURN.
3545
3546 012526          045 116 045 EWMSG: .ASCIZ /#N#AEARLY WARNING DURING WRITE AT END OF TRACK #D2/
          012531 101 105 101
          012534 122 114 131
          012537 040 127 101
          012542 122 116 111
          012545 116 107 040
          012550 104 125 122
          012553 111 116 107
          012556 040 127 122
          012561 111 124 105
          012564 040 101 124
          012567 040 105 116
          012572 104 040 117
          012575 106 040 124
          012600 122 101 103
          012603 113 040 045
          012606 104 062 000
3547          .EVEN

```

N7

GLOBAL AREAS MACRO M1200 21 MAR 84 16:45 PAGE 77
GLOBAL SUBROUTINES SECTION

SEQ 0091

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

```

3560      ; SUBROUTINE TO HANDLE EARLY WARNING WRITE ERRORS.
3561      ;
3562      ; THIS ROUTINE WILL SIMPLY TAKE CARE OF EARLY WARNING WRITE ERRORS
3563      ; BY REISSUING THE COMMAND WHICH FAILED WITH THE RETRY MODIFIER SET.
3564      ; NOTE THAT NO ERROR FLAGS, COUNTS, ETC ARE CHANGED.
3565      ; (EW ERRORS WILL NOT SHOW UP IN ERROR TALLIES).
3566
3567 012632      EWRTRY::
3568 012632      LET -(SP) := CMDPKT      ;SAVE THE COMMAND PACKET
                                MOV      CMDPKT, (SP)
3569 012632      013746 002314      LET CMDPKT := CMDPKT SET.BY #MOD.C1      ;SET THE RETRY MODIFIER IN COMMAND
                                BIS      #MOD.C1,CMDPKT
3570 012644      052737 001000 002314      LET ERRREC :B= #1      ;SHOW ERROR RECOVERY IN PROCESS
                                MOVB     #1,ERRREC
3571 012652      112737 000001 003475      JSR      PC,EXCUTE      ;DO THE COMMAND
3572 012656      004737 010654      LET ERRREC :B= #0      ;NO MORE ERROR RECOVERY
                                CLRB     ERRREC
3573 012662      012637 002314      LET CMDPKT := (SP).      ;RESTORE THE ORIGINAL COMMAND STATUS
                                MOV      (SP).,CMDPKT
3574 012666      012602      LET R2 := (SP).      ;MODIFY THE RETURN ADDRESS
                                MOV      (SP).,R2
3575 012670      012746 000012      LET -(SP) := #10.      ;OVERALL LOOP CONTROL
                                MOV      #10.,(SP)
3576 012674      REPEAT      ;LOOP
3577 012674      501661:      DELAY 250.      ;WAIT FOR THE UNIT TO STOP
                                MOV      #250.,(PC).
                                .WORD    0
                                MOV      L#DLV,(PC).
                                .WORD    0
                                DEC      -6(PC)
                                BNE      . 4
                                DEC      22(PC)
                                BNE      .-20
3578 012724      LET (SP) := (SP) - #1      ;ONE LESS ITERATION
                                DEC      (SP)
3579 012726      005316      UNTIL (SP) EQ #0      ;UNTIL = 0
                                TST      (SP)
                                BNE      501661
3580 012732      005716      LET SP := SP + #2      ;CORRECT THE STACK
                                ADD      #2,SP
3581 012736      000207      RTS      PC      ;RETURN TO CALLER

```

```

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

```

```

3611      ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 2, TAPE STATUS ALERT.
3612      ; A STATUS CONDITION HAS BEEN ENCOUNTERED THAT MAY HAVE SIGNIFICANCE
3613      ; TO THE PROGRAM. BITS OF INTEREST INCLUDE TMK, RLS, LET, RLL, BOT, EOT.
3614      ; INPUTS:
3615      ; OUTPUTS:
3616      ; REGISTERS:
3617      ; CALLS:
3618
3619 013000 TCC2:: IF #X0.BOT SETIN MSGPKT*MS.X50 ANDB EXPBOT NE #0 THEN
013000      BIT      #X0.BOT,MSGPKT*MS.X50
013006      BEQ      50167#
013010      TSTB     EXPBOT
013014      BEQ      50167#
3620      ; IF AT BOT AND BOT IS EXPECTED:
3621 013016      BR IC2RTN      ; RETURN TCC2 CAUSED BY EXPECTED BOT.
3622 013020      ENDIF
013020
3623 013020      IF L#TEST EQ #3 THEN
013020      CMP      L#TEST,#3
013026      BNE      50170#
3624 013030      IF PCMDWD EQ #WTM AND CMDWRD EQ #SRR THEN
013030      CMP      PCMDWD,#WTM
013036      BNE      50171#
013040      CMP      CMDWRD,#SRR
013046      BNE      50171#
3625 013050      BR IC2RTN
3626 013052      ENDIF
013052
3627 013052      ENDIF
013052
3628 013052      IF #X0.RLS!X0.RLL!X0.TMK!X0.LET!X0.BOT SETIN MSGPKT*MS.X50 THEN
013052      BIT      #X0.RLS!X0.RLL!X0.TMK!X0.LET
013060      BEQ      50172#
3629      ; IF TCC2 CAUSED BY ANYTHING BUT EOT:
3630 013062      IFB RANDOM EQ #0 ORB VFYFLG NE #0 THEN
013062      TSTB     RANDOM
013066      BEQ      50173#
013070      TSTB     VFYFLG
013074      BEQ      50174#
013076
3631      50173#:
3632 013076      IFB IRE EQ #0 THEN      ; IF NOT IN RANDOM OR IF CMD IS WTV:
013076      ; IF RFC ERROR REPORTS ARE ALLOWED:
013102      TSTB     IRE
013102      BNE      50175#
3633 013104      IFB ERRREC NE #0 THEN      ; IF WE ARE IN ERROR RECOVERY THEN:
013104      TSTB     ERRREC
013110      BEQ      50176#
3634 013112      LET UNREC :B= UNREC + #1      ; SET UNRECOVERABLE FLAG FOR LOG.
013112      INCB     UNREC
3635 013116      ELSE      ; ELSE IF NOT IN ERROR RECOVERY:
013116      BR      50177#
013120      50176#:
3636 013120      LET SCCNT(R5) := SCCNT(R5) + #1 ; INCREMENT THE SPEC COND COUNTER.
013120      INC      SCCNT(R5)
3637 013124      ENDIF
013124
3638 013124      50177#:
      LET HADCNT(P5) := HADCNT(R5) + #1 ; UPDATE HARD ERROR COUNT.

```

```

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

ERRHRD 7,TSAM,STAERM
;REPORT TAPE STATUS ALERT.
INC HRDCNT(R5)
TRAP C$ERRHRD
.WORD 7
.WORD TSAM
.WORD STAERM

JSR PC,RANDUM
ENDIF
;GO DO RAM DUMP
50175:
50174:
50172:
;RETURN.

TC2RTN: RTS PC

; THE SPECIFIED FUNCTION WAS NOT INITIATED. BITS OF INTEREST ARE
; RMR, OFL, VCK, BOT, ILC, WLE, ILA, AND NBA.
; INPUTS:
; OUTPUTS:
; REGISTERS: R2,R4
; CALLS: DROPU

TCC3:: ERRDF 8,FUNRM,STAERM
;REPORT FUNCTION REJECT.
TRAP C$ERRDF
.WORD 8
.WORD FUNRM
.WORD STAERM

JSR PC,RANDUM
JSR PC,DROPU
RTS PC
;GO DO RAM DUMP
;DROP THE UNIT.
;RETURN.

```

Line	Address	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	Op464	Op46
------	---------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	------

```

3696 013312
      013312 013737 002322 014732
3697 013320
      013320 105237 003473
3698 013324
      013324 105237 003467
3699 013330
      013330
3700 013330
      013330 005265 003250
3701 013334
      013334 005037 003464
3702 013340
      013340 105037 003466
3703 013344
      013344 004737 014350
3704 013350
      013350 026527 002536 177774
      013356 001001
3705 013360
      013360 000542
3706 013362
      013362
3707 013362
      013362 105737 003470
      013366 001404
      013370 027727 170122 000050
      013376 103754
      013400
3708
3709 013400
      013400 027727 170112 000050
      013406 103406
3710 013410
      013410 004737 015146
3711 013414
      013414 005365 003330
3712 013420
      013420 004737 014050
3713 013424
      013424
3714 013424
      013424 027727 170066 001000
      013432 103417
3715 013434
      013434 012746 015021
      013440 012746 000001
      013444 010600
      013446 104414
      013450 062706 000004
3716 013454
      013454 004737 017402
3717 013460
      013460 005065 003330
3718 013464
      013464 012775 002334 002456
3719 013472
      013472
3720 013472
      013472 105037 003467
3721 013476
      013476 105237 003537

```

```

LET WTYBRF := CNDPKT.CP.CNT ;
LET RWERR :B= RWERR + #1 ;LOG SUBR FLAG: COUNT WRT ERRORS
LET WRTYFG :B= WRTYFG + #1 ;RETRY IN PROGRESS FLAG
REPEAT
  LET WRTYCT(R5) := WRTYCT(R5) + #1 ;COUNT GLOBAL WRITE RETRIES
  LET RETRYC := #0 ;CLEAR # OF RETRIES PER RECORD
  LET RPTCNT :B= #0 ;CLEAR # OF REPEATS
  JSR PC,WRTY ;CALL WRITE RETRY
  IF DEVTBL(R5) EQ #NINUSE THEN
    BR 3 ;
    ENDIF
  UNTILB WRTYER EQ #0 OR #BTPT HIS #40.
  IF #BTPT HIS #40. THEN
    JSR PC,BORERS ;ERASE BAD RECORD
    ET RECCNT(R5) := RECCNT(R5) - #1
    JSR PC,RANDUM ;GO DO RAM DUMP
    ENDIF
    IF #BTPT HIS #512. THEN
      PRINTB #BTMSG2 ;BAD TAPE OVERFLOW MESSAGE
      JSR PC,DROPU ;DROP THE UNIT.
      LET RECCNT(R5) := #0
      LET #TSDB(R5) := #RWCPK ;REWIND UNIT
    ENDIF
    LET WRTYFG :B= #0 ;RETRY COMPLETE
    LET MISCFG :B= MISCFG + #1 ;DO NOT

```

MOV CNDPKT.CP.CNT,WTYBRF
INC RWERR
INC WRTYFG
50205:
INC WRTYCT(R5)
CLR RETRYC
CLR RPTCNT
CMP DEVTBL(R5),#NINUSE
BNE 50206:
50206:
;REPEAT RETRIES ON SAME RECORD
TSTB WRTYER
BEQ 50207:
CMP #BTPT,#40.
BLO 50205:
50207:
;UNTIL RECOVERED OR 20 BAD SPOTS
;WHEN 20 BAD SPOTS LOGGED (TWR)
CMP #BTPT,#40.
BLO 50210:
DEC RECCNT(R5)
50210:
CMP #BTPT,#512.
BLO 50211:
MOV #BTMSG2,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C#PNTB
ADD #4,SP
CLR RECCNT(R5)
MOV #RWCPK,#TSDB(R5)
50211:
FLAG
CLR WTYFG
HALT ON THIS CMD FLG
INC MISCFG

```

3722 013502          LET PCMDWD := WTYWRD          ;RESTORE ORIGINAL WRT CMD AFTER RECOVERY
013502 013737 014730 003432          MOV          WTYWRD,PCMDWD
3723 013510          ENDIF
013510
3724 013510          ELSE
013510 000402
013512
3725 013512          LET UNREC :B= UNREC * #1      ;
013512 105237 003474          ;
3726 013516          ENDIF
013516
3727 013516          ELSE
013516 000463
013520
3728 013520 004737 014212          JSR PC,RTLE          ;CHECK FOR RETRY LIMIT EXCEEDED.
3729 013524          IF CMDLG GT #2 THEN          ;IF READ CMD THEN:
013524 023727 003434 000002          CMP          CMDLG,#2
013532 003411          BLE          502148
3730 013534          LET R2 := #RRECL SHIFT 1      ;R2=READ RETRY COUNT LIMIT / 2
013534 012702 000020          MOV          #RRECL,R2
013540 006202          ASR          R2
3731 013542          IF RETRYC GE R2 THEN          ;IF RETRY COUNT IS MORE THAN HALF LIMIT:
013542 023702 003464          CMP          RETRYC,R2
013546 002403          BLT          502158
3732 013550          LET CMDPKT := CMDPKT SET.BY #OPP.C
013550 052737 020000 002314          ;SET OPPOSITE BIT FOR RETRY 2.
3733 013556          ENDIF          BIS          #OPP.C,CMDPKT
013556
3734 013556          ENDIF
013556
3735 013556          IF RETRYC EQ #0 ANDB ERCVER NE #0 THEN ;IF THIS IS THE ORIGINAL ERROR THEN:
013556 005737 003464          TST          RETRYC
013562 001011          BNE          502168
013564 105737 002207          TSTB         ERCVER
013570 001406          BEQ          502168
3736 013572          ERRSOFT 9,RERM,STAERM          ;REPORT RECOVERABLE ERROR
013572 104457          TRAP          C:ERSOFT
013574 000011          .WORD        9
013576 004632          .WORD        RERM
013600 005626          .WORD        STAERM
3737 013602 004737 014050          JSR PC,RANDUM          ;GO DO RAM DUMP
3738 013606          ENDIF          ;PROVIDED OPERATOR HAS ENABLED THE REPORT
013606
3739 013606          LET RETRYC := RETRYC * #1      ;UPDATE RETRY COUNT.
013606 005237 003464          INC          RETRYC
3740 013612          LET CMDPKT := CMDPKT SET.BY #MOD.C1 ;SET RETRY BIT IN CMD PACKET.
013612 052737 001000 002314          BIS          #MOD.C1,CMDPKT
3741 013620          IFB IREC EQ #0 THEN          ;IF ERROR RECOVERY ENABLED:
013620 105737 002210          TSTB         IREC
013624 001016          BNE          502178
3742 013626          IF DEVTBL(R5) EQ #NINUSE THEN
013626 026527 002536 177774          CMP          DEVTBL(R5),#NINUSE
013634 001001          BNE          502208
3743 013636 000413          BR 38
3744 013640          ENDIF
013640
3745 013640          LET ERRREC :B= ERRREC * #1      ;SET ERROR RECOVERY FLAG.

```

Address	Hex	Hex	Hex	Assembly	Comment
3746	013640	105237	003475		
	013644			POP R2,R2	INCB ERRREC ;POP 2 RTN ADRS FROM STACK.
	013644	012602			MOV (SP),R2
	013646	012602			MOV (SP),R2
3747	013650	004737	010654	JSR PC,EXECUTE	;GO EXECUTE THE RETRY COMMAND.
3748	013654	000137	011274	JMP GOWAIT	;GO WAIT FOR INTERRUPT . CHECK STATUS.
3749	013660			ELSE	;ELSE IF ERROR RECOVERY IS NOT ENABLED:
	013660	000402			BR 502218
	013662				502178:
3750	013662			LET UNREC :B= UNREC . #1	;SET UNRECOVERABLE ERROR FLAG.
	013662	105237	003474		INCB UNREC
3751	013664			ENDIF	502218:
	013666				
3752	013666			ENDIF	502138:
	013666				
3753	013666	000207		38: RTS PC	;RETURN

```

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

```

```

3783      ;      SUBROUTINE TO HANDLE TERMINATION CLASS CODE 6, UNRECOVERABLE ERROR.
3784      ;      TAPE POSITION HAS BEEN LOST.  THE ONLY VALID RECOVERY PROCEDURE
3785      ;      IS TO REWIND AND START OVER AT BOT UNLESS THE TAPE HAS LABELS OR
3786      ;      SEQUENCE NUMBERS.  THIS DIAGNOSTIC WILL REWIND AND RETRY THE
3787      ;      COMMAND ONLY IF DENSITY CHECK IS SET, OTHERWISE THE UNIT WILL BE
3788      ;      DROPPED FROM THE TEST SEQUENCE.
3789      ;      INPUTS:
3790      ;      OUTPUTS:
3791      ;      REGISTERS:      R2, R4
3792      ;      CALLS:      RTLE, WSSR, EXCUTE, GOWAIT, DROPU
3793
3794 013772      TCC6:: LET @TSDB(R5) := @RWCPK      ;ISSUE A REWIND COMMAND.
3795 013772 012775 002334 002456      MOV      @RWCPK,@TSDB(R5)
3796 014000 004737 011672      JSR PC,WSSR      ;WAIT FOR SUBSYSTEM READY.
3796 014004 104455      ERRDF 11,URERM,STAERM      ;REPORT UNRECOVERABLE ERROR.
3796 014006 000013      TRAP      C$ERDF
3796 014010 004654      .WORD      11
3796 014012 005626      .WORD      URERM
3797 014014 004737 014050      JSR      PC,RAMDUM      ;GO DO RAM DUMP
3798 014020 004737 017402      JSR      PC,DROPU      ;DROP THE UNIT.
3799 014024 000207      RTS PC      ;RETURN

```

```

3801      ; SUBROUTINE TO HANDLE TERMINATION CLASS CODE 7, FATAL SUBSYSTEM
3802      ; ERROR. THE SUBSYSTEM IS INCAPABLE OF PROPERLY PERFORMING
3803      ; COMMANDS OR AT LEAST ITS INTEGRITY IS SERIOUSLY QUESTIONABLE.
3804      ; REFER TO THE FATAL CLASS CODE FIELD IN THE TSSR REGISTER FOR
3805      ; ADDITIONAL INFORMATION ON THE TYPE OF FATAL ERROR.
3806      ; INPUTS:
3807      ; OUTPUTS:
3808      ; REGISTERS:      R2, R4
3809      ; CALLS:
3810
3811      TCC7:: ERRDF 12,FATSM,STAERM      ;REPORT FATAL SUBSYSTEM ERROR.
      014026      104455      TRAP      C$ERDF
      014030      000014      .WORD      12
      014032      004455      .WORD      FATSM
      014034      005626      .WORD      STAERM
3812      014036      004737      014050      JSR      PC,RAMDUM      ;GO DO RAM DUMP
3813      014042      004737      017402      JSR      PC,DROPU      ;DROP THE UNIT.
3814      014046      000207      RTS PC      ;RETURN.
3815
3816
3817
3818
3819      014050      RAMDUM::
3820      014050      IFB RAMWRT NE #0 THEN
      014050      105737      002214      TSTB      RAMWRT
      014054      001455      BEQ      50225$
3821      014056      PRINTX #RAMFHR
      014056      012746      005306      MOV      #RAMFHR,-(SP)
      014062      012746      000001      MOV      #1,-(SP)
      014066      010600      MOV      SP,R0
      014070      104415      TRAP      C$PNTX
      014072      062706      000004      ADD      #4,SP
3822      014076      012737      000010      003406      MOV      #8.,RAMSIZ      ;RAM FIELD IS 8 BYTES LONG
3823      014104      012737      000020      003342      MOV      #20,RAMHLD      ;COMMAND PACKET ADDRESS
3824      014112      004737      015736      JSR      PC,RAMER      ;READ AND PRINT THEM
3825      014116      012737      000200      003342      MOV      #200,RAMHLD      ;CHARACTERISTICS PACKET ADDRESS
3826      014124      004737      015736      JSR      PC,RAMER      ;READ AND PRINT THEM
3827      014130      012737      000016      003406      MOV      #14.,RAMSIZ      ;RAM FIELD IS 8 BYTES LONG
3828      014136      012737      000214      003342      MOV      #214,RAMHLD      ;MESSAGE PACKET ADDRESS
3829      014144      004737      015736      JSR      PC,RAMER      ;READ AND PRINT THEM
3830      014150      012737      000020      003406      MOV      #16.,RAMSIZ      ;RAM FIELD IS SIXTEEN BYTES LONG
3831      014156      012737      000060      003342      MOV      #60,RAMHLD      ;SENCE BYTES ADDRESS
3832      014164      004737      015736      JSR      PC,RAMER      ;READ AND PRINT THEM
3833      014170      PRINTX #RAMLIN
      014170      012746      005450      MOV      #RAMLIN,(SP)
      014174      012746      000001      MOV      #1,(SP)
      014200      010600      MOV      SP,R0
      014202      104415      TRAP      C$PNTX
      014204      062706      000004      ADD      #4,SP
3834      014210      ENDIF
      014210      50225$:
3835      014210      000207      RTS      PC
3836

```

```

3838      ;      SUBROUTINE TO CHECK FOR RETRY LIMIT EXCEEDED. PRINTS ERROR MESSAGE
3839      ;      IF EXCEEDED AND DROP UNIT UNLESS COMMAND IS A READ.
3840      ;      INPUTS:
3841      ;      OUTPUTS:
3842      ;      REGISTERS:      R2, R4.
3843      ;      CALLS:      DROPU
3844
3845
3846      RTLE:: IF CMDLG EQ #0 THEN      ;IF CMD IS NOT A READ OR WRITE THEN:
      014212      005737      003434      TST      CMDLG
      014216      001012      BNE      50226$
3847      014220      ERRDF 11,URERM,STAERM      ;REPORT UNRECOVERABLE ERROR.
      014220      104455      TRAP      C$ERDF
      014222      000013      .WORD      11
      014224      004654      .WORD      URERM
      014226      005626      .WORD      STAERM
3848      014230      004737      014050      JSR      PC,RAMDUM      ;GO DO RAM DUMP
3849      014234      004737      017402      JSR      PC,DROPU      ;DROP THE UNIT.
3850      014240      POP      R2
      014240      012602      MOV      (SP)+,R2
3851      014242      000441      BR      RTLRTN      ;AND RETURN.
3852      014244      ENDIF
      014244
3853      014244      LET RWERR :B= RWERR + #1      ;SET READ/WRITE ERROR FLAG.
      014244      105237      003473      INCB      RWERR
3854      014250      IF CMDLG EQ #2 THEN      ;IF CMD IS A WRT OR WTM:
      014250      023727      003434      000002      CMP      CMDLG,#2
      014256      001020      BNE      50227$
3855      014260      IF RETRYC EQ #WRECL THEN      ;IF RETRY COUNT HAS REACHED LIMIT:
      014260      023727      003464      000020      CMP      RETRYC,#WRECL
      014266      001013      BNE      50230$
3856      014270      LET UNREC :B= UNREC + #1      ;SET UNRECOVERABLE FLAG
      014270      105237      003474      INCB      UNREC
3857      014274      ERRDF 14,RLEXM,STAERM      ;REPORT RETRY LIMIT EXCEEDED.
      014274      104455      TRAP      C$ERDF
      014276      000016      .WORD      14
      014300      004372      .WORD      RLEXM
      014302      005626      .WORD      STAERM
3858      014304      004737      014050      JSR      PC,RAMDUM      ;GO DO RAM DUMP
3859      014310      004737      017402      JSR      PC,DROPU      ;DROP THE UNIT.
3860      014314      012602      MOV      (SP)+,R2
      014314
3861      014316      ENDIF
      014316
3862      014316      ELSE
      014316      000413
      014320
3863      014320      IF RETRYC EQ #RRECL THEN      ;IF RETRY COUNT HAS REACHED LIMIT:
      014320      023727      003464      000020      CMP      RETRYC,#RRECL
      014326      001007      BNE      50232$
3864      014330      LET UNREC :B= UNREC + #1      ;SET UNRECOVERABLE FLAG
      014330      105237      003474      INCB      UNREC
3865      014334      ERRHRD .4,RLEXM,STAERM      ;REPORT RECOVERABLE ERROR.
      014334      104456      TRAP      C$ERHRD
      014336      000016      .WORD      14
      014340      004372      .WORD      RLEXM
      014342      005626      .WORD      STAERM

```

N8

GLOBAL AREAS MACRO M1200 21 MAR 84 16:45 PAGE 85-1
GLOBAL SUBROUTINES SECTION

SEQ 0104

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

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

;RETURN

```

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

```

3906	014454				IFB WRTYER EQ #0 THEN		TSTB WRTYER
	014454	105737	003470				BNE 502441
	014460	001001					EXIT RETRY LOOP IF RECOVERED
3907	014462				LEAVE RETRY		BR 502341
	014462	000457					
3908	014464				ELSE		502441:
	014464						
3909	014464				IFB ERCVER NE #0 THEN		TSTB ERCVER
	014464	105737	002207				BEQ 502461
	014470	001415					PRINT SUSPECTED BAD SPOT
3910	014472				PRINTB #BTMSG1,RETRYC,<B,RPTCNT>		CLR -(SP)
	014472	005046					BISB RPTCNT,(SP)
	014474	153716	003466				MOV RETRYC, -(SP)
	014500	013746	003464				MOV #BTMSG1, -(SP)
	014504	012746	014734				MOV #3, -(SP)
	014510	012746	000003				MOV SP,R0
	014514	010600					TRAP C#PNTB
	014516	104414					ADD #10,SP
	014520	062706	000010				
3911	014524				ENDIF		502461:
	014524						ON FIRST RETRY, LOGG BAD SPOT
3912	014524				IF RETRYC EQ #1 THEN		CMP RETRYC,#1
	014524	023727	003464	000001			BNE 502471
	014532	001021					BTPT IS BOTH THE BAD SPOT COUNTER
3913	014534				LET BTPT := BTADDR(R5)		MOV BTADDR(R5),BTPT
	014534	016537	002550	003516			AND THE LOGGING INDEX
3914	014542				LET R4 := #BTPT * #2		MOV #BTPT,R4
	014542	017704	166750				ADD #2,R4
	014546	062704	000002				
3915	014552				LET #BTPT := R4		MOV R4,#BTPT
	014552	010477	166740				
3916	014556				IF R4 LOS #40. THEN		CMP R4,#40.
	014556	020427	000050				BHI 502501
	014562	101005					STORE FIRST 20 BAD SPOTS
3917	014564				LET R3 := BTPT		MOV BTPT,R3
	014564	013703	003516				
3918	014570				LET R4 := R4 * R3		ADD R3,R4
	014570	060304					
3919	014572				LET (R4) := RECCNT(R5)		MOV RECCNT(R5),(R4)
	014572	016514	003330				
3920	014576				ENDIF		502501:
	014576						
3921	014576				ENDIF		502471:
	014576						ERASE FLAG TO ERASE BAD RECORD
3922	014576				LET ERSFLG := ERSFLG * #1		INCB ERSFLG
	014576	105237	003533				CANCEL "LOG" ERROR FLAG ON FAILING RETRY
3923	014602				LET RWERR := #0		CLRB RWERR
	014602	105037	003473				CLEAR REPEAT COUNT FOR NEXT RETRY
3924	014606				LET RPTCNT := #0		CLRB RPTCNT
	014606	105037	003466				
3925	014612				ENDIF		502451:
	014612						LIMIT: 4 RETRIES
3926	014612				UNTIL RETRYC EQ #4		CMP RETRYC,#4
	014612	023727	003464	000004			BNE 502351
	014620	001257					
3927	014622				END RETRY		502341:
	014622						

```

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

```

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

.EVEN

```

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

```

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

```

4005      ; SUBROUTINE TO LOG BYTES READ/WITTEN.
4006      ; ALSO UPDATES READ/WRITE ERROR COUNTERS.
4007      ;
4008      ; INPUTS:
4009      ; OUTPUTS:
4010      ; REGISTERS:      R2, R3, R4.
4011      ; CALLS:
4012      LOG:: IFB ERLOG EQ #0 THEN      ; IF DATA AND ERRORS HAVE NOT BEEN LOGGED THEN
      015436      105737      003472      TSTB      ERLOG
      015442      001126      BNE      502578
4013      015444      LET ERLOG :B= ERLOG + #1      ; SET LOG DONE FLAG.
      015444      105237      003472      INCB      ERLOG
4014      015450      LET R4 := CMDLG      ; GET CURRENT CMD LOGGING CODE.
      015450      013704      003434      MOV      CMDLG,R4
4015      015454      IF R4 NE #0 THEN      ; IF THERE IS A CODE THEN:
      015454      005704      TST      R4
      015456      001520      BEQ      502608
4016      015460      LET R4 := R4 - #2      ; ADJUST THE CODE FOR TABLE INDEX.
      015460      162704      000002      SUB      #2,R4
4017      015464      LET R2 := R5 + BINC(R4) + #CNTBGN ; R2 = ADR OF BYTE COUNT LSW.
      015464      010502      MOV      R5,R2
      015466      066402      015722      ADD      BINC(R4),R2
      015472      062702      002560      ADD      #CNTBGN,R2
4018      015476      LET (R2) := (R2) + BRFCNT      ; ADD BRFCNT TO LSW.
      015476      063712      003424      ADD      BRFCNT,(R2)
4019      015502      IF MSGPKT*MS.RFC LOS BRFCNT THEN ; IF THE RFC IS LOWER OR THE SAME AS BRFCNT THEN:
      015502      023737      002344      003424      CMP      MSGPKT*MS.RFC,BRFCNT
      015510      101002      BHI      502618
4020      015512      LET (R2) := (R2) - MSGPKT*MS.RFC ; SUBTRACT RFC FROM EXPECTED BRFC.
      015512      163712      002344      SUB      MSGPKT*MS.RFC,(R2)
4021      015516      ENDDIF
      015516
4022      015516      LET R3 := R2 + #10      ; R3 = ADR OF 2ND WORD.
      015516      010203      MOV      R2,R3
      015520      062703      000010      ADD      #10,R3
4023      015524      WHILE (R2) GT #999. DO
      015524      021227      001747      502618:
      015530      003404      CMP      (R2),#999.
      015532      015532      BLE      502638
4024      015532      LET (R2) := (R2) + #1000. ; UPDATE BYTE COUNT
      015532      162712      001750      SUB      #1000.,(R2)
4025      015536      LET (R3) := (R3) + #1      ; 2ND WORD.
      015536      005213      INC      (R3)
4026      015540      ENDDO
      015540      000771      BR      502628
      015542      502638:
4027      015542      LET R2 := R3 + #10      ; R2 = ADR OF 3RD WORD.
      015542      010302      MOV      R3,R2
      015544      062702      000010      ADD      #10,R2
4028      015550      WHILE (R3) GT #999. DO
      015550      021327      001747      502648:
      015554      003404      CMP      (R3),#999.
      015556      015556      BLE      502658
4029      015556      LET (R3) := (R3) + #1000. ; UPDATE BYTE COUNT
      015556      162713      001750      SUB      #1000.,(R3)
4030      015562      LET (R2) := (R2) + #1      ; 3RD WORD.
      015562      005212      INC      (R2)

```

```

4031 015564          ENDDO
      015564 000771
      015566
4032 015566          LET R3 := R2 * #10          ;R3 = ADR OF 4TH WORD.
      015566 010203          50265$:
      015570 062703 000010          MOV R2,R3
      015574          WHILE (R2) GT #999. DO          ADD #10,R3
      015574          50266$:
      015574 021227 001747          CMP (R2),#999.
      015600 003404          BLE 50267$
4034 015602          LET (R2) := (R2) * #1000.      ;UPDATE BYTE COUNT
      015602 162712 001750          SUB #1000.,(R2)
4035 015606          LET (R3) := (R3) * #1          ;4TH WORD.
      015606 005213          INC (R3)
4036 015610          ENDDO
      015610 000771          BR 50266$
      015612          50267$:
4037 015612          IFB RWERR NE #0 THEN          ;IF R/W ERROR, UPDATE ERROR COUNT.
      015612 105737 003473          TSTB RWERR
      015616 001440          BEQ 50270$
4038 015620          LET R2 := R5 * EINC(R4) * #WRREC ;R2 = ADR OF COUNTER.
      015620 010502          MOV R5,R2
      015622 066402 015730          ADD EINC(R4),R2
      015626 062702 002720          ADD #WRREC,R2
4039 015632          IFB UNREC NE #0 THEN          ;IS THE ERROR UNRECOVERABLE?
      015632 105737 003474          TSTB UNREC
      015636 001404          BEQ 50271$
4040 015640          LET R2 := R2 * #10          ;YES, POINT TO NEXT COUNTER.
      015640 062702 000010          ADD #10,R2
4041 015644          LET (R2) := (R2) * #1          ;UPDATE THE ERROR COUNTER
      015644 005212          INC (R2)
4042 015646          ELSE          ;ELSE IF ERROR IS RECOVERABLE:
      015646 000424          BR 50272$
      015650          50271$:
4043 015650          LET (R2) := (R2) * #1          ;UPDATE THE ERROR COUNTER
      015650 005212          INC (R2)
4044 015652          IFB IREC EQ #0 THEN          ;IF ERROR RECOVERY IS ENABLED:
      015652 105737 002210          TSTB IREC
      015656 001020          BNE 50273$
4045 015660          IFB DROPED EQ #0 ANDB ERCVER NE #0 THEN ;IF UNIT HAS NOT BEEN DROPPED:
      015660 105737 003527          TSTB DROPED
      015664 001015          BNE 50274$
      015666 105737 002207          TSTB ERCVER
      015672 001412          BEQ 50274$
4046 015674          PRINTB #NURTY1,RETRYC          ;PRINT # OF RETRIES TO RECOVER
      015674 013746 003464          MOV RETRYC,(SP)
      015700 012746 005155          MOV #NURTY1,(SP)
      015704 012746 000002          MOV #2,(SP)
      015710 010600          MOV SP,R0
      015712 104414          TRAP C:PNTB
      015714 062706 000006          ADD #6,SP
4047 015720          ENDF          ;PROVIDED PRINT HAS BEEN ENABLED
      015720          50274$:
4048 015720          ENDF          50273$:
4049 015720          ENDF          50272$:
      015720

```

GLOBAL AREAS MACRO M1200 21 MAR 84 16:45 PAGE 88-2
GLOBAL SUBROUTINES SECTION

SEQ 0113

```
4050 015720          ENDIF
      015720
4051 015720          ENDIF
      015720
4052 015720          ENDIF
      015720
4053 015720 000207    RTS PC
4054                                INDEXES TO BYTE COUNTERS.
4055 015722 000000    ; BINC: 0 ;WRITE.
4056 015724 000040    40 ;READ REV.
4057 015726 000100    100 ;READ FWD.
4058                                INDEXES TO READ/WRITE ERROR COUNTERS.
4059 015730 000000    ; EINC: 0 ;WRITE.
4060 015732 000020    20 ;READ REV.
4061 015734 000040    40 ;READ FWD.
4062
4063
4064
```

502701:
502601:
502571:

```

4066                                     .SBTTL RAMER - READ AND DISPLAY SELECTED RAM
4067                                     ;*
4068                                     ;ROUTINE TO READ THE SELECTED RAM LOCATIONS
4069                                     ;
RAMER:: MOV      R5, -(SP)
        MOV      R4, -(SP)
        MOV      R3, -(SP)
        MOV      R2, -(SP)
        MOV      R1, -(SP)
        MOV      @RAMDATA, R1      ;ADDRESS TO SAVE THE RAM DATA
        MOV      RAMHLD, R2      ;BYTE ADDRESS OF THE FIRST RAM DATA
        MOV      RAMSIZ, R3      ;SET THE SIZE OF THE READ UP
        MOV      TSDB(R5), R4      ;MOV THE TSDB ADDRESS INTO R4
        INC      R4      ;ADD 1 TO IT
10$:   NOP
        JSR      PC, WSSR      ;WAIT FOR THE SSR TO SET
        MOV      R2, (R4)      ;SELECT NEXT RAM ADDRESS
        JSR      PC, WSSR      ;WAIT FOR SSR TO SET
        MOV      @TSBA(R5), (R1) ;READ THE RAM DATA
20$:   ADD      @1, R2      ;ADDRESS OF THE NEXT RAM LOCATION
        SOB      R3, 10$      ;NUMBER OF LOCATIONS COUNTER
        MOV      RAMSIZ, R4      ;GET THE RAM SIZE
        MOV      RAMHLD, R2      ;GET THE STARTING RAM ADDRESS
        ADD      R2, R4      ;CALCULATE THE END ADDRESS
        SUB      @1, R4      ;CORRECT VALUE OF PRINTOUT
        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$PNTX
        ADD      @10, SP
4092   MOV      @RAMDATA, R1      ;ADDRESS OF WHERE RAM DATA IS
4093   MOV      RAMSIZ, R3      ;THE SIZE OF THE RAM FIELD READ
4094   CLR      R4      ;NO EXTRA DATA LEFT OVER
4095   MOV      (R1), R4      ;PICK UP BYTE OF RAM DATA
4096   BIC      @177400, R4      ;GET RID OF SIGN EXTEND
4097   PRINTX   @RAMPD, R4      ;"010 211 111 222 377 000 123 134 ETC."
        MOV      R4, (SP)
        MOV      @RAMPD, -(SP)
        MOV      @2, (SP)
        MOV      SP, R0
        TRAP     C$PNTX
        ADD      @6, SP
4098   SOB      R3, 30$      ;LOOP UNTIL ALL PRINTED
4099   MOV      (SP), R1
4100   MOV      (SP), R2
4101   MOV      (SP), R3
4102   MOV      (SP), R4
4103   MOV      (SP), R5
4104   RTS      PC      ;RETURN
4105   ; IF A WRITE/VERIFY COMMAND IS ISSUED, CONTROL IS THEN
4106   ; TRANSFERRED TO THIS SUBROUTINE TO READ REVERSE, CHECK DATA,
4107   ; READ FORWARD, CHECK DATA, THEN CONTINUE TO NEXT COMMAND.
4108   ; INPUTS:
4109   ; OUTPUTS:

```

4110					:	REGISTERS:			
4111					:	CALLS:	VFEXC.		
4112									
4113	016142				VFYDAT::	LET CMPDAT :- #0		;BTL	
	016142	005037	003410						CLR CMPDAT
4114	016146					IFB VFYFLG NE #0 THEN		;IF DATA IS TO BE VERIFIED:	
	016146	105737	003522						TSTB VFYFLG
	016152	001437							BEQ 50275\$
4115	016154					IFB STREAM EQ #0 THEN			TSTB STREAM
	016154	105737	003532						BNE 50276\$
	016160	001015						;SAVE THE PREVIOUS COMMAND WORD.	
4116	016162					LET PCMDWD := CMDWRD			MOV CMDWRD,PCMDWD
	016162	013737	003426	003432				;COMMAND IS READ REV.	
4117	016170					LET CMDWRD := #RDR			MOV #RDR,CMDWRD
	016170	012737	104401	003426				;SET UP CMD LOGGING INDEX.	
4118	016176					LET CMDLG := #4			MOV #4,CMDLG
	016176	012737	000004	003434				;GO READ ALL THE RECORDS REV.	
4119	016204	004737	016254			JSR PC,VFEXC			
4120	016210					LET CMPDAT := #0		;BTL	
	016210	005037	003410						CLR CMPDAT
4121	016214					ENDIF			50276\$:
	016214								
4122	016214					IF DEVTBL(R5) NE #NINUSE THEN			CMP DEVTBL(R5),#NINUSE
	016214	026527	002536	177774					BEQ 50277\$
	016222	001413						;SAVE THE PREVIOUS COMMAND WORD.	
4123	016224					LET PCMDWD := CMDWRD			MOV CMDWRD,PCMDWD
	016224	013737	003426	003432				;COMMAND IS READ FWD.	
4124	016232					LET CMDWRD := #RDF			MOV #RDF,CMDWRD
	016232	012737	104001	003426				;SET UP CMD LOGGING INDEX.	
4125	016240					LET CMDLG := #6			MOV #6,CMDLG
	016240	012737	000006	003434				;GO READ ALL RECORDS FWD.	
4126	016246	004737	016254			JSR PC,VFEXC			
4127	016252					ENDIF			50277\$:
	016252								
4128	016252					ENDIF			50275\$:
	016252								
4129	016252	000207				RTS PC		;RETURN.	

```

4131      ; SUBROUTINE TO EXECUTE THE READ AND VERIFY, FORWARD OR REVERSE.
4132      ; INPUTS:
4133      ; OUTPUTS:
4134      ; REGISTERS: R2
4135      ; CALLS: CMDAC, FIRSTU, VFISU, NEXTU, CKHAE.
4136
4137 016254 VFEXC:: LET CMDPKT := CMDWRD CLR.BY #BRF.C ;COMMAND PACKET = READ REV OR FWD.
016254 013737 003426 002314 MOV CMDWRD,CMDPKT
016262 042737 004000 002314 BIC #BRF.C,CMDPKT
4138 016270 IFB SWBFLG NE #0 THEN ;IF BYTES ARE TO BE SWAPPED:
016270 105737 003524 TSTB SWBFLG
016274 001403 BEQ 50300$
4139 016276 LET CMDPKT := CMDPKT SET.BY #SWB.C ;SET SWAB BIT IN CMD PACKET.
016276 052737 010000 002314 BIS #SWB.C,CMDPKT
4140 016304 ENDIF
016304
4141 016304 LET CMDSAV := CMDPKT ;SAVE COMMAND PACKET 1ST WORD.
016304 013737 002314 003430 MOV CMDPKT,CMDSAV
4142 016312 013737 003416 002316 MOV DATARD,CMDPKT+CP.ADL ;SAVE BUFFER START ADDRESS.
4143 016320 005037 003420 LET NCNT := #0 ;CLEAR NUMBER OF OPERATIONS.
4144 016324 WHILE NCNT LT NCNT1 DO ;WHILE THERE ARE RECORDS REMAINING:
016324 023737 003420 003422 50301$:
016332 002101 CMP NCNT,NCNT1
016332 004737 007600 BGE 50302$
4145 016334 004737 007600 JSR PC,CMDAC ;STORE CMD ASCII IN ERROR MSG.
4146 016340 105737 003532 TSTB STREAM ;CHECK IF WE ARE STREAMING
4147 016344 001006 BNE 1$ ;BRANCH OVER DEVTBL CHECK. THIS ENABLES
4148 ;US TO TEST ONE DRIVE AT A TIME.
4149 016346 004737 017300 JSR PC,FIRSTU ;SET UP FOR FIRST UNIT.
4150 016352 WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE DEVICES REMAINING:
016352 026527 002536 177777 50303$:
016360 001445 CMP DEVTBL(R5),#END
4151 016362 032737 000400 003426 1$: IF #MOD.CO SETIN CMDWRD THEN ;IF CMD IS REVERSE THEN:
016370 001421 BEQ 50304$
4152 016372 IF #X0.BOT NOTSETIN EOTFLG(R5) THEN ;IF NOT AT BOT
016372 032765 000002 003506 BIT #X0.BOT,EOTFLG(R5)
016400 001014 BNE 50306$
4153 016402 IF #X0.EOT SETIN EOTFLG(R5) THEN ;BUT IF AT EOT
016402 032765 000001 003506 BIT #X0.EOT,EOTFLG(R5)
016410 001406 BEQ 50307$
4154 016412 IFB ALLEOT NE #0 THEN ;AND ALL OTHERS AT EOT
016412 105737 003531 TSTB ALLEOT
016416 001402 BEQ 50310$
4155 016420 004737 016540 JSR PC,VFISU ;THEN READ VERIFY
4156 016424 ENDIF ;IF NOT ALL AT EOT, FREEZE UNIT(S) AT E
016424 50310$:
4157 016424 ELSE ;IF NOT AT BOT AND
016424 000402 BR 50311$
016426 50307$:
4158 016426 004737 016540 JSR PC,VFISU ;NOT AT EOT, READ VFY
4159 016432 ENDIF
016432 50311$:
4160 016432 ENDIF
016432 50306$:

```

```

4161 016432      ELSE
      016432 000412
      016434
4162 016434      IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN
      016434 032765 000001 003576
      016442 001404
      016444 032737 000001 003426
      016452 001002
      016454
4163
4164 016454 004737 016540      JSR PC,VFISU
4165 016460      ENDIF
      016460
4166 016460      ENDIF
      016460
4167 016460 105737 003532      TSTB STREAM
4168 016464 001003      BNE 2$
4169 016466 004737 017346      JSR PC,NEXTU
4170 016472      ENDDO
      016472 000727
      016474
4171 016474 004737 017706      2$: JSR PC,CKHAE
4172 016500      IF DEVTBL(R5) EQ #NINUSE THEN
      016500 026527 002536 177774
      016506 001005
4173 016510      LET NCNT := NCNT1 - #1
      016510 013737 003422 003420
      016516 005337 003420
4174 016522      ENDIF
      016522
4175 016522      LET NCNT := NCNT + #1
      016522 005237 003420
4176 016526      LET PCMDWD := CMDWRD
      016526 013737 003426 003432
4177
4178 016534      ENDDO
      016534 000673
      016536
4179 016536 000207      RTS      PC

```

```

;ELSE IF CMD IS NOT REVERSE:
BR      50312$
50305$:
BIT      #X0.EOT,EOTFLG(R5)
BEQ      50313$
BIT      #CMD.CO,CMDWRD
BNE      50314$
50313$:
;IF NOT AT EOT OR NOT A MOTION CMD THEN:
;ISSUE CMD, CHECK STATUS AND DATA.
50314$:
50312$:
;CHECK FOR TEST OF ON UNIT AT A TIME.
;BRANCH, IF STREAMING TESTS.
;GO FIND THE NEXT UNIT.
BR      50303$
50304$:
;CHECK FOR HALT AFTER EACH CMD.
;IF DRIVES BEEN DROPPED EXIT
CMP      DEVTBL(R5),#NINUSE
BNE      50315$
MOV      NCNT1,NCNT
DEC      NCNT
50315$:
;UPDATE THE RECORD COUNT.
INC      NCNT
;SAVE PREVIOUS COMMAND WORD.
MOV      CMDWRD,PCMDWD
BR      50301$
50302$:
;RETURN.

```

```

4181      ; SUBROUTINE TO ISSUE COMMAND, AWAIT INTERRUPT,
4182      ; CHECK STATUS, CHECK DATA.
4183      ; INPUTS:
4184      ; OUTPUTS:
4185      ; REGISTERS: R2
4186      ; CALLS: EXECUTE, GOWAIT, CKDATA.
4187
4188      VFISU::
4189      LET R2 := DATARD * #8.      ;INIT READ BUFFER POINTER.
                                MOV DATARD,R2
                                ADD #8.,R2
016540 013702 003416
016544 062702 000010
4190 016550      WHILE R2 NE DATARD DO      ;UNTIL 8 BYTES HAVE BEEN SET.
                                503161:
016550      020237 003416      CMP R2,DATARD
016554 001403      BEQ 503171
4191 016556      LET (R2) := # 1      ;INIT READ BUFFER.
                                MOV # 1. (R2)
4192 016562      012742 177777      ENDDO
                                BR 503161
016564      503171:
4193 016564 004737 010654      JSR PC,EXECUTE      ;GO EXECUTE THE COMMAND.
4194 016570      IFB DROPED EQ #0 THEN      ;IF UNIT HAS NOT BEEN DROPPED THEN:
                                TSTB DROPED
016570 105737 003527      BNE 503201
016574 001002
4195 016576 004737 011274      JSR PC,GOWAIT      ;GO WAIT FOR DONE BIT.
4196 016602      ENDIF
                                503201:
016602      IFB DROPED EQ #0 THEN      ;IF UNIT HAS NOT BEEN DROPPED THEN:
                                TSTB DROPED
016606 001025      BNE 503211
4198 016610      IF #X0.BOT NOTSETIN FOTFLG(R5) THEN      ;WHEN NOT REVERSED INTO BOT, THEN
                                BIT #X0.BOT.EOTFLG(R5)
016610 032765 000002 003506      BNE 503221
016616 001021
4199 016620      IF L1TEST NE #3 THEN
                                CMP L1TEST,#3
016620 023727 002114 000003      BEQ 503231
016626 001403
4200 016630 004737 016664      JSR PC,CKDATA      ;GO VERIFY DATA.
4201 016634      ELSE
                                BR 503241
016634 000412      503231:
016636      INC CMPDAT      ;ONE MORE XFER BEFORE A COMPARISON
4202 016636 005237 003410      IF CMPDAT GT DATRAT THEN
                                CMP CMPDAT,DATRAT
4203 016642      023737 003410 003412      BLE 503251
016650 003404
4204 016652 004737 016664      JSR PC,CKDATA
4205 016656      LET CMPDAT := #0
                                CLR CMPDAT
016656 005037 003410
4206 016662      ENDIF
                                503251:
016662
4207 016662      ENDIF
                                503241:
016662
4208 016662      ENDIF
                                503221:
016662
4209 016662      ENDIF
                                503211:
016662
4210 016662 000207      RTS PC
4211

```

```

4213      SUBROUTINE TO COMPARE DATA BETWEEN READ AND WRITE BUFFERS
4214      AND PRINT ERROR MESSAGE ON MISCOMPARE.
4215      INPUTS:
4216      OUTPUTS:
4217      REGISTERS:      R2, R3, R4.
4218      CALLS:          GCMOA
4219
4220      CKDATA:: LET R3 := BRCNT MSGPKT*MS.RFC ;COMPUTE REC LENGTH READ
                                MOV      BRCNT,R3
                                SUB      MSGPKT*MS.RFC,R3
4221      IF R3 EQ #0 THEN                ;WHEN NO DATA RECEIVED
                                TST      R3
                                BNE      50326$
4222      ERRMRD 17,WTVERM,DTAERM          ;PRINT ERROR AND EXIT
                                TRAP     C$ERRMRD
                                .WORD    17
                                .WORD    WTVERM
                                .WORD    DTAERM
4223      PRINTB #DTAER4                  ;COMPARE ROUTINE
                                MOV      #DTAER4, (SP)
                                MOV      #1, (SP)
                                MOV      SP,R0
                                TRAP     C$PNTB
                                ADD      #4,SP
4224      ELSE
                                BR      50327$
4225      IF R3 HI BRCNT THEN              ;WHEN REC READ IS LONGER
                                CMP      R3,BRCNT
                                BLOS     50330$
4226      ERRMRD 17,WTVERM,DTAERM          ;THAN EXPECTED. PRINT
                                TRAP     C$ERRMRD
                                .WORD    17
                                .WORD    WTVERM
                                .WORD    DTAERM
4227      PRINTB #DTAER5,CMPKPT*CP.CNT    ;AN ERROR MESSAGE
                                MOV      CMPKPT*CP.CNT, (SP)
                                MOV      #DTAER5, (SP)
                                MOV      #2, (SP)
                                MOV      SP,R0
                                TRAP     C$PNTB
                                ADD      #6,SP
4228      ELSE                          ;AND EXIT ROUTINE
                                BR      50331$
4229      LET CKDCNT := R3 - #1            ;SAVE VERIFICATION LENGTH 1.
                                MOV      R3,CKDCNT
                                DEC      CKDCNT
4230      CLR CKDFF                      ;CLEAR # OF BYTES IN ERROR COUNTER.
4231      CLR R2                          ;INIT BYTE COUNTER
4232      LET R3 := DATAW                ;GET WRITE BUFFER ADDRESS.
                                MOV      DATAW,R3
4233      LET R4 := DATARD                ;GET READ BUFFER ADDRESS.
                                MOV      DATARD,R4
4234      IFB T1SWB NE #0 THEN            ;WHEN RUNNING TEST1-SUB 12.
                                TSTB     T1SWB
                                BEQ      50332$

```

4235	017032	000313		SWAB (R3)		;SWAP FIRST WORD OF WRT BFR
4236	017034			ENDIF		;WHICH CONTAINS THE RECORD COUNT
	017034					50332\$:
4237	017034			REPEAT		;REPEAT UNTIL ALL DATA IS COMPARED:
	017034					50333\$:
4238	017034			IF R2 EQ CKDCNT THEN		;IF THIS IS THE LAST BYTE THEN:
	017034	020237	017274			CMP R2,CKDCNT
	017040	001011				BNE 50334\$
4239	017042			IFB SWBFLG NE #0 THEN		;IF BYTE SWAPPING IS ENABLED THEN:
	017042	105737	003524			TSTB SWBFLG
	017046	001406				BEQ 50335\$
4240	017050			IF #BIT00 NOTSETIN CKDCNT THEN		;IF RECORD LENGTH IS ODD THEN:
	017050	032737	000001 017274			BIT #BIT00,CKDCNT
	017056	001002				BNE 50336\$
4241	017060	105723		TSTB (R3).		;LAST BYTE WILL BE IN
4242	017062	105724		TSTB (R4).		;THE UPPER BYTE.
4243	017064			ENDIF		
	017064					50336\$:
4244	017064			ENDIF		
	017064					50335\$:
4245	017064			ENDIF		
	017064					50334\$:
4246	017064	121314		CMPB (R3),(R4)		;ARE THEY EQUAL.
4247	017066	001452		BEQ 3\$		;BR IF SO.
4248	017070	005737	017276	TST CKDFF		;1 ST TIME THRU?
4249	017074	001010		BNE 2\$		;BR IF NOT.
4250	017076	005265	003300	INC VFYCNTR(R5)		;INC THE VERIFY ERROR COUNTER.
4251	017102	005265	003310	INC WRDCNT(R5)		;INC THE HARD ERROR COUNT.
4252	017106			ERRHRD 17,WTVERM,DTAERM		;REPORT WRITE/VERIFY ERROR.
	017106	104456				TRAP C\$ERRHRD
	017110	000021				.WORD 17
	017112	004246				.WORD WTVERM
	017114	005460				.WORD DTAERM
4253	017116			LET CKDFF := CKDFF + #1		;INCREMENT # OF BYTES IN ERROR.
	017116	005237	017276			INC CKDFF
4254	017122	111437	003444	MOVB (R4),TIME1		;SAVE WAS DATA FOR TYP0UT.
4255	017126	042737	177400 003444	BIC #177400,TIME1		;CLEAR GARBAGE.
4256	017134	111337	003446	MOVB (R3),TIME2		;SAVE SHOULD BE DATA FOR TYP0UT.
4257	017140	042737	177400 003446	BIC #177400,TIME2		;CLEAR GARBAGE.
4258	017146			IF CKDFF LT #11. THEN		;IF ERROR BYTE COUNT IS LESS THAN 11:
	017146	023727	017276 000013			CMP CKDFF,#11.
	017154	002017				BGE 50337\$
4259	017156			PRINTX #DTAER2,R2,<B,TIME1>,<B,TIME2>		;PRINT EXP. ACT DATA.
	017156	005046				CLR -(SP)
	017160	153716	003446			BISB TIME2,(SP)
	017164	005046				CLR -(SP)
	017166	153716	003444			BISB TIME1,(SP)
	017172	010246				MOV R2,-(SP)
	017174	012746	004761			MOV #DTAER2,-(SP)
	017200	012746	000004			MOV #4,-(SP)
	017204	010600				MOV SP,R0
	017206	104415				TRAP C\$PNTX
	017210	062706	000012			ADD #12,SP
4260	017214			ENDIF		
	017214					50337\$:
4261	017216	105723		TSTB (R3).		;UPDATE WRITE BUFFER ADDRESS.
4262	017216	105724		TSTB (R4).		;UPDATE READ BUFFER ADDRESS.

```

4263 017220 105722          TSTB (R2),
4264 017222          UNTIL R2 GT CKDCNT          ;UPDATE BYTE COUNTER.
          017222 020237 017274          ;END OF DATA COMPARE REPEAT LOOP.
          017226 003702          CMP      R2,CKDCNT
4265 017230          LET CKDCNT := CKDCNT + #1    ;CKDCNT EQUALS RECORD LENGTH.
          017230 005237 017274          BLE      50333$
4266 017234          IF CKDFF NE #0 THEN          INC      CKDCNT
          017234 005737 017276          ;IF COMPARE ERROR HAS OCCURED THEN:
          017240 001414          TST      CKDFF
4267 017242          PRINTB # TAER3,CKDFF,CKDCNT    ;PRINT # OF BYTES IN ERROR.
          017242 013746 017274          BEQ      50340$
          017246 013746 017276          MOV      CKDCNT,-(SP)
          017252 012746 005030          MOV      CKDFF,-(SP)
          017256 012746 000003          MOV      #DTAER3,-(SP)
          017262 010600          MOV      #3, (SP)
          017264 104414          MOV      SP,R0
          017266 062706 000010          TRAP     C$PNTB
          017272          ADD      #10,SP
4268 017272          ENDIF
          017272          50340$:
4269 017272          ENDIF
          017272          50331$:
4270 017272          ENDIF
          017272          50327$:
4271 017272 000207          RTS      PC          ;OTHERWISE, RETURN.
4272          4272          ;# OF BYTES TO BE VERIFIED -1.
4273 017274 000000          CKDCNT: .WORD 0    ;# OF BYTES IN ERROR COUNTER.
4274 017276 000000          CKDFF:  .WORD 0

```

```

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

```

```

4314      ;      SUBROUTINE TO DROP A DEVICE FROM THE TEST SEQUENCE.
4315      ;      INPUTS:
4316      ;      OUTPUTS:
4317      ;      REGISTERS:
4318      ;      CALLS:      MOVMSG, PRXST, LOG
4319
4320 017402      DROPU:: LET R5SAVE := R5
4321 017402      010537 003460      LET FTLCNT(R5) := FTLCNT(R5) + #1 ;INCREMENT THE FATAL ERROR COUNT.
4322 017406      005265 003320      LET R4 := MSGPKT+MS.XS3 CLR.BY #377 ;GET UDIAG ERROR CODE FROM XSTAT3.
4323 017412      013704 002354      LET R3 := MSGPKA(R5) ;ADR OF THIS UNIT'S MSG PACKET.
4324 017416      042704 000377      LET R2 := #0 ;CLR COUNTER.
4325 017422      016503 002506      WHILE R2 NE #MSGCNT DO ;WHILE THERE ARE MORE LOCATIONS:
4326 017426      005002      50345$:
4327 017430      020227 000016      CMP R2,#MSGCNT
4328 017434      001405      BEQ 50346$
4329 017436      012723 177777      LET (R3)+ := # -1 ;INIT THE MSG PACKET WITH ALL 1'S
4330 017442      062702 000002      LET R2 := R2 + #2 ;UPDATE COUNTER.
4331 017446      000770      ENDDO      ADD #2,R2
4332 017450      012775 002324 002456      LET @TSDB(R5) := #GSCPK ;INITIATE A GET STATUS COMMAND.
4333 017454      004737 011672      JSR PC,WSSR ;WAIT A WHILE FOR SSR=1
4334 017458      004737 011726      JSR PC,MOVMSG ;MOVE MSG PACKET TO COMMON AREA.
4335 017462      020427 157400      IF R4 EQ #X3.RNY THEN ;IF WE HAVE A CAPSTAN RUNAWAY THEN:
4336 017466      001005      ERRDF 16,RNYM,STAERM ;REPORT CAPSTAN RUN AW WITH TACH CNT.
4337 017470      104455      TRAP C$ERDF
4338 017474      000020      .WORD 16
4339 017478      004566      .WORD RNYM
4340 017482      005626      .WORD STAERM
4341 017504      ELSE ;ELSE IF NOT A RUNAWAY:
4342 017508      000402      BR 50350$
4343 017512      004737 017624      JSR PC,PRXST ;PRINT EXTENDED STATUS REGISTERS.
4344 017516      001404      ENDF
4345 017520      105737 003471      IFB RECLOG NE #0 THEN ;IF THE RECORD HAS BEEN LOGGED THEN:
4346 017524      105237 003527      LET DROPED := DROPED + #1 ;SET UNIT DROPPED FLAG.
4347 017528      004737 015436      JSR PC,LOG ;LOG DATA BYTES + RD/WR ERRORS.
4348 017532      001404      ENDF
4349 017536      104424      DORPT ;PRINT PERFORMANCE REPORT
4350 017540      DROPUA: IF PASCNT(R5) NE #0 THEN

```

H10

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

SEQ 0124

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

```

4362      ; SUBROUTINE TO PRINT EXTENDED STATUS REGISTERS.
4363      ;
4364      ; INPUTS:
4365      ; OUTPUTS:
4366      ; REGISTERS:
4367      ; CALLS:
4368      PRXST:: PRINTX @GETSTM
                                MOV      @GETSTM, (SP)
                                MOV      @1, (SP)
                                MOV      SP, R0
                                TRAP     C$PNTX
                                ADD      @4, SP
4369      PRINTX @STAERS, MSGPKT+MS.XS0, MSGPKT+MS.XS1, MSGPKT+MS.XS2, MSGPKT+MS.XS3
                                MOV      MSGPKT+MS.XS3, (SP)
                                MOV      MSGPKT+MS.XS2, -(SP)
                                MOV      MSGPKT+MS.XS1, -(SP)
                                MOV      MSGPKT+MS.XS0, -(SP)
                                MOV      @STAERS, (SP)
                                MOV      @5, (SP)
                                MOV      SP, R0
                                TRAP     C$PNTX
                                ADD      @14, SP
017624 012746 005241
017630 012746 000001
017634 010600
017636 104415
017640 062706 000004
017644 013746 002354
017650 013746 002352
017654 013746 002350
017660 013746 002346
017664 012746 006453
017670 012746 000005
017674 010600
017676 104415
017700 062706 000014
017704 000207
                                RTS PC
4370
4371
4372
4373
4374
4375      ; SUBROUTINE TO HALT AFTER EACH COMMAND.
4376      ;
4377      ; INPUTS:
4378      ; OUTPUTS:
4379      ; REGISTERS: R3, R4
4380      ; CALLS:
4381      CKHAE:: IFB HAE NE #0 THEN
                                ; IF HALT FLAG IS SET:
                                TSTB     HAE
                                BEQ      50355$
017706 105737 002206
017712 001430
                                ;
                                TSTB     MISCFC
                                BNE      50356$
4382      IFB MISCFC EQ #0 THEN
                                ; IS MANUAL INTERVENTION ALLOWED?
                                TRAP     C$MANI
                                BCC      CKHRTN
017714 105737 003537
017720 001023
                                ; BR IF NOT.
                                MOV      CMDWRD, R4
                                ; COMMAND WORD.
                                ; FETCH ADR OF CMD ASCII.
                                ; MOVE CMD ASCII
                                MOV      (R3)+, HALTM
                                MOV      (R3)+, HALTM+1
                                ; INTO MESSAGE.
                                MOV      (R3), HALTM+2
                                ; HALT WAIT FOR AN OPERATOR INPUT.
                                TRAP     C$GMAN
                                BR      10000$
                                .WORD    TIME1
                                .WORD    T$CODE
4383      MANUAL
017722 104450
                                BNCOMPLETE CKHRTN
4384      017724 103023
017726 013704 003426
017732 004737 007654
                                JSR PC, GCMOA
                                LET HALTM :B= (R3)+
                                LET HALTM+1 :B= (R3)+
                                LET HALTM+2 :B= (R3)
                                GMANIL HALTM, TIME1, 1, YES
4385      017726 013704 003426
4386      017732 004737 007654
4387      017736 112337 004124
4388      017742 112337 004125
4389      017746 111337 004126
4390      017752 104443
017754 000404
017756 003444
017760 000130

```

Address	Hex	Dec	Label	Instruction	Comment
4391	017762	004124			.WORD
	017764	000001			.WORD
	017766		10000\$:		
	017766		ELSE		
	017766	000402			BR 50357\$
	017770				50356\$:
4392	017770		LET MISCFC :B= #0		
	017770	105037 003537			CLRB MISCFC
4393	017774		ENDIF		50357\$:
	017774				
4394	017774		ENDIF		50355\$:
	017774				
4395	017774	000207	CKHRTN: RTS PC	;RETURN	

```

4397      ;      SUBROUTINE TO CREATE THE SEQUENCE FOR A WRITE TAPE MARK
4398      ;      COMMAND.  WILL EXECUTE COMMAND TO UUT.
4399      ;      INPUTS:
4400      ;      OUTPUTS: CMDSEQ
4401      ;      CALLS:  SETUP, CMDAC, EXECUTE, GOWAIT
4402
4403 017776      WRITEM::
4404 017776      LET R1 := #CMDSFQ
4405 020002      LET (R1)+ := #WTM      ;COMMAND      MOV      #CMDSEQ,R1
4406 020002      LET (R1)+ := #1      ;BRF      MOV      #WTM,(R1)+
4407 020006      LET (R1)+ := #1      ;ITERATIONS      MOV      #1,(R1)+
4408 020012      TST (R1)+      ;PATTERN      MOV      #1,(R1)+
4409 020016      LET (R1)+ := #END      ;TERMINATOR
4410 020020      LET R1 := #CMDSEQ      MOV      #END,(R1)+
4411 020024      JSR PC, SETUP      ;TOP OF BUFFER      MOV      #CMDSEQ,R1
4412 020030      JSR PC, CMDAC      ;SET UP THE TABLE
4413 020034      JSR PC, EXECUTE      ;LOAD THE ASCII
4414 020040      JSR PC, GOWAIT      ;ISSUE THE WTM COMMAND
4415 020050      RTS PC      ;WAIT FOR THE COMMAND TO FINISH
4416      .EVEN      ;RETURN TO CALLER
4417
4418 020052      ENDMOD

```

```

4430
4431      .TITLE MISCELLANEOUS SECTIONS
4432      .SBTTL  REPORT CODING SECTION
4441
4442 020052      BGNMOD
4443
4444      ;**
4445      ; THE REPORT CODING SECTION CONTAINS THE
4446      ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
4447      ;--
4448
4449 020052      BGNRPT
4450 020052      L$RPT::
4451
4456 020052      LET      RSSAVE := R5          ;SAVE CURRENT DEVICE POINTER.
4457 020056 010537 003460      MOV      R5,RSSAVE
4458 020062 004737 017300      JSR      PC,FIRSTU      ;FIND THE FIRST UNIT.
4459 020062      WHILE DEVTBL(R5) NE #END DO      ;WHILE THERE ARE MORE DEVICES:
4460 020062      50360$:
4461 020062 026527 002536 177777      CMP      DEVTBL(R5),#END
4462 020070 001562      BEQ      50361$
4463
4464      PRINTS      #RPT1A,DEVTBL(R5),PASCNT(R5),RECCNT(R5)
4465 020072 016546 003330      MOV      RECCNT(R5),-(SP)
4466 020076 016546 003260      MOV      PASCNT(R5),-(SP)
4467 020102 016546 002536      MOV      DEVTBL(R5),-(SP)
4468 020105 012746 020714      MOV      #RPT1A, -(SP)
4469 020112 012746 000004      MOV      #4, (SP)
4470 020116 010600      MOV      SP,R0
4471 020120 104416      TRAP      C$PNTS
4472 020122 062706 000012      ADD      #12,SP
4473
4474      PRINTS      #RPT1B,WRBC+30(R5),WRBC+20(R5),WRBC+10(R5),WRBC(R5)
4475 020126 016546 002560      MOV      WRBC(R5),-(SP)
4476 020132 016546 002570      MOV      WRBC+10(R5),-(SP)
4477 020136 016546 002600      MOV      WRBC+20(R5),-(SP)
4478 020142 016546 002610      MOV      WRBC+30(R5),-(SP)
4479 020146 012746 020771      MOV      #RPT1B, (SP)
4480 020152 012746 000005      MOV      #5, -(SP)
4481 020156 010600      MOV      SP,R0
4482 020160 104416      TRAP      C$PNTS
4483 020162 062706 000014      ADD      #14,SP
4484
4485      PRINTS      #RPT1C,RRBC+30(R5),RRBC+20(R5),RRBC+10(R5),RRBC(R5)
4486 020166 016546 002620      MOV      RRBC(R5),-(SP)
4487 020172 016546 002630      MOV      RRBC+10(R5),-(SP)
4488 020176 016546 002640      MOV      RRBC+20(R5),-(SP)
4489 020202 016546 002650      MOV      RRBC+30(R5),-(SP)
4490 020206 012746 021042      MOV      #RPT1C, -(SP)
4491 020212 012746 000005      MOV      #5, -(SP)
4492 020216 010600      MOV      SP,R0
4493 020220 104416      TRAP      C$PNTS
4494 020222 062706 000014      ADD      #14,SP
4495
4496      PRINTS      #RPT1D,RFBC+30(R5),RFBC+20(R5),RFBC+10(R5),RFBC(R5)
4497 020226 016546 002660      MOV      RFBC(R5),-(SP)
4498 020232 016546 002670      MOV      RFBC+10(R5),-(SP)
4499 020236 016546 002700      MOV      RFBC+20(R5),-(SP)
4500 020242 016546 002710      MOV      RFBC+30(R5),-(SP)
4501 020246 012746 021113      MOV      #RPT1D, -(SP)
4502 020252 012746 000005      MOV      #5, (SP)

```

```

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

```

```

MOV SP,RO
TRAP C$PNTS
ADD #14,SP
MOV RFREC(R5),-(SP)
MOV RRREC(R5),-(SP)
MOV WRREC(R5),-(SP)
MOV 0RPT1F,-(SP)
MOV #4,-(SP)
MOV SP,RO
TRAP C$PNTS
ADD #12,SP
MOV RFUNR(R5),-(SP)
MOV RRUNR(R5),-(SP)
MOV WRUNR(R5),-(SP)
MOV 0RPT1G,-(SP)
MOV #4,-(SP)
MOV SP,RO
TRAP C$PNTS
ADD #12,SP
TSTB BADTSW
BEQ 50362$
50362$:
MOV VFYCNTR(R5),-(SP)
MOV FTLCNT(R5),-(SP)
MOV HRDCNT(R5),-(SP)
MOV SCCNT(R5),-(SP)
MOV 0RPT1I,-(SP)
MOV #5,-(SP)
MOV SP,RO
TRAP C$PNTS
ADD #14,SP
BR 50360$
50361$:
MOV R5SAVE,R5
WORD J$JMP
WORD L10010 2 .
MOV WRTYCT(R5),-(SP)
MOV 0RPT1F,-(SP)
MOV #2,-(SP)
MOV SP,RO

```

```

020464 104416
020466 062706 000006
4480 020472 016537 002550 003516
020472 016537 002550 003516
4481 020500 017703 163012
020500 017703 163012
020504 006203
4482 020506 010346
020506 010346
020510 012746 021371
020514 012746 000002
020520 010600
020522 104416
020524 062706 000006
4483 020530 005703
020530 005703
020532 001457
4484 020534 020327 000024
020534 020327 000024
020540 101402
4485 020542 012703 000024
020542 012703 000024
4486 020546
020546
4487 020546 012746 005300
020546 012746 005300
020552 012746 000001
020556 010600
020560 104416
020562 062706 000004
4488 020566 013704 003516
020566 013704 003516
020572 062704 000002
4489 020576 005002
020576 005002
4490 020600
020600
4491 020600 011446
020600 011446
020602 012746 021451
020606 012746 000002
020612 010600
020614 104416
020616 062706 000006
4492 020622 005202
020622 005202
4493 020624 062704 000002
020624 062704 000002
4494 020630 020227 000012
020630 020227 000012
020634 001014
4495 020636 012746 005300
020636 012746 005300
020642 012746 000001
020646 010600
020650 104416
020652 062706 000004

LET BTPT := BTADDR(R5) ;BTPT IS BOTH THE BAD TAPE SPOT
LET R3 := @BTPT SHIFT 1 ;AND THE LOGGING INDEX
PRINTS @RPT1J,R3 ;PRINT # OF BAD TAPE SPOTS

IF R3 NE #0 THEN ;PRINT RECORD # IF BAD SPOTS DETECTED
    IF R3 HI #20. THEN ;
        LET R3 := #20. ;20 BAD SPOTS IS THE LIMIT
    ENDIF
    PRINTS @CRLFSP ;
    50364$:
        MOV @CRLFSP,-(SP)
        MOV #1,-(SP)
        MOV SP,R0
        TRAP C$PNTS
        ADD #4,SP

LET R4 := BTPT * #2 ;FETCH A BAD SPOT ID
LET R2 := #0 ;R2 = PRINT COUNT PER LINE: 10 MAX
REPEAT ;
    PRINTS @RPT1K,(R4) ;PRINT A BAD SPOT ID
    50365$:
        MOV (R4),-(SP)
        MOV @RPT1K,-(SP)
        MOV #2,-(SP)
        MOV SP,R0
        TRAP C$PNTS
        ADD #6,SP

LET R2 := R2 * #1 ;COUNT PRINTS
LET R4 := R4 * #2 ;NEXT
IF R2 EQ #10. THEN ;
    PRINTS @CRLFSP ;GO TO NEXT PRINT LINE
    50366$:
        MOV @CRLFSP,(SP)
        MOV #1,-(SP)
        MOV SP,R0
        TRAP C$PNTS
        ADD #4,SP

```

```

4496 020656          LET R3 := R3  #10.          ;ADJUST BAD SPOT COUNT
          020656 162703 000012          SUB          #10.,R3
4497 020662          LET R2 := R2 - #10.          ;ADJUST PRINT COUNT
          020662 162702 000012          SUB          #10.,R2
4498 020666          ENDIF
          020666          ;
4499 020666          UNTIL R2 EQ R3          ;LIMIT: # OF BAD SPOTS
          020666 020203          503661:
          020670 001343          CMP          R2,R3
          020672          BNE          503651
4500 020672          ENDIF
          020672          ;
4501 020672          PRINTS #CRLF          ;
          020672 012746 005275          MOV          #CRLF, (SP)
          020676 012746 000001          MOV          #1, -(SP)
          020702 010600          MOV          SP,R0
          020704 104416          TRAP          C1PNTS
          020706 062706 000004          ADD          #4,SP
4502 020712 000207          RTS PC
4503
4515          .NLIST BEX
4516 020714          045          116          045 RPT1A: .ASCIIZ /#NN#AUNIT #D1#S3#APASS:#D5#S3#ARECORD:#D5#N/
4517 020771          045          101          102 RPT1B: .ASCIIZ /#BYTES WRITTEN #D3#A,#Z3#A,#Z3#A,#Z3#N/
4518 021042          045          101          102 RPT1C: .ASCIIZ /#BYTES READ REV #D3#A,#Z3#A,#Z3#A,#Z3#N/
4519 021113          045          101          102 RPT1D: .ASCIIZ /#BYTES READ FWD #D3#A,#Z3#A,#Z3#A,#Z3#N/
4520 021163          045          123          062          .ASCIIZ /#S23#AWRT#S4#ARDR#S4#ARDF#N/
4521 021217          045          101          122 RPT1F: .ASCIIZ /#ARECOVERABLE ERRORS #D5#S2#D5#S2#D5#N/
4522 021270          045          101          125 RPT1G: .ASCIIZ /#AUNRECOVERABLE ERRORS #D5#S2#D5#S2#D5#N/
4523 021341          045          101          127 RPT1E: .ASCIIZ /#AWRITE RETRIES#S8#D5#N/
4524 021371          045          116          045 RPT1J: .ASCIIZ /#N#D2#A BAD SPOTS THIS PASS PRECEDING RECORD #:/
4525 021451          045          104          065 RPT1K: .ASCIIZ /#D5#S1/
4526 021460          045          101          123 RPT1I: .ASCIIZ /#ASPEC COND#S3#AHARD#S3#AFATAL#S3#ACOMPARE#N/
4527 021534          045          123          063          .ASCIIZ /#S3#D5#S3#D5#S3#D5#S3#D5#N#N/
4528 021571          045          116          045 TAPCAP: .ASCIIZ /#N#ATAPE IN CARTRIDGE MUST BE 600 IN LENGTH.#N#N/
4529          .LIST BEX
4530          .EVEN
4531 021654          ENDRPT
          021654
          021654 104425          L10010:
          021654          TRAP          C1RPT
4532
4533          ;LOAD DEVICE PROTECTION TABLE
4534          ;TABLE FOR SUPERVISOR TO IDENTIFY THE P TBL FOR THE LOAD DEV
4535          ;THE SUPERVISOR USES THE TBL TO WARN THE OPERATOR WHEN HE TRIES TO TEST THE LOAD DEV
4536
4537 021656          BGNPROT
          021656          L1PROT::
4538 021656 000000          .WORD 0          ;P TBL OFFSET OF ISSR, THE TK25 CSR
4539 021660 177777          .WORD 1          ;P TBL OFFSET OF MASS BUS UNIT #: 1 = NOT A MASS BUS DEV
4540 021662 177777          .WORD 1          ;P TBL OFFSET OF DRIVE #: 1 = NONE, ONE DRIVE PER UNIBUS AD
4541 021664          ENDPROT

```

```

4543 .SBTTL INITIALIZE SECTION
4544
4545
4546 ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
4547 ; AT THE BEGINNING OF EACH PASS.
4548 ;
4549
4550 021664 BGNINIT
021664 L$INIT::
4551
4561 021664 INIT10: IF #BIT0!BIT1 SET IN #CMDPKT THEN ; IF CMD PACKET IS NOT ON MODULO 4 BOUNDARY:
021664 032727 000003 002314 BIT #BIT0!BIT1, #CMDPKT
021672 001421 BEQ 50367$
4562 021674 ERRSF 1, CMDPKM ; PRINT ERROR MSG,
021674 104454 TRAP C$ERSF
021676 000001 .WORD 1
021700 004164 .WORD CMDPKM
021702 000000 .WORD 0
4563 021704 DELAY 20. ; GO TO SUPERVISOR, WAIT 2 SECONDS.
021704 012727 000024 MOV #20., (PC).
021710 000000 .WORD 0
021712 013727 002116 MOV L$DLY, (PC).
021716 000000 .WORD 0
021720 005367 177772 DEC -6(PC)
021724 001375 BNE .-4
021726 005367 177756 DEC -22(PC)
021732 001367 BNE .-20
4564 021734 BR INIT10
4565 021736 ENDIF
021736
4566
4567 021736 IFB CLRFLG NE #0 THEN ; IF CLR COUNTERS FLAG SET:
021736 105737 002204 TSTB CLRFLG
021742 001413 BEQ 50370$
4568 021744 105037 002204 CLRB CLRFLG ; INIT CLR FLAG.
4569 021750 LET R2 := #0
021750 005002 CLR R2
4570 021752 WHILE R2 NE #CNTLEN DO
021752 020227 000550 50371$:
021756 001405 CMP R2, #CNTLEN
4571 021760 LET WRBC(R2) := #0 ; CLR ALL STATISTICAL COUNTERS.
021760 005062 002560 BEQ 50372$
4572 021764 LET R2 := R2 + #2 CLR WRBC(R2)
021764 062702 000002 ADD #2, R2
4573 021770 ENDDO BR 50371$
021770 000170 50372$:
021772 021772 ENDIF
021772
4575
4576 021772 IFB RRANV NE #0 THEN ; IF RESET RANDOM VARIABLE FLAG IS SET THEN:
021772 105737 002205 TSTB RRANV
021776 001406 BEQ 50373$
4577 022000 LET RANB := #RANBC ; RESET RANDOM BASE #.
022000 012737 153624 003440 MOV #RANBC, RANB
4578 022006 LET RANS := #RANSC ; RESET RANDOM SAVE LOCATION.

```

4579	022006	012737	032561	003442	ENDIF	MOV	#RANSC,RANS
4580	022014				READEF #EF.START	503731:	
	022014					;READ START COMMAND EVENT FLAG.	
	022014	012700	000040			MOV	#EF.START,RO
	022020	104447				TRAP	C\$REFG
4581	022022				BNCOMPLETE INIT15	;BRANCH IF NOT STARTING.	
	022022	103030				BCC	INIT15
4582	022024				LET STAF LG :B= STAF LG * #1	;SET START COMMAND FLAG.	
	022024	105237	003534			INCB	STAF LG
4583	022030				LET HERE :B= #0	;RESET THE TEST 3 ASCII SEMAPHORE	
	022030	105037	003340			CLRB	HERE
4584	022034				LET R5 := #6		
	022034	012705	000006			MOV	#6,R5
4585	022040				REPEAT	;INITIATE UNIT NUMBER TABLE	
	022040					503741:	
4586	022040				LET DEVTBL(R5) := #NINUSE	;BY STORING NOT IN USE IN EACH LOCATION.	
	022040	012765	177774	002536		MOV	#NINUSE,DEVTBL(R5)
4587	022046				LET R5 := R5 - #2		
	022046	162705	000002			SUB	#2,R5
4588	022052				UNTIL R5 EQ #0		
	022052	005705				TST	R5
	022054	001371				BNE	503741
4589	022056				LET R5 := L\$UNIT SHIFT 1		
	022056	013705	002012			MOV	L\$UNIT,R5
	022062	006305				ASL	R5
4590	022064				REPEAT	;STORE ALL UNIT	
	022064					503751:	
4591	022064				LET R5 := R5 - #2	;NUMBERS IN DEVTBL.	
	022064	162705	000002			SUB	#2,R5
4592	022070				LET DEVTBL(R5) := R5 SHIFT -1		
	022070	010565	002536			MOV	R5,DEVTBL(R5)
	022074	006265	002536			ASR	DEVTBL(R5)
4593	022100				UNTIL R5 EQ #0		
	022100	005705				TST	R5
	022102	001370				BNE	503751
4594							
4595	022104				INIT15: READEF #EF.PWR	;HAS THERE BE A POWER FAILURE?	
	022104	012700	000034			MOV	#EF.PWR,RO
	022110	104447				TRAP	C\$REFG
4596	022112				BNCOMPLETE INIT16	;BRANCH IF NOT.	
	022112	103004				BCC	INIT16
4597	022114				LET STAF LG :B= STAF LG * #1	;IF SO - SET THE START FLAG.	
	022114	105237	003534			INCB	STAF LG
4598	022120				LET PWRFLG :B= PWRFLG * #1	;IF SO SET THE POWER FAIL FLAG.	
	022120	105237	003535			INCB	PWRFLG
4599							
4600	022124				INIT16: RFLAGS OPFLAG	;READ AND STORE FLAGS SET BY OPERATOR	
	022124	104421				TRAP	C\$RFLA
	022126	010037	003540			MOV	RO,OPFLAG
4601	022132				LET R3 := #0	;CLEAR EVENT FLAG	
	022132	005003				CLR	R3
4602	022134				IFB PWRFLG EQ #0 THEN	;IF POWER FAIL HAS NOT OCCURRED THEN:	
	022134	105737	003535			TSTB	PWRFLG
	022140	001020				BNE	503761
4603	022142				READEF #EF.NEW	;UPDATE PASS COUNT WHEN	
	022142	012700	000035			MOV	#EF.NEW,RO

022264	016546	002476				MOV	TSVCT(R5), -(SP)
022270	012746	000003				MOV	#3, -(SP)
022274	104437					TRAP	C\$SVEC
022276	062706	000010				ADD	#10, SP
4628	022302			IF R3 NE #0 THEN	;ACTUAL PASSCOUNT UPDATE PER R3	TST	R3
	022302	005703				BEG	50407\$
	022304	001410					
4629	022306			IF R3 LT #0 THEN		TST	R3
	022306	005703				BGE	50410\$
	022310	002003					
4630	022312			LET PASCNT(R5) := PASCNT(R5) + #1		INC	PASCNT(R5)
	022312	005265	003260				
4631	022316			ELSE		BR	50411\$
	022316	000403				50410\$:	
	022320						
4632	022320			LET PASCNT(R5) := #1		MOV	#1, PASCNT(R5)
	022320	012765	000001	003260			
4633	022326			ENDIF		50411\$:	
	022326						
4634	022326			ENDIF		50407\$:	
	022326						
4635	022326			ENDIF		50406\$:	
	022326						
4636	022326			LET RECCNT(R5) := #0	;CLEAR RECORD COUNT	CLR	RECCNT(R5)
	022326	005065	003330				
4637	022332	004737	017346	JSR PC, NEXTU	;DO IT FOR ALL DEVICES.		
4638	022336			ENDDO		BR	50404\$
	022336	000724				50405\$:	
	022340						
4639							
4640	022340			IF R2 EQ #0 THEN	;IF THERE ARE NO UNITS:	TST	R2
	022340	005702				BNE	50412\$
	022342	001026					
4641	022344			PRINTF #AUDRPM	;PRINT ALL UNITS DROPPED.	MOV	#AUDRPM, -(SP)
	022344	012746	004727			MOV	#1, (SP)
	022350	012746	000001			MOV	SP, R0
	022354	010600				TRAP	C\$PNTF
	022356	104417				ADD	#4, SP
	022360	062706	000004				
4642	022364			DELAY 20.	;GO TO SUPERVISOR, WAIT 2 SECONDS.	MOV	#20, (PC)
	022364	012727	000024			.WORD	0
	022370	000000				MOV	L\$DLY, (PC)
	022372	013727	002116			.WORD	0
	022376	000000				DEC	6(PC)
	022400	005367	177772			BNE	4
	022404	001375				DEC	22(PC)
	022406	005367	177756			BNE	20
	022412	001367					
4643	022414			BREAK	;GO TO SUPERVISOR, CHECK TTY.	TRAP	C\$BRK
	022414	104422					
4644	022416			DOCLN	;DO CLEAN CODE + ABORT PASS.	TRAP	C\$DCLN
	022416	104444					
4645	022420			ENDIF			
	022420						
4646						50412\$:	
4647	022420			SETPRI #PRI00	;LOWER CPU PRIORITY TO 0	MOV	#PRI00, R0
	022420	012700	000000				

Address	Hex	Dec	Label	Text	Comment
4648	022424	104441		IFB IREC EQ #0 AND #ADR NOTSETIN OPFLAG THEN	TRAP C:SPRI ;IF ERROR RECOVERY IS ENABLED
	022426	105737	002210		TSTB IREC
	022432	001145			BNE 50413:
	022434	032737	000020 003540		BIT #ADR,OPFLAG
	022442	001141			BNE 50413:
4649	022444	004737	017300	JSR PC,FIRSTU ;AND AUTO-DROP NOT CALLED, THEN SET UP FOR FIRST UNI	
4650	022450			WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:	
	022450	026527	002536 177777		50414: CMP DEVTBL(R5),#END
	022456	001533			BEQ 50415:
4651	022460			BEGIN COUNTER ;START 3.5 MINUTE COUNTER	
4652	022460			INCR TIME1 FROM #1 TO #25 BY #1	
	022460	012737	000001 003444		MOV #1,TIME1
	022466	000402			BR 50417:
	022470				50420: INC TIME1
	022474	005237	003444		50417: CMP TIME1,#25
	022474	023727	003444 000025		BGT 50421:
	022502	003106			
4653	022504			LET #TSD8(R5) := #GSCPK ;AND GET UNITS STATUS	
4654	022504	012775	002324 002456		MOV #GSCPK,#TSD8(R5)
	022512			DELAY 250. ;WAIT 25 MSEC.	
	022512	012727	000372		MOV #250,.(PC).
	022516	000000			.WORD 0
	022520	013727	002116		MOV L:DLY,.(PC).
	022524	000000			.WORD 0
	022526	005367	177772		DEC 6(PC)
	022532	001375			BNE .-4
	022534	005367	177756		DEC -22(PC)
	022540	001367			BNE .-20
4655	022542			IF #TS.SSR SETIN #TSSR(R5) THEN	
	022542	032775	000200 002466		BIT #TS.SSR,#TSSR(R5)
	022550	001420			BEQ 50422:
4656	022552			IF #TS.OFL NOTSETIN #TSSR(R5) THEN	
	022552	032775	000100 002466		BIT #TS.OFL,#TSSR(R5)
	022560	001001			BNE 50423:
4657	022562			LEAVE COUNTER ;EXIT COUNTER WHEN UNIT ON LINE	
	022562	000456			BR 50416:
4658	022564			ELSE	
	022564				50423: PRINTF #OFLINH,DEVTBL(R5) ;PRINT UNIT OFF LINE EVERY 10 SEC
4659	022564				MOV DEVTBL(R5),.(SP)
	022564	016546	002536		MOV #OFLINH,.(SP)
	022570	012746	005211		MOV #2,.(SP)
	022574	012746	000002		MOV SP,R0
	022600	010600			TRAP C:PNTF
	022602	104417			ADD #6,SP
	022604	062706	000006		
4660	022610			ENDIF	
	022610				50424: BR 50425:
4661	022610			ELSE	
	022610	000412			50422: BR 50425:
4662	022612			PRINTF #NRDYM,DEVTBL(R5)	
	022612	016546	002536		MOV DEVTBL(R5),.(SP)
	022616	012746	023544		MOV #NRDYM,.(SP)
	022622	012746	000002		MOV #2,.(SP)

Address	Offset	Hex	Label	Comment	Assembly	Register
022626	010600				MOV	SP,RO
022630	104417				TRAP	C\$PNTF
022632	062706	000006			ADD	#6,SP
4663	022636		ENDIF			
022636						
4664	022636		INCR TIME2 FROM #1 TO #13 BY #1		50425\$:	
022636	012737	000001	003446		MOV	#1,TIME2
022644	000402				BR	50426\$
022646					50427\$:	
022646	005237	003446			INC	TIME2
022652					50426\$:	
022652	023727	003446	000013		CMP	TIME2,#13
022660	003016				BGT	50430\$
4665	022662		DELAY 200.	;WAIT FOR UNIT TO BE SET ON LINE		
022662	012727	000310			MOV	#200.,(PC).
022666	000000				.WORD	0
022670	013727	002116			MOV	L\$DL1,(PC).
022674	000000				.WORD	0
022676	005367	177772			DEC	6(PC)
022702	001375				BNE	. 4
022704	005367	177756			DEC	22(PC)
022710	001367				BNE	. 20
4666	022712		BREAK	;ALLOW TERMINAL INTERRUPT		
022712	104422				TRAP	C\$BRK
4667	022714		ENDINC			
022714	000754				BR	50427\$
022716					50430\$:	
4668	022716		ENDINC			
022716	000664				BR	50420\$
022720					50421\$:	
4669	022720		END COUNTER			
022720					50416\$:	
4670	022720		IF TIME1 GT #25 THEN	;IF OFF LINE FOR 3.5 MINUTES		
022720	023727	003444	000025		CMP	TIME1,#25
022726	003404				BLE	50431\$
4671	022730	004737	011726			
4672	022734	004737	012762			
4673	022740		ENDIF			
022740					50431\$:	
4674						
4675	022740	004737	017346			
4676	022744		JSR PC,NEXTU	;REPEAT UNTIL ON LINE OR TIMED OUT.		
022744	000641		ENDDO	;SET UP FOR NEXT UNIT.		
022746					BR	50414\$
4677	022746		ENDIF		50415\$:	
022746						
4678	022746		IFB PWRFLG EQ #0 THEN		50413\$:	
022746	105737	003535			TSTB	PWRFLG
022752	001026				BNE	50432\$
4679	022754		MEMORY DATAWT	;REQUEST MEMORY FROM SUPER FOR RD/WR BUFFERS.		
022754	104431				TRAP	C\$MEM
022756	010037	003414			MOV	RO,DATAWT
4680	022762		LET DATARD := DATAWT * #DATCNT	;SET RD BFR AD		
022762	013737	003414	003416		MOV	DATAWT,DATARD
022770	062737	010000	003416		ADD	#DATCNT,DATARD
4681	022776		IF #DATAWT LT #DATCNT THEN	;WHEN NOT ENOUGH FREE MEMO AVAILABLE		
022776	027727	160412	010000		CMP	#DATAWT,#DATCNT

```

4682 023004 002011
023006 012746 023054 PRINTF #MEMOM ;WARN OPERATOR BGE 50433$
023012 012746 000001 MOV #MEMOM, (SP)
023016 010600 MOV #1, (SP)
023020 104417 MOV SP, R0
023022 062706 000004 TRAP C$PNTF
DOCLN ;AND ABORT PASS ADD #4, SP
4683 023026 DOCLN TRAP C$DCLN
023026 104444 ;DIAG MUST BE RE-LOADED IN A CPU WITH LARGER MEMO
4684 023030 ENDIF 50433$:
023030 ENDIF 50432$:
4685 023030
023030
4686 023030 LET CHGFLG :B= #0 ;CLR CHANGE CMD SEQ TBL FLAG.
4687 023030 105037 002217 CLR CHGFLG
4688 023034 LET R3 := #ENDFLG MOV #ENDFLG, R3
023034 012703 003534 ;CLEAR ALL FLAGS.
4689 023040 004737 011656 JSR PC, CLRERR ;CLEAR THE POWER FAIL FLAG.
4690 023044 LET PWRFLG :B= #0 CLR PWRFLG
023044 105037 003535
4691 023050 EXIT INIT
4692 023050 104432 TRAP C$EXIT
023052 000104 .WORD L10012-.
4704 023054 045 101 106 MEMOM: .ASCII /#AFREE MEMO TOO SMALL FOR RD WR BFRS#N/
023057 122 105 105
023062 040 115 105
023065 115 117 040
023070 124 117 117
023073 040 123 115
023076 101 114 114
023101 040 106 117
023104 122 040 122
023107 104 055 127
023112 122 040 102
023115 106 122 123
023120 045 116
4705 023122 045 101 122 .ASCIZ /#ARE LOAD IN LARGER MEMO#N/
023125 105 055 114
023130 117 101 104
023133 040 111 116
023136 040 114 101
023141 122 107 105
023144 122 040 115
023147 105 115 117
023152 045 116 000
4706 .EVEN
4707
4708 023156 ENDINIT
023156 L10012:
023156 104411 TRAP C$INIT

```

```

4710      .SBTTL  AUTO DROP SECTION
4711
4712      ;**
4713      ;SECTION EXECUTED AFTER THE INIT CODE WHEN 'ADR' FLAG IS SET BY OPERATOR
4714      ;SECTION CHECKS FOR A VALID INTERFACE LOCATION.  DROPS UNIT IF NO RESPONSE
4715      ;FROM INTERFACE
4716      ;*
4717
4718      BGNAUTO
4719      L$AUTO::
4720      023160 004737 017300      JSR PC,FIRSTU      ;FIND FIRST UNIT
4721      023164      WHILE DEVTBL(R5) NE #END DO      ;
4722      023164      026527 002536 177777      50434$:
4723      023172 001525      CMP      DEVTBL(R5),#END
4724      023174      LET TRAPD4 :B= #0      ;
4725      023174 105037 003536      BEQ      50435$
4726      023200      SETVEC #4,#TRAP4,#PRI07      ;SET VECTOR 4
4727      023200 012746 000340      CLR8      TRAPD4
4728      023204 012746 023574      MOV      #PRI07,(SP)
4729      023210 012746 000004      MOV      #TRAP4,(SP)
4730      023214 012746 000003      MOV      #4,(SP)
4731      023220 104437      MOV      #3,(SP)
4732      023222 062706 000010      TRAP      C$SVEC
4733      023226      LET R2 := #TSSR(R5)      ;ADDRESS TK25 INTERFACE
4734      023226 017502 002466      ADD      #10,SP
4735      023232      CLRVEC #4      ;CLEAR VECTOR AT 4
4736      023232 012'00 000004      MOV      #TSSR(R5),R2
4737      023236 104436      MOV      #4,R0
4738      023240      IFB TRAPD4 NE #0 THEN      TRAP      C$CVEC
4739      023240 105757 003536      TSTB      TRAPD4
4740      023244 001423      BEQ      50436$
4741      023246      LET FTLCNT(R5) := FTLCNT(R5) + #1
4742      023246 005265 003320      INC      FTLCNT(R5)
4743      023252      PRINTF #AUTODM,TSSR(R5)      ;PRINT ERROR
4744      023252 016546 002466      MOV      TSSR(R5),-(SP)
4745      023256 012746 023450      MOV      #AUTODM, -(SP)
4746      023262 012746 000002      MOV      #2,(SP)
4747      023266 010600      MOV      SP,R0
4748      023270 104417      TRAP      C$PNTF
4749      023272 062706 000006      ADD      #6,SP
4750      023276      LET DROPN := DEVTBL(R5)      ;SAVE # OF UNIT TO BE DROPPED.
4751      023276 016537 002536 017622      MOV      DEVTBL(R5),DROPN
4752      023304      LET R0 := R5 SHIFT 1      ;R0=LOGICAL DEVICE NUMBER
4753      023304 010500      MOV      R5,R0
4754      023306 006200      ASR      R0
4755      023310      DODU R0      ;DROP THE UNIT: EXEC BGNDU-ENDDU CODE IF IDL = 0
4756      023310 104451      TRAP      C$DODU
4757      023312      ELSE
4758      023312 000452      BR      50437$
4759      023314      LET #TSDB(R5) := #GSCPK      ;SEND GET STATUS COMMAND
4760      023314 012775 002324 002456      MOV      #GSCPK,#TSDB(R5)
4761      023322 004737 011672      JSR PC,WSSR      ;WAIT
4762      023326      IF #TS.SSR SETIN #TSSR(R5) THEN
4763      023326 032775 000200 002466      BIT      #TS.SSR,#TSSR(R5)

```


L11

MISCELLANEOUS SECTIONS MACRO M1200 21-MAR-84 16:45 PAGE 99-2
 AUTO DROP SECTION

SEQ 0141

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

4755					.EVEN
4756					
4757					: DEVICE BUS TRAP HANDLER
4758					: OUTPUT: TRAPD4 BYTE 1: TRAPED AT 4
4759					: 0: NO TRAP
4760					
4761	023574				TRAP4:: LET TRAPD4 :B= TRAPD4 + #1
	023574	105237	003536		
4762	023600	000002			RTI
4763					
4764					
4765					

INCB TRAPD4

```

4767      .SBTTL  CLEANUP CODING SECTION
4768
4769      ;**
4770      ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
4771      ; AT THE END OF EACH PASS.
4772      ;--
4773
4774      023602      BGNCLN
023602      L$CLEAN::
4775
4782
4783      023602      004737      017300      JSR      PC,FIRSTU      ;FIND FIRST UNIT.
4784      023606      WHILE DEVTBL(R5) NE #END DO
                                50443$:
                                CMP      DEVTBL(R5),#END
                                BEQ      50444$
4785      023616      004737      011672      JSR PC,WSSR      ;WAIT FOR UNIT READY OR TIMEOUT.
4786      023622      CLRVEC      TSVCT(R5)      ;RELEASE INTERRUPT VECTORS FOR ALL DEV.
                                MOV      TSVCT(R5),R0
                                TRAP     C$CVEC
                                50444$:
4787      023630      004737      017346      JSR      PC,NEXTU      ;FIND NEXT UNIT.
4788      023634      ENDDO
                                BR      50443$
                                50444$:
4789
4790      023636      EXIT      CLN
                                TRAP     C$EXIT
                                .WORD    L10014-.
4802
4803      .EVEN
4804      023642      ENDCLN
                                L10014:
                                TRAP     C$CLEAN
                                023642      104412

```

```

4806      .SBTTL  DROP UNIT SECTION
4807
4808      ;**
4809      ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
4810      ; TO NO LONGER BE TESTED.  THAT CODE SHALL BE EXECUTED WHEN DDDU
4811      ;MACRO IS CALLED WHILE IDU FLAG IS NOT SET BY OPERATOR
4812      ;--
4813
4814      023644      BGNDU
4815      023644      L$DU::
4821      023644      LET R5 := R0 SHIFT 1      ;R5 = LOGICAL DEVICE NUMBER X 2.
4822      023644      010005      MOV      R0,R5
4823      023646      006305      ASL      R5
4824      023650      LET DEVTBL(R5) := #NINUSE      ;SET NOT IN USE FLAG FOR THE DEVICE.
4825      023650      012765      177774      002536      MOV      #NINUSE,DEVTBL(R5)
4826      023656      PRINTF  #DROPDM,DROPN      ;PRINT DROP DEVICE MESSAGE
4827      023656      013746      017622      MOV      DROPN, (SP)
4828      023662      012746      004700      MOV      #DROPDM, (SP)
4829      023666      012746      000002      MOV      #2, -(SP)
4830      023672      010600      MOV      SP,R0
4831      023674      104417      TRAP     C$PNTF
4832      023676      062706      000006      ADD      #6,SP
4833      023702      EXIT      DU
4834      023702      000167      .WORD   J$JMP
4835      023704      000000      .WORD   L10015-2-.
4836
4837      .EVEN
4838      023706      ENDDU
4839      023706      L10015:
4840      023706      104453      TRAP     C$DU

```

```

4841      .SBTTL  ADD UNIT SECTION
4842
4843      ;**
4844      ; THE ADD UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
4845      ; TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING.  IF
4846      ; "EF.AUNIT" IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
4847      ;
4848
4849      023710      BGN AU
4850      023710      L1AU::
4851
4852      LET R5 := R0 SHIFT 1      ;R5 = LOGICAL DEVICE NUMBER X 2.
4853      MOV      R0,R5
4854      ISL      R5
4855
4856      LET DEVTBL(R5) := R0      ;STORE UNIT # IN DEVICE #ABLE.
4857      MOV      R0,DEVTBL(R5)
4858      GPHARD R0,R0      ;GET HARDWARE #ABLE FROM SUPER.
4859      TRAP     C#GPHRD
4860      LET TSSR(R5) := (R0)      ;SAVE TSSR ADDRESS.
4861      MOV      (R0),TSSR(R5)
4862      LET TSDB(R5) := (R0) - #2      ;SAVE TSDB ADDRESS.
4863      MOV      (R0),TSDB(R5)
4864      SUB      #2,TSDB(R5)
4865      LET TSVCT(R5) := (R0)      ;SAVE INTERRUPT VECTOR ADDRESS.
4866      MOV      (R0),TSVCT(R5)
4867      SETVEC TSVCT(R5),TS4INT(R5),#INTPRI      ;SET UP INTERRUPT PROCESSING CONDITIONS.
4868      MOV      #INTPRI,-(SP)
4869      MOV      TS4INT(R5),-(SP)
4870      MOV      TSVCT(R5),-(SP)
4871      MOV      #3,-(SP)
4872      TRAP     C#SVEC
4873      ADD      #10,SP
4874
4875      LET INTFLG(R5) := #0      ;CLEAR INTERRUPT FLAGS.
4876      CLR      INTFLG(R5)
4877
4878      EXIT      AU
4879
4880      .EVEN
4881      ENDAU
4882      L10016:
4883      TRAP     C#AU
4884
4885      ENDMOD

```

```

4887
4898 .TITLE HARDWARE TESTS
4899
4900 .SBTTL TEST 1: BASIC FUNCTIONS.
4901
4902
4903
4904
4905
4906 024004 BGNMOD
4907
4908 024004 BGNTST
4909 T1::
4910 024004 LET RANDOM :B= #0 ;CLR THE RANDOM OPERATIONS FLAG.
4911 024010 105037 003521 LET EXPBOT :B= #0 ;CLR EXPECT BOT FLAG.
4912 024010 105037 003520 ;CLRB RANDOM
4913 024014 BGNSUB ;SUBTEST 1 - SET CHAR, DRIVE INIT, GET STATUS.
4914 024014 104402 T1.1: TRAP C1BSUB
4915 024016 LET R2 := #BFSEQO ;ADR OF CMD SEQ.
4916 024016 012702 024642 ;SET UP CMD SEQ. MOV #BFSEQO,R2
4917 024022 004737 024616 JSR PC,BFSEQ ;EXECUTE CMD SEQ ON ALL DEVICES.
4918 024026 004737 006744 JSR PC,EXALL ;FIND THE FIRST UNIT.
4919 024032 004737 017300 JSR PC,FIRSTU ;FIND THE FIRST UNIT.
4920 024036 WHILE DEVTBL(R5) NE #END DO ;WHILE THERE ARE MORE DEVICES:
4921 024036 026527 002536 177777 50445: CMP DEVTBL(R5),#END
4922 024044 001434 BEQ 50446:
4923 024046 LET R2 := MSGPKA(R5) ;GET MSG PACKET ADR.
4924 024046 016502 002506 MOV MSGPKA(R5),R2
4925 024052 LET R2 := R2 + #12 ;GET XSTAT2 ADR.
4926 024052 062702 000012 ADD #12,R2
4927 024056 LET TS4CL(R5) := (R2) CLR BY #177400 ;STORE CODE LEVEL FROM DTR BYTE.
4928 024056 011265 002526 MOV (R2),TS4CL(R5)
4929 024062 042765 177400 002526 BIC #177400,TS4CL(R5)
4930 024070 IF PASCNT(R5) EQ #1 THEN ;IF THIS IS PASS 1 THEN:
4931 024070 026527 003260 000001 CMP PASCNT(R5),#1
4932 024076 001014 BNE 50447:
4933 024100 PRINTF #CODELM,DEVTBL(R5),TS4CL(R5) ;PRINT THE TK25 MICROCODE LEVEL.
4934 024100 016546 002526 MOV TS4CL(R5),-(SP)
4935 024104 016546 002536 MOV DEVTBL(R5),-(SP)
4936 024110 012746 004054 MOV #CODELM,-(SP)
4937 024114 012746 000003 MOV #3,(SP)
4938 024120 010600 MOV SP,R0
4939 024122 104417 TRAP C1PNTF
4940 024124 062706 000010 ADD #10,SP
4941 024130 ENDF
4942 024130 004737 017346 JSR PC,NEXTU ;FIND NEXT UNIT.
4943 024134 000740 ENDDO
4944 024136 ENDSUB
50447:
BR 50445:
50446:

```

Address	Instruction	Label	Comment
024136		L10020:	
024136	104403		TRAP C1ESUB
4929			
4930	024140	T1.2:	BGNSUB ;SUBTEST 2 - REWIND.
024140			
024140	104402		TRAP C1BSUB
4931			
4932	024142		LET R2 := #BFSEQ1 ;ADR OF CMD SEQ.
024142	012702 024714		MOV #BFSEQ1,R2
4933	024146 004737 024616		;SET UP CMD SEQ.
4934	024152 004737 006744		;EXECUTE CMD SEQ ON ALL DEVICES.
4935	024156		;CLEAR START FLAG,
024156	105037 003534		CLRB STAFLG
4936	024162		ENDSUB
024162		L10021:	
024162	104403		TRAP C1ESUB
4937			
4938	024164	T1.3:	BGNSUB ;SUBTEST 3 - WRITE/VERIFY.
024164			
024164	104402		TRAP C1BSUB
4939			
4940	024166		LET R2 := #BFSEQ2 ;ADR OF CMD SEQ.
024166	012702 024726		MOV #BFSEQ2,R2
4941	024172 004737 024616		;SET UP CMD SEQ.
4942	024176 004737 006744		;EXECUTE CMD SEQ ON ALL DEVICES.
4943	024202		ENDSUB
024202		L10022:	
024202	104403		TRAP C1ESUB
4944			
4945	024204	T1.4:	BGNSUB ;SUBTEST 4 WRITE TAPE MARK, ERASE.
024204			
024204	104402		TRAP C1BSUB
4946			
4947	024206		LET R2 := #BFSEQ3 ;ADR OF CMD SEQ.
024206	012702 025020		MOV #BFSEQ3,R2
4948	024212 004737 024616		;SET UP CMD SEQ.
4949	024216 004737 006744		;EXECUTE CMD SEQ ON ALL DEVICES.
4950	024222		ENDSUB
024222		L10023:	
024222	104403		TRAP C1ESUB
4951			
4952	024224	T1.5:	BGNSUB ;SUBTEST 5 SPACE FILES.
024224			
024224	104402		TRAP C1BSUB
4953			
4954	024226		LET R2 := #BFSEQ4 ;ADR OF CMD SEQ.
024226	012702 025072		MOV #BFSEQ4,R2
4955	024232 004737 024616		;SET UP CMD SEQ.
4956	024236 004737 006744		;EXECUTE CMD SEQ ON ALL DEVICES.
4957	024242		ENDSUB
024242		L10024:	
024242	104403		TRAP C1ESUB
4958			
4959	024244	T1.6:	BGNSUB ;SUBTEST 6 SPACE RECORDS.
024244			
024244	104402		TRAP C1BSUB
4960			

HARDWARE TESTS MACRO M1200 21 MAR 84 16:45 PAGE 104 2
TEST 1: BASIC FUNCTIONS.

SEQ 0147

```

4961 024246          LET R2 := #BFSEQ5
      024246 012702 025134
4962 024252          JSR    PC,BFSEQ
4963 024256 004737 006744 JSR    PC,EXALL
4964 024262          ENDSUB
      024262          L10025:
      024262 104403
4965
4966 024264          BGNSUB
      024264          T1.7:
      024264 104402
4967
4968 024266          LET R2 := #BFSEQ6
      024266 012702 025206
4969 024272          JSR    PC,BFSEQ
4970 024276 004737 006744 JSR    PC,EXALL
4971 024302          ENDSUB
      024302          L10026:
      024302 104403
4972
4973 024304          BGNSUB
      024304          T1.8:
      024304 104402
4974
4975 024306          LET R2 := #BFSEQ7
      024306 012702 025240
4976 024312          JSR    PC,BFSEQ
4977 024316 004737 006744 JSR    PC,EXALL
4978 024322          ENDSUB
      024322          L10027:
      024322 104403
4979
4980 024324          BGNSUB
      024324          T1.9:
      024324 104402
4981
4982 024326          LET R2 := #BFSEQ8
      024326 012702 025272
4983 024332          JSR    PC,BFSEQ
4984 024336 004737 006744 JSR    PC,EXALL
4985 024342          ENDSUB
      024342          L10030:
      024342 104403
4986
4987 024344          BGNSUB
      024344          T1.10:
      024344 104402
4988
4989 024346          LET R2 := #BFSEQ9
      024346 012702 025324
4990 024352          JSR    PC,BFSEQ
4991 024356 004737 006744 JSR    PC,EXALL
4992 024362          ENDSUB
      024362          L10031:
      024362 104403
4993
4994 024364          BGNSUB

```

```

;ADR OF CMD SEQ.
      MOV    #BFSEQ5,R2
;SET UP CMD SEQ.
;EXECUTE CMD SEQ ON ALL DEVICES.

      TRAP    C$ESUB

;SUBTEST 7  WRITE RETRY.

      TRAP    C$BSUB

;ADR OF CMD SEQ.
      MOV    #BFSEQ6,R2
;SET UP CMD SEQ.
;EXECUTE CMD SEQ ON ALL DEVICES.

      TRAP    C$ESUB

;SUBTEST 8  READ REV RETRY.

      TRAP    C$BSUB

;ADR OF CMD SEQ.
      MOV    #BFSEQ7,R2
;SET UP CMD SEQ.
;EXECUTE CMD SEQ ON ALL DEVICES.

      TRAP    C$ESUB

;SUBTEST 9 - READ FWD RETRY.

      TRAP    C$BSUB

;ADR OF CMD SEQ.
      MOV    #BFSEQ8,R2
;SET UP CMD SEQ.
;EXECUTE CMD SEQ ON ALL DEVICES.

      TRAP    C$ESUB

;SUBTEST 10- CLEAN.

      TRAP    C$BSUB

;ADR OF CMD SEQ.
      MOV    #BFSEQ9,R2
;SET UP CMD SEQ.
;EXECUTE CMD SEQ ON ALL DEVICES.

      TRAP    C$ESUB

;SUBTEST 11 - WTV SWAPPED DATA BYTES.

```

024364				T1.11:	
4995	024364	104402			TRAP C#BSUB
4996	024366			LET R2 := #BFSE10	;ADR OF CMD SEQ.
	024366	012702	025346		MOV #BFSE10,R2
4997	024372	004737	024616	JSR PC,BFSEQ	;SET UP CMD SEQ.
4998	024376	004737	006744	JSR PC,EXALL	;WRITE/VERIFY RECORDS 1 AND 2.
4999	024402			LET SWBFLG :B= #1	;ENABLE BYTE SWAPPING.
	024402	112737	000001		MOVB #1,SWBFLG
5000	024410	004737	006744	JSR PC,EXALL	;WRITE/VERIFY RECORDS 3 AND 4.
5001	024414			LET SWBFLG :B= #0	;DISABLE BYTE SWAPPING.
	024414	105037	003524		CLRB SWBFLG
5002	024420			ENDSUB	
	024420			L10032:	
	024420	104403			TRAP C#ESUB
5003					
5004	024422			LET R2 := DATAWT * #10.	;INIT WRITE BUFFER POINTER.
	024422	013702	003414		MOV DATAWT,R2
	024426	062702	000012		ADD #10.,R2
5005	024432			WHILE R2 NE DATAWT DO ;UNTIL 10 BYTES HAVE BEEN SWAPPED.	
	024432				504501:
	024432	020237	003414		CMP R2,DATAWT
	024436	001402			BEQ 504511
5006	024440	000342		SWAB -(R2)	;SWAP DATA BYTES IN WRITE BUFFER.
5007	024442			ENDDO	
	024442	000773			BR 504501
	024444				504511:
5008	024444			LET T1SWB :B= T1SWB * #1	;SET T1 SWAP BYTES FLAG FOR 'CKDATA' SUBR
	024444	105237	003530		INCB T1SWB
5009					
5010	024450			BGNSUB	;SUBTEST 12 READ SWAPPED DATA BYTES.
	024450			T1.12:	
	024450	104402			TRAP C#BSUB
5011					
5012	024452			LET CMDWRD := #RDR	;CMD IS READ REV.
	024452	012737	104401		MOV #RDR,CMDWRD
5013	024460	004737	016254	JSR PC,VFEXC	;VERIFY ODD LENGTH SWAP (RECORD 4).
5014	024464			LET CMDPKT.CP.CNT := #12	;CHANGE BYTE COUNT TO 10.
	024464	012737	000012		MOV #12,CMDPKT.CP.CNT
5015	024472	004737	016254	JSR PC,VFEXC	;VERIFY EVEN LENGTH SWAP (RECORD 3).
5016	024476			LET SWBFLG :B= #1	;ENABLE BYTE SWAPPING.
	024476	112737	000001		MOVB #1,SWBFLG
5017	024504			LET CMDPKT.CP.CNT := #11	;CHANGE BYTE COUNT TO 9.
	024504	012737	000011		MOV #11,CMDPKT.CP.CNT
5018	024512	004737	016254	JSR PC,VFEXC	;VERIFY ODD LENGTH SWAP (RECORD 2).
5019	024516			LET CMDPKT.CP.CNT := #12	;CHANGE BYTE COUNT TO 10.
	024516	012737	000012		MOV #12,CMDPKT.CP.CNT
5020	024524	004737	016254	JSR PC,VFEXC	;VERIFY EVEN LENGTH SWAP (RECORD 1).
5021	024530			LET CMDWRD := #RDF	;CMD IS READ FWD.
	024530	012737	104001		MOV #RDF,CMDWRD
5022	024536	004737	016254	JSR PC,VFEXC	;VERIFY EVEN LENGTH SWAP (RECORD 1).
5023	024542			LET CMDPKT.CP.CNT := #11	;CHANGE BYTE COUNT TO 9.
	024542	012737	000011		MOV #11,CMDPKT.CP.CNT
5024	024550	004737	016254	JSR PC,VFEXC	;VERIFY ODD LENGTH SWAP (RECORD 2).
5025	024554			LET SWBFLG :B= #0	;DISABLE BYTE SWAPPING.
	024554	105037	003524		CLRB SWBFLG
5026	024560			LET CMDPKT.CP.CNT := #12	;CHANGE BYTE COUNT TO 10.

```

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

```

```

MOV      #12,CMDPKT.CP.CNT
;VERIFY EVEN LENGTH SWAP (RECORD 3).
;CHANGE BYTE COUNT TO 9.
MOV      #11,CMDPKT.CP.CNT
;VERIFY ODD LENGTH SWAP (RECORD 4).

TRAP     C$ESUB

;CLEAR T1 SWAP BYTES FLAG
CLR      T1SWB

TRAP     C$EXIT
.WORD    L10017

```

```

5038      ;      SUBROUTINE TO MOVE A COMMAND SEQUENCE TO THE SEQUENCE TABLE.
5039      ;      INPUTS:      R2 = FWA OF COMMAND SEQUENCE.
5040      ;      OUTPUTS:
5041      ;      REGISTERS:
5042      ;      CALLS:
5043
5044 024616      BFSEQ: LET R1 := #CMDSEQ      ;INIT SEQ TABLE ADDRESS.
5045 024616 012701 003542      MOV      #CMDSEQ,R1
5046 024622      WHILE (R2) NE #END DO      ;WHILE THERE ARE MORE COMMANDS:
5047 024622      50452$: CMP      (R2),#END
5048 024626 021227 177777      BEQ      50453$
5049 024630      LET (R1)+ := (R2)+      ;MOVE COMMANDS TO SEQ TABLE.
5050 024630 012221      MOV      (R2)+,(R1)+
5051 024632      ENDDO
5052 024632 000773      BR      50452$
5053 024634      LET (R1) := #END      ;STORE END OF SEQUENCE CODE.
5054 024634 012711 177777      MOV      #END,(R1)
5055 024640 000207      RTS      PC      ;RETURN.
5056
5057      ;      BASIC FUNCTION COMMAND SEQUENCE
5058
5059 024642 140004      BFSEQ0: .WORD      SCH      ;SET CHAR. 200.      (1)
5060 024644 000200      200
5061 024646 000001      1
5062 024650 000000      0
5063 024652 100013      DRI      ;DRIVE INIT.      (2)
5064 024654 000001      1
5065 024656 000001      1
5066 024660 000000      0
5067 024662 140004      SCH      ;SET CHAR. 20      (3)
5068 024664 000020      20
5069 024666 000001      1
5070 024670 000000      0
5071 024672 100017      GES      ;GET STATUS.      (4)
5072 024674 000001      1
5073 024676 000001      1
5074 024700 000000      0
5075 024702 140004      SCH      ;SET CHAR. 40.      (5)
5076 024704 000040      40
5077 024706 000001      1
5078 024710 000000      0
5079 024712 177777      .WORD      END
5080
5081 024714 102010      BFSEQ1:      RWD      ;REWIND TWICE.      (6)
5082 024716 000001      1
5083 024720 000002      2
5084 024722 000000      0
5085 024724 177777      .WORD      END
5086
5087 024726 104105      BFSEQ2:      WTV      ;WRITE/VERIFY PAT 1.      (7)
5088 024730 010000      DATCNT
5089 024732 000001      1
5090 024734 000001      1

```

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

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

```

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

```

```

5240      .SBTTL  TEST 2: DATA RELIABILITY.
5241
5242      ;**
5243      ; TEST TO CHECK THE DATA RELIABILITY OF THE TK25.
5244      ;*
5245      BGNTST
5246      T2::
5247      025372      LET RANDOM :B= #1      ;SET THE RANDOM OPERATIONS FLAG.
5248      025372      112737 000001 003521      MOV      #1,RANDOM
5249      025400      LET EXPBOT :B= #0      ;CLEAR EXPECT BOT FLAG.
5250      025400      105037 003520      CLR      EXPBOT
5251      025404      LET R2 := #DATCNT #1      ;SET UP THE RECORD LENGTH MASK,
5252      025404      012702 010000      MOV      #DATCNT,R2
5253      025410      005302      DEC      R2
5254      025412      LET LENMSK := COMP R2      ;ALLOW MAXIMUM BUFFER.
5255      025412      010237 003436      MOV      R2,LENMSK
5256      025416      005137 003436      COM      LENMSK
5257      025422      004737 006700      JSR PC,SETCH      ;CMD 1 = SET CHARACTERISTIC.
5258      025426      IFB STAFLG NE #0 THEN      ;IF STARTING THEN:
5259      025426      105737 003534      TSTB      STAFLG
5260      025432      001404      BEQ      50454$
5261      025434      004737 006724      JSR PC,SETRW      ;CMD2=REWIND
5262      025440      LET STAFLG :B= #0      ;CLR START FLAG.
5263      025440      105037 003534      CLR      STAFLG
5264      025444      ENDIF
5265      025444      50454$:
5266      025444      LET (R1)+ := #WRT      ;CMD3 = WRITE.
5267      025444      012721 104005      MOV      #WRT,(R1)+
5268      025450      LET (R1)+ := #DATCNT      ;SET BRG TO MAX FOR PATTERN GENERATION.
5269      025450      012721 010000      MOV      #DATCNT,(R1)+
5270      025454      LET R2 := COMP #RNOPSC
5271      025454      012702 177740      MOV      #RNOPSC,R2
5272      025460      005102      COM      R2
5273      025462      LET (R1)+ := R2      ;31 OPERATIONS.
5274      025462      010221      MOV      R2,(R1)+
5275      025464      LET (R1)+ := #RANP      ;RANDOM PATTERN.
5276      025464      012721 000007      MOV      #RANP,(R1)+
5277      025470      REPEAT      ;REPEAT TO EOT:
5278      025470      50455$:
5279      025470      WHILE R1 LT #SEQEND DO      ;FILL SEQ TBL WITH RANDOM CMDS.
5280      025470      020127 003632      50456$:
5281      025474      002012      CMP      R1,#SEQEND
5282      025476      LET RANS := RANS + RANB      BGE      50457$
5283      025476      063737 003440 003442      ADD      RANB,RANS
5284      025504      LET R2 := RANS CLR.BY #177741 ;R2 = RANDOM # (0 36).
5285      025504      013702 003442      MOV      RANS,R2
5286      025510      042702 177741      BIC      #177741,R2
5287      025514      004772 025652      JSR PC,#RANCMD(R2)      ;SET UP A RANDOM CMD + BRG.
5288      025520      ENDDO
5289      025520      000763      BR      50456$
5290      025522      50457$:
5291      025522      LET (R1) := #END      ;STORE END OF SEQUENCE CODE IN TABLE.
5292      025522      012711 177777      MOV      #END,(R1)
5293      025526      004737 006744      JSR PC,EXALL      ;GO EXECUTE ALL CMDS IN SEQUENCE TABLE.
5294      025532      LET R1 := #CMDSEQ      ;INIT CMD SEQ TBL POINTER,

```

SEQ 0155

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

```

5294      ;      ADDRESSES OF SUBROUTINES USED TO SET UP RANDOM OPERATIONS IN
5295      ;      THE DATA RELIABILITY TEST.
5296
5297 025652 025766      RANCMD: RANWR      ;WRITE.
5298 025654 025766      RANWR      ;WRITE.
5299 025656 025766      RANWR      ;WRITE.
5300 025660 025766      RANWR      ;WRITE.
5301 025662 025766      RANWR      ;WRITE.
5302 025664 025766      RANWR      ;WRITE.
5303 025666 025766      RANWR      ;WRITE.
5304 025670 025766      RANWR      ;WRITE.
5305 025672 025712      RANRD      ;READ.
5306 025674 025712      RANRD      ;READ.
5307 025676 025712      RANRD      ;READ.
5308 025700 025712      RANRD      ;READ.
5309 025702 025712      RANRD      ;READ.
5310 025704 025712      RANRD      ;READ.
5311 025706 025712      RANRD      ;READ.
5312 025710 025712      RANRD      ;READ.
5313
5314
5315
5316
5317
5318      ;      SUBROUTINE TO SET UP READ COMMANDS IN SEQUENCE TABLE.
5319      ;      INPUTS:
5320      ;      OUTPUTS:
5321      ;      REGISTERS:      R2
5322      ;      CALLS:
5323
5324 025712      RANRD: LET (R1)+ := #SRR      ;STORE SPACE RECORD REVERSE CMD
5325 025712 012721 104410      MOV      #SRR,(R1)+
5326 025716      LET RANB := RANB + RANS
5327 025724 063737 003442 003440      ADD      RANS,RANB
5328 025724 013702 003440      LET R2 := RANB CLR.BY #RNOPSC
5329 025730 042702 177740      MOV      RANB,R2
5330 025734      LET (R1)+ :- R2      ;SET REPOSITION COUNT
5331 025736 010221      BIC      #RNOPSC,R2
5332 025736 012721 000001      MOV      R2,(R1)+
5333 025742      LET (R1)+ := #1      ;DO ONCE
5334 025742 012721 000007      MOV      #1,(R1)+
5335 025746      LET (R1)+ := #RANP      ;RANDOM PATTERN.
5336 025746 012721 104001      MOV      #RANP,(R1)+
5337 025752      LET (R1)+ := #RDF      ;STORE READ FWD CMD.
5338 025752 012721 010000      MOV      #RDF,(R1)+
5339 025756      LET (R1)+ := #DATCNT      ;SET BRF TO MAX TO READ RANDOM LENGTHS.
5340 025756 010221      MOV      #DATCNT,(R1)+
5341 025760      LET (R1)+ := R2      ;SET RANDOM # OF OPERATIONS.
5342 025760 012721 000007      MOV      R2,(R1)+
5343 025764      LET (R1)+ := #RANP      ;RANDOM PATTERN.
5344      MOV      #RANP,(R1)+
5345      RTS PC

```

```

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

```

```

5384 .SBTTL TEST 3: WRITE AND READ STREAMING TEST.
5385
5386
5387
5388
5389
5390
5391
5392
5393
5394 026046      BGNTST
      026046      T3::
5395
5396 026046      IFB HERE EQ #0 THEN
      026046 105737 003340
      026052 001012
5397 026054      PRINTF #TAPCAP
      026054 012746 021571
      026060 012746 000001
      026064 010600
      026066 104417
      026070 062706 000004
5398 026074      LET HERE :B= HERE + #1
      026074 105237 003340
5399 026100      ENDIF
      026100
5400 026100      LET RANDOM :B= #0
      026100 105037 003521
5401 026104      LET EXPBOT :B= #0
      026104 105037 003520
5402 026110      JSR PC, SETCH
      026110 004737 006700
5403 026114      JSR PC, SETRW
      026114 004737 006724
5404 026120      LET STAF LG :B= #0
      026120 105037 003534
5405 026124      LET (R1) := #END
      026124 012711 177777
5406 026130      JSR PC, EXALL
      026130 004737 006744
5407
5408 026134      JSR PC, FIRSTU
      026134 004737 017300
5409
5410
5411
5412
5413
5414
5415
5416 026140      WHILE DEVTBL(R5) NE #END DO
      026140
      026140 026527 002536 177777
      026146 001544
5417
5418 026150      LET BTPT := BTADDR(R5)
      026150 016537 002550 003516
5419 026156      LET #BTPT := #0
      026156 005077 155334
5420 026162      LET STREAM :B= #255.
      026162 112737 000377 003532

```

```

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

TSTB HERE
BNE 504601

MOV #TAPCAP, -(SP)
MOV #1, -(SP)
MOV SP, R0
TRAP C$PRINTF
ADD #4, SP

INCB HERE

504601:
;CLEAR THE RANDOM OPERATIONS FLAG.
CLRB RANDOM
;CLEAR THE EXPECT BOT FLAG.
CLRB EXPBOT
;SET CHARACTERISTICS.
;SET REWIND COMMAND IN BUFFER.
;CLEAR THE START FLAG.
CLRB STAF LG
;PLACE END FLAG IN SEQUENCE TABLE.
MOV #END, (R1)
;REWIND ALL UNITS.

;FIND THE FIRST UNIT TO TEST (UUT)

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

;WHILE THERE ARE MORE DEVICES:
504611:
CMP DEVTBL(R5), #END
BEQ 504621

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

```

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

```

5451 026360      012721 104001
5452 026364      012721 010000
5453 026370      012721 021450
5454 026374      012721 000005
5455 026400      012721 177777
5456 026404      012701 003542
5457 026410      004737 007706
5458 026414      112737 000001 003522
5459 026422      010537 003460
5460 026426      004737 016142
5461 026432      013705 003460
5462 026436
5463 026436      005037 003420
5464 026442      105037 003522
5465 026446      105037 003520
5466 026452      004737 017346
5467
5468 026456      000630
5469
5470 026460      105037 003532
5471 026464      105037 003531
5472 026470      112737 000001 003523
5473 026476      012701 003542
5474 026502      004737 006724
5475 026506      012711 177777
5476 026512      004737 006744
5477
5478 026516      104432
5479 026516      000002
5480
5481 026522
5482 026522      104401

```

```

LET (R1) := #RDF
LET (R1) := #DATCNT
LET (R1) := #9000.
LET (R1) := #5
LET (R1) := #END
LET R1 := #CMDSEQ
JSR PC, SETUP
LET VFYFLG :B= #1
LET RSSAVE := R5
JSR PC, VFYDAT
LET R5 := RSSAVE
ENDIF
LET NCNT := #0
LET VFYFLG :B= #0
LET EXPBOT :B= #0
JSR PC, NEXTU
ENDDO
LET STREAM :B= #0
LET ALLEOT :B= #0
LET RPTFLG :B= #1
LET R1 := #CMDSEQ
JSR PC, SETRW
LET (R1) := #END
JSR PC, EXALL
EXIT TST
.EVEN
ENDTST

```

L10035:

```

;READ FORWARD COMMAND
MOV #RDF,(R1)
;4096 BYTE RECORDS
MOV #DATCNT,(R1)
;9000 ITERATIONS
MOV #9000.,(R1)
;READ PATTERN NUMBER 5
MOV #5,(R1)
;TABLE TERMINATOR
MOV #END,(R1)
;TOP OF TABLE, AGAIN!
MOV #CMDSEQ,R1
;SET UP THE COMMAND TABLE
;ALLOW THE DATA VERIFY
MOVB #1,VFYFLG
;SAVE R5
MOV R5,RSSAVE
;GO OFF AND CHECK READ OPERATIONS
;RESTORE R5
MOV RSSAVE,R5
50466:
;CLEAR RECORD COUNT
CLR NCNT
;CLEAR VERIFY FLAG
CLRB VFYFLG
;CLEAR EXPECT BOT FLAG.
CLRB EXPBOT
;GET NEXT UNIT TO TEST (UUT).
;END OF UUT LOOP
BR 50461
50462:
;CLEAR STREAMING FLAG FOR OTHER TESTS.
CLPB STREAM
;RESET THE UNIT @ EOT STATUS
CLRB ALLEOT
;REQUEST A REPORT
MOVB #1,RPTFLG
;TOP OF TABLE
MOV #CMDSEQ,R1
;STORE THE REWIND COMMAND
;TERMINATOR
MOV #END,(R1)
;REWIND AND REPORT STATUS FOR ALL UNITS
;EXIT TEST
TRAP C$EXIT
;JUST IN CASE.
.WORD L10035
TRAP C$ETST

```

```

5483
5484
5485
5486
5487
5488
5489
5490 026524
5491 026524
5492 026524 112737 000001 003521
5493 026532 105037 003520
5494 026536 012702 010000
5495 026542 005302
5496 026544 010237 003436
5497 026550 005137 003436
5498 026554 004737 006700
5499 026560 004737 006724
5500 026564 105037 003534
5501 026570
5502 026570
5503 026570 020127 003632
5504 026574 002003
5505 026576 004737 025766
5506 026602
5507 026602 000772
5508 026604
5509 026604 012711 177777
5510 026610 004737 006744
5511 026614 012701 003542
5512 026620 005702
5513 026622 001762
5514 026624 105237 003531
5515 026630 000240
5516 026632 000240
5517 026634 000240
5518 026636 004737 027640
5519
5520
5521
5522
5523
5524
5525
5526
5527
5528
5529
5530
5531
5532
5533
5534
5535
5536
5537
5538
5539
5540
5541
5542
5543
5544
5545
5546
5547
5548
5549
5550
5551
5552
5553
5554
5555
5556
5557
5558
5559
5560
5561
5562
5563
5564
5565
5566
5567
5568
5569
5570
5571
5572
5573
5574
5575
5576
5577
5578
5579
5580
5581
5582
5583
5584
5585
5586
5587
5588
5589
5590
5591
5592
5593
5594
5595
5596
5597
5598
5599
5600
5601
5602
5603
5604
5605
5606
5607
5608
5609
5610
5611
5612
5613
5614
5615
5616
5617
5618
5619
5620
5621
5622
5623
5624
5625
5626
5627
5628
5629
5630
5631
5632
5633
5634
5635
5636
5637
5638
5639
5640
5641
5642
5643
5644
5645
5646
5647
5648
5649
5650
5651
5652
5653
5654
5655
5656
5657
5658
5659
5660
5661
5662
5663
5664
5665
5666
5667
5668
5669
5670
5671
5672
5673
5674
5675
5676
5677
5678
5679
5680
5681
5682
5683
5684
5685
5686
5687
5688
5689
5690
5691
5692
5693
5694
5695
5696
5697
5698
5699
5700
5701
5702
5703
5704
5705
5706
5707
5708
5709
5710
5711
5712
5713
5714
5715
5716
5717
5718
5719
5720
5721
5722
5723
5724
5725
5726
5727
5728
5729
5730
5731
5732
5733
5734
5735
5736
5737
5738
5739
5740
5741
5742
5743
5744
5745
5746
5747
5748
5749
5750
5751
5752
5753
5754
5755
5756
5757
5758
5759
5760
5761
5762
5763
5764
5765
5766
5767
5768
5769
5770
5771
5772
5773
5774
5775
5776
5777
5778
5779
5780
5781
5782
5783
5784
5785
5786
5787
5788
5789
5790
5791
5792
5793
5794
5795
5796
5797
5798
5799
5800
5801
5802
5803
5804
5805
5806
5807
5808
5809
5810
5811
5812
5813
5814
5815
5816
5817
5818
5819
5820
5821
5822
5823
5824
5825
5826
5827
5828
5829
5830
5831
5832
5833
5834
5835
5836
5837
5838
5839
5840
5841
5842
5843
5844
5845
5846
5847
5848
5849
5850
5851
5852
5853
5854
5855
5856
5857
5858
5859
5860
5861
5862
5863
5864
5865
5866
5867
5868
5869
5870
5871
5872
5873
5874
5875
5876
5877
5878
5879
5880
5881
5882
5883
5884
5885
5886
5887
5888
5889
5890
5891
5892
5893
5894
5895
5896
5897
5898
5899
5900
5901
5902
5903
5904
5905
5906
5907
5908
5909
5910
5911
5912
5913
5914
5915
5916
5917
5918
5919
5920
5921
5922
5923
5924
5925
5926
5927
5928
5929
5930
5931
5932
5933
5934
5935
5936
5937
5938
5939
5940
5941
5942
5943
5944
5945
5946
5947
5948
5949
5950
5951
5952
5953
5954
5955
5956
5957
5958
5959
5960
5961
5962
5963
5964
5965
5966
5967
5968
5969
5970
5971
5972
5973
5974
5975
5976
5977
5978
5979
5980
5981
5982
5983
5984
5985
5986
5987
5988
5989
5990
5991
5992
5993
5994
5995
5996
5997
5998
5999
6000
6001
6002
6003
6004
6005
6006
6007
6008
6009
6010
6011
6012
6013
6014
6015
6016
6017
6018
6019
6020
6021
6022
6023
6024
6025
6026
6027
6028
6029
6030
6031
6032
6033
6034
6035
6036
6037
6038
6039
6040
6041
6042
6043
6044
6045
6046
6047
6048
6049
6050
6051
6052
6053
6054
6055
6056
6057
6058
6059
6060
6061
6062
6063
6064
6065
6066
6067
6068
6069
6070
6071
6072
6073
6074
6075
6076
6077
6078
6079
6080
6081
6082
6083
6084
6085
6086
6087
6088
6089
6090
6091
6092
6093
6094
6095
6096
6097
6098
6099
6100
6101
6102
6103
6104
6105
6106
6107
6108
6109
6110
6111
6112
6113
6114
6115
6116
6117
6118
6119
6120
6121
6122
6123
6124
6125
6126
6127
6128
6129
6130
6131
6132
6133
6134
6135
6136
6137
6138
6139
6140
6141
6142
6143
6144
6145
6146
6147
6148
6149
6150
6151
6152
6153
6154
6155
6156
6157
6158
6159
6160
6161
6162
6163
6164
6165
6166
6167
6168
6169
6170
6171
6172
6173
6174
6175
6176
6177
6178
6179
6180
6181
6182
6183
6184
6185
6186
6187
6188
6189
6190
6191
6192
6193
6194
6195
6196
6197
6198
6199
6200
6201
6202
6203
6204
6205
6206
6207
6208
6209
6210
6211
6212
6213
6214
6215
6216
6217
6218
6219
6220
6221
6222
6223
6224
6225
6226
6227
6228
6229
6230
6231
6232
6233
6234
6235
6236
6237
6238
6239
6240
6241
6242
6243
6244
6245
6246
6247
6248
6249
6250
6251
6252
6253
6254
6255
6256
6257
6258
6259
6260
6261
6262
6263
6264
6265
6266
6267
6268
6269
6270
6271
6272
6273
6274
6275
6276
6277
6278
6279
6280
6281
6282
6283
6284
6285
6286
6287
6288
6289
6290
6291
6292
6293
6294
6295
6296
6297
6298
6299
6300
6301
6302
6303
6304
6305
6306
6307
6308
6309
6310
6311
6312
6313
6314
6315
6316
6317
6318
6319
6320
6321
6322
6323
6324
6325
6326
6327
6328
6329
6330
6331
6332
6333
6334
6335
6336
6337
6338
6339
6340
6341
6342
6343
6344
6345
6346
6347
6348
6349
6350
6351
6352
6353
6354
6355
6356
6357
6358
6359
6360
6361
6362
6363
6364
6365
6366
6367
6368
6369
6370
6371
6372
6373
6374
6375
6376
6377
6378
6379
6380
6381
6382
6383
6384
6385
6386
6387
6388
6389
6390
6391
6392
6393
6394
6395
6396
6397
6398
6399
6400
6401
6402
6403
6404
6405
6406
6407
6408
6409
6410
6411
6412
6413
6414
6415
6416
6417
6418
6419
6420
6421
6422
6423
6424
6425
6426
6427
6428
6429
6430
6431
6432
6433
6434
6435
6436
6437
6438
6439
6440
6441
6442
6443
6444
6445
6446
6447
6448
6449
6450
6451
6452
6453
6454
6455
6456
6457
6458
6459
6460
6461
6462
6463
6464
6465
6466
6467
6468
6469
6470
6471
6472
6473
6474
6475
6476
6477
6478
6479
6480
6481
6482
6483
6484
6485
6486
6487
6488
6489
6490
6491
6492
6493
6494
6495
6496
6497
6498
6499
6500
6501
6502
6503
6504
6505
6506
6507
6508
6509
6510
6511
6512
6513
6514
6515
6516
6517
6518
6519
6520
6521
6522
6523
6524
6525
6526
6527
6528
6529
6530
6531
6532
6533
6534
6535
6536
6537
6538
6539
6540
6541
6542
6543
6544
6545
6546
6547
6548
6549
6550
6551
6552
6553
6554
6555
6556
6557
6558
6559
6560
6561
6562
6563
6564
6565
6566
6567
6568
6569
6570
6571
6572
6573
6574
6575
6576
6577
6578
6579
6580
6581
6582
6583
6584
6585
6586
6587
6588
6589
6590
6591
6592
6593
6594
6595
6596
6597
6598
6599
6600
6601
6602
6603
6604
6605
6606
6607
6608
6609
6610
6611
6612
6613
6614
6615
6616
6617
6618
6619
6620
6621
6622
6623
6624
6625
6626
6627
6628
6629
6630
6631
6632
6633
6634
6635
6636
6637
6638
6639
6640
6641
6642
6643
6644
6645
6646
6647
6648
6649
6650
6651
6652
6653
6654
6655
6656
6657
6658
6659
6660
6661
6662
6663
6664
6665
6666
6667
6668
6669
6670
6671
6672
6673
6674
6675
6676
6677
6678
6679
6680
6681
6682
6683
6684
6685
6686
6687
6688
6689
6690
6691
6692
6693
6694
6695
6696
6697
6698
6699
6700
6701
6702
6703
6704
6705
6706
6707
6708
6709
6710
6711
6712
6713
6714
6715
6716
6717
6718
6719
6720
6721
6722
6723
6724
6725
6726
6727
6728
6729
6730
6731
6732
6733
6734
6735
6736
6737
6738
6739
6740
6741
6742
6743
6744
6745
6746
6747
6748
6749
6750
6751
6752
6753
6754
6755
6756
6757
6758
6759
6760
6761
6762
6763
6764
6765
6766
6767
6768
6769
6770
6771
6772
6773
6774
6775
6776
6777
6778
6779
6780
6781
6782
6783
6784
6785
6786
6787
6788
6789
6790
6791
6792
6793
6794
6795
6796
6797
6798
6799
6800
6801
6802
6803
6804
6805
6806
6807
6808
6809
6810
6811
6812
6813
6814
6815
6816
6817
6818
6819
6820
6821
6822
6823
6824
6825
6826
6827
6828
6829
6830
6831
6832
6833
6834
6835
6836
6837
6838
6839
6840
6841
6842
6843
6844
6845
6846
6847
6848
6849
6850
6851
6852
6853
6854
6855
6856
6857
6858
6859
6860
6861
6862
6863
6864
6865
6866
6867
6868
6869
6870
6871
6872
6873
6874
6875
6876
6877
6878
6879
6880
6881
6882
6883
6884
6885
6886
6887
6888
6889
6890
6891
6892
6893
6894
6895
6896
6897
6898
6899
6900
6901
6902
6903
6904
6905
6906
6907
6908
6909
6910
6911
6912
6913
6914
6915
6916
6917
6918
6919
6920
6921
6922
6923
6924
6925
6926
6927
6928
6929
6930
6931
6932
6933
6934
6935
6936
6937
6938
6939
6940
6941
6942
6943
6944
6945
6946
6947
6948
6949
6950
6951
6952
6953
6954
6955
6956
6957
6958
6959
6960
6961
6962
6963
6964
6965
6966
6967
6968
6969
6970
6971
6972
6973
6974
6975
6976
6977
6978
6979
6980
6981
6982
6983
6984
6985
6986
6987
6988
6989
6990
6991
6992
6993
6994
6995
6996
6997
6998
6999
7000
7001
7002
7003
7004
7005
7006
7007
7008
7009
7010
7011
7012
7013
7014
7015
7016
7017
7018
7019
7020
7021
7022
7023
7024
7025
7026
7027
7028
7029
7030
7031
7032
7033
7034
7035
7036
7037
7038
7039
7040
7041
7042
7043
7044
7045
7046
7047
7048
7049
7050
7051
7052
7053
7054
7055
7056
7057
7058
7059
7060
7061
7062
7063
7064
7065
7066
7067
7068
7069
7070
7071
7072
7073
7074
7075
7076
7077
7078
7079
7080
7081
7082
7083
7084
7085
7086
7087
7088
7089
7090
7091
7092
7093
7094
7095
7096
7097
7098
7099
7100
7101
7102
7103
7104
7105
7106
7107
7108
7109
7110
7111
7112
7113
7114
7115
7116
7117
7118
7119
7120
7121
7122
7123
7124
7125
7126
7127
7128
7129
7130
7131
7132
7133
7134
7135
7136
7137
7138
7139
7140
7141
7142
7143
7144
7145
7146
7147
7148
7149
7150
7151
7152
7153
7154
7155
7156
7157
7158
7159
7160
7161
7162
7163
7164
7165
7166
7167
7168
7169
7170
7171
7172
7173
7174
7175
7176
7177
7178
7179
7180
7181
7182
7183
7184
7185
7186
7187
7188
7189
7190
7191
7192
7193
7194
7195
7196
7197
7198
7199
7200
7201
7202
7203
7204
7205
7206
7207
7208
7209
7210
7211
7212
7213
7214
7215
7216
7217
7218
7219
7220
7221
7222
7223
7224
7225
7226
7227
7228
7229
7230
7231
7232
7233
7234
7235
7236
7237
7238
7239
7240
7241
7242
7243
7244
7245
7246
7247
7248
7249
7250
7251
7252
7253
7254
7255
7256
7257
7258
7259
7260
7261
7262
7263
7264
7265
7266
7267
7268
7269
7270
7271
7272
7273
7274
7275
7276
7277
7278
7279
7280
7281
7282
7283
7284
7285
7286
7287
7288
7289
7290
7291
7292
7293
7294
7295
7296
7297
7298
7299
7300
7301
7302
7303
7304
7305
7306
7307
7308
7309
7310
7311
7312
7313
7314
7315
7316
7317
7318
7319
7320
7321
7322
7323
7324
7325
7326
7327
7328
7329
7330
7331
7332
7333
7334
7335
7336
7337
7338
7339
7340
7341
7342
7343
7344
7345
7346
7347
7348
7349
7350
7351
7352
7353
7354
7355
7356
7357
7358
7359
7360
7361
7362
7363
7364
7365
7366
7367
7368
7369
7370
7371
7372
7373
7374
7375
7376
7377
7378
7379
7380
7381
7382
7383
7384
7385
7386
7387
7388
7389
7390
7391
7392
7393
7394
7395
7396
7397
7398
7399
7400
7401
7402
7403
7404
7405
7406
7407
7408
7409
7410
7411
7412
7413
7414
7415
7416
7417
7418
7419
7420
7421
7422
7423
7424
7425
7426
7427
7428
7429
7430
7431
7432
7433
7434
7435
7436
7437
7438
7439
7440
7441
7442
7443
7444
7445
7446
7447
7448
7449
7450
7451
7452
7453
7454
7455
7456
7457
7458
7459
7460
7461
7462
7463
7464
7465
7466
7467
7468
7469
7470
7471
7472
7473
7474
7475
7476
7477
7478
7479
7480
7481
7482
7483
7484
7485
7486
7487
7488
7489
7490
7491
7492
7493
7494
7495
7496
7497
7498
7499
7500
7501
7502
7503
7504
7505
7506
7507
7508
7509
7510
7511
7512
7513
7514
7515
7516
7517
7518
7519
7520
7521
7522
7523
7524
7525
7526
7527
7528
7529
7530
7531
7532
7533
7534
7535
7536
7537
7538
7539
7540
7541
7542
7543
7544
7545
7546
7547
7548
7549
7550
7551
7552
7553
7554
7555
7556
7557
7558
7559
7560
7561
7562
7563
7564
7565
7566
7567
7568
7569
7570
7571
7572
7573
7574
7575
7576
7577
7578
7579
7580
7581
7582
7583
7584
7585
7586
7587
7588
7589
7590
7591
7592
7593
7594
7595
7596
7597
7598
7599
7600
7601
7602
7603
7604
7605
7606
7607
7608
7609
7610
7611
7612
7613
7614
7615
7616
7617
7618
7619
7620
7621
7622
7623
7624
7625
7626
7627
7628
7629
7630
7631
7632
7633
7634
7635
7636
7637
7638
7639
7640
7641
7642
7643
7644
7645
7646
7647
7648
7649
7650
7651
7652
7653
7654
7655
7656
7657
7658
7659
7660
7661
7662
7663
7664
7665
7666
7667
7668
7669
7670
7671
7672
7673
7674
7675
7676
7677
7678
7679
7680
7681
7682
7683
7684
7685
7686
7687
7688
7689
7690
7691
7692
7693
7694
7695
7696
7697
7698
7699
7700
7701
7702
7703
7704
7705
7706
7707
7708
7709
7710
7711
7712
7713
7714
7715
7716
7717
7718
7719
7720
7721
7722
7723
7724
7725
7726
7727
7728
7729
7730
7731
7732
7733
7734
7735
7736
7737
7738
7739
7740
7741
7742
7743
7744
7745
7746
7747
7748
7749
7750
7751
7752
7753
7754
7755
7756
7757
7758
7759
7760
7761
7762
7763
7764
7765
7766
7767
7768
7769
7770
7771
7772
7773
7774
7775
7776
7777
7778
7779
7780
7781
7782
7783
7784
7785
7786
7787
7788
7789
7790
7791
7792
7793
7794
7795
7796
7797
7798
7799
7800
7801
7802
7803
7804
7805
7806
7807
7808
7809
7810
7811
7812
7813
7814
7815
7816
7817
7818
7819
7820
7821
7822
7823
7824
7825
7826
7827
7828
7829
7830
```

G13

HARDWARE TESTS MACRO M1200 21-MAR-84 16:45 PAGE 110 1
TEST 4: WRITE COMPATABILITY/WRITE UTILITY.

SEQ 0162

5519
5520 026662 EXIT TST
026662 104432
026664 000002
5521
5522 .EVEN
5523
5524 026666 ENDTST
026666 L10036:
026666 104401
5525

TRAP C\$EXIT
.WORD L10036-.

TRAP C\$ETST

```

5527
5528 .SBTTL TEST 5: READ COMPATABILITY/READ UTILITY.
5529
5530 ;**
5531 ; TEST TO READ ENTIRE TAPE FORWARD AND REVERSE.
5532 ;--
5533
5534 026670          BGNTST
5535 026670          T5::
5536 026670          112737 000001 003521      LET RANDOM :B= #1          ;SET THE RANDOM OPERATIONS FLAG.
5537 026676          112737 000001 003520      LET EXPBOT :B= #1          ;SET EXPECT BOT FLAG.
5538 026704          004737 006700              JSR PC,SETCH          ;CMD 1 = SET CHARACTERISTIC.
5539 026710          004737 006724              JSR PC,SETRW          ;CMD2=REWIND.
5540 026714          105037 003534              LET STAF LG :B= #0          ;CLEAR START FLAG
5541 026720          012721 104001              LET (R1)+ := #RDF          ;CMD3 = READ FORWARD.
5542 026724          012721 010000              LET (R1)+ := #DATCNT          ;SET LENGTH TO MAX FOR UNKNOWN LENGTHS.
5543 026730          012721 077777              LET (R1)+ := #77777          ;SET RECORD COUNT TO MAX FOR WHOLE TAPE.
5544 026734          012721 000007              LET (R1)+ := #RANP          ;PATTERN = RANDOM.
5545 026740          012711 177777              LET (R1) := #END          ;STORE END OF SEQUENCE CODE IN TABLE.
5546 026744          004737 006744              JSR PC,EXALL          ;EXECUTE ALL CMDS IN SEQ TBL ON ALL UNITS.
5547          ;          LET ALLEOT :B= ALLEOT + #1          ;FLAG TO ALLOW ALL UNITS AT EOT TO READ REV
5548          ;          LET R1 := #CMOSEQ          ;INIT CMD SEQ TBL POINTER.
5549          ;          LET (R1)+ := #RDR          ;CMD1 = READ REVERSE.
5550          ;          LET (R1)+ := #DATCNT          ;SET LENGTH TO MAX FOR UNKNOWN LENGTHS.
5551          ;          LET (R1)+ := #77777          ;RECORD COUNT = MAX FOR WHOLE TAPE.
5552          ;          LET (R1)+ := #RANP          ;PATTERN = RANDOM.
5553          ;          LET (R1) := #END          ;STORE END OF SEQUENCE CODE IN TABLE.
5554          ;          JSR PC,EXALL          ;GO EXECUTE READ REV. OF ENTIRE TAPE.
5555 026750          105037 003531              LET ALLEOT :B= #0          ;CLEAR ALL UNITS @ EOT FLAG
5556 026754          012701 003542              LET R1 := #CMOSEQ          ;TOP OF COMMAND TABLE
5557 026760          004737 006724              JSR PC,SETRW          ;ISSUE A REWIND COMMAND (TWR)
5558 026764          012721 177777              LET (R1)+ := #END          ;TERMINATOR
5559 026770          004737 006744              JSR PC,EXALL          ;ISSUE THE REWIND
5560
5561 026774          104432                      EXIT TST
5562 026776          000002                      TRAP C$EXIT
5563          .EVEN                      .WORD L10037-.
5564
5565 027000          L10037:                      ENDTST
5566 027000          104401                      TRAP C$ETST
5567 027000

```

```

5567
5568 .SBTTL TEST 6: EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.
5569
5570
5571 ;**
5572 ; TEST TO EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.
5573 ; -
5574 027002          BGNTST
      027002      T6::
5575
5576 027002          LET RANDOM :B= #0          ;CLEAR RANDOM MODE FLAG.
      027002 105037 003521                    ;CLR RANDOM
5577 027006          LET EXPBOT :B= #1          ;SET EXPECT BOT FLAG.
      027006 112737 000001 003520              ;MOV #1,EXPBOT
5578 027014          LET IRE :B= PIRE          ;MOVE INHIBIT RFC ERROR REPORT FLAG.
      027014 113737 002213 003526              ;MOV PIRE,IRE
5579 027022          JSR PC,SETCH              ;CMD 1 = SET CHARACTERISTIC.
5580 027026          LET CMDSEQ+2 := CHAR      ;MOVE CHAR CODE FROM P TBL TO SEQ TBL.
      027026 013737 002220 003544              ;MOV CHAR,CMDSEQ+2
5581 027034          LET R2 := #CMD0          ;R2 POINTS TO CMD2 IN SOFT P TABLE.
      027034 012702 002222                    ;MOV #CMD0,R2
5582 027040          JSR PC,PTCMDS             ;MOVE CMD 2 FROM P TBL TO SEQ TBL.
5583 027044          JSR PC,PTCMDS             ;MOVE CMD 3 FROM P TBL TO SEQ TBL.
5584 027050          JSR PC,PTCMDS             ;MOVE CMD 4 FROM P TBL TO SEQ TBL.
5585 027054          JSR PC,PTCMDS             ;MOVE CMD 5 FROM P TBL TO SEQ TBL.
5586 027060          JSR PC,PTCMDS             ;MOVE CMD 6 FROM P TBL TO SEQ TBL.
5587 027064          JSR PC,PTCMDS             ;MOVE CMD 7 FROM P TBL TO SEQ TBL.
5588 027070          JSR PC,PTCMDS             ;MOVE END CMD FROM P TBL TO SEQ TBL.
5589 027074          LET JLOOP := #0          ;CLEAR JMP CMD LOOP COUNT.
      027074 005037 003450                    ;CLR JLOOP
5590 027100          LET STAFLG :B= #0          ;CLEAR START FLAG
      027100 105037 003534                    ;CLR STAFLG
5591 027104          LET R1 := #CMDSEQ         ;INIT SEQUENCE TABLE POINTER.
      027104 012701 003542                    ;MOV #CMDSEQ,R1
5592
5593          000000          $BRJMP=0          ;ENABLE JMP SUBSTITUTION FOR BR, IF NECESSARY.
5594
5595 027110          3$: WHILE (R1) NE #END DO    ;WHILE THERE ARE CMDS LEFT IN SEQUENCE TBL:
      027110          021127 177777              50472$:
      027114          001002                    CMP (R1),#END
      027116          000137 027560              BNE #6
5596 027122          022711 000040              JMP 50473$
5597 027126          001024                    ;IS THIS A JUMP CMD?
5598 027130          062701 000002              ;BR IF NOT.
      027130          012137 003452              ;POINT TO BR.
5599 027134          022137 003450              ADD #2,R1
5600 027140          001003                    ;SAVE BR (LOCATION).
      027146          062701 000002              ;HAS LOOP COUNT BE SATISFIED?
5601 027144          000756                    ;IF NOT, JMP AGAIN.
5602 027146          000756                    ;IF SO, ADJUST SEQ POUNTER
      027146          000756                    ADD #2,R1
5603 027152          000756                    ;AND GO TO NEXT COMMAND.
5604 027154          005237 003450              1$: LET JLOOP := JLOOP + #1
      027154          005237 003450              ;UPDATE THE LOOP COUNT.
5605 027160          012701 003542              LET R1 := #CMDSEQ
      027160          005337 003452              ;INIT CMD SEQ TABLE POINTER.
5606 027164          005337 003452              2$: DEC JLOC
      027164          005337 003452              ;DECR LOCATION COUNTER.

```

SEQ 0165

5607	027170	001747		BEQ 3\$		;IF THIS IS THE RIGHT LOCATION TO JMP TO, GO SET UP.
5608	027172			LET R1 := R1 + #10		;IF NOT, UPDATE SEQ POINTER TO NEXT CMD. ADD #10,R1
5609	027176	062701	000010	BR 2\$		;DO IT AGAIN.
5610	027200	022711	000020	CMP #DLY.C,(R1)	6\$:	;DELAY?
5611	027204	001026		BNE 4\$		;BR IF NOT.
5612	027206			LET R1 := R1 + #4		;R1 = LOCATION OF N COUNT. ADD #4,R1
5613	027212	062701	000004			
5614	027212	011137	003446	LET TIME2 := (R1)		;SAVE N COUNT. MOV (R1),TIME2
5615	027216			DELAY 10.	7\$:	;GO TO SUPER WAIT 1 MSEC.
	027216	012727	000012			MOV #10.,(PC).
	027222	000000				.WORD 0
	027224	013727	002116			MOV L\$DLY,(PC).
	027230	000000				.WORD 0
	027232	005367	177772			DEC 6(PC)
	027236	001375				BNE . 4
	027240	005367	177756			DEC -22(PC)
	027244	001367				BNE . 20
5615	027246	005337	003446	DEC TIME2		
5616	027252	001361		BNE 7\$		
5617	027254			LET R1 := R1 + #4		;POINT TO NEXT CMD. ADD #4,R1
5618	027260	000713		BR 3\$		;GO CHECK NEXT CMD.
5619	027262	004737	007706	JSR PC,SETUP	4\$:	;GC SETUP THE COMMAND BLOCK.
5620	027266			WHILE NCNT LT NCNT1 DO		;WHILE THERE ARE RECORDS REMAINING: 50474\$: CMP NCNT,NCNT1 BLT .+6 JMP 50475\$
	027266	023737	003420			
	027274	002402	003422			
	027276	000137	027550			
5621	027302	004737	007600	JSR PC,CMDAC		;STORE CMD ASCII IN ERROR MSG.
5622	027306	004737	007240	JSR PC,EXSUB		;ISSUE CMD TO ALL,AWAIT INTS,CHECK STATUS.
5623	027312			IF CMDWRD EQ #GES THEN		;IF CMD IS GET STATUS THEN: CMP CMDWRD,#GES BEQ .+6 JMP 50476\$
	027312	023727	003426			
	027320	001402	100017			
	027322	000137	027332			
5624	027326	004737	017624	JSR PC,PRXST		;PRINT EXTENDED STATUS REGISTERS.
5625	027332			ENDIF		
	027332					50476\$: ;CHECK HALT AFTER EACH CMD FLAG. ;SET ALL UNITS AT BOT/EOT. MOV #1,R2
5626	027332	004737	017706	JSR PC,CKHAE		
5627	027336			LET R2 := #1		
	027336	012702	000001			
5628	027342	004737	017300	JSR PC,FIRSTU		;FIND FIRST UNIT.
5629	027346			WHILE DEVTBL(R5) NE #END DO		;WHILE THERE ARE MORE UNITS: 50477\$: CMP DEVTBL(R5),#END BNE .+6 JMP 50500\$
	027346	026527	002536			
	027354	001002	177777			
	027356	000137	027454			
5630	027362			IF #MOD.CO SETIN CMDWRD THEN		;IF CMD IS REVERSE THEN: BIT #MOD.CO,CMDWRD BNE .+6 JMP 50501\$
	027362	032737	000400			
	027370	001002	003426			
	027372	000137	027420			
5631	027376			IF #XO.BOT NOTSETIN EOTFLG(R5) THEN		;IF NOT AT BOT THEN: BIT #XO.BOT,EOTFLG(R5) BEQ .+6 JMP 50502\$
	027376	032765	000002			
	027404	001402	003506			
	027406	000137	027414			
5632	027412			LET R2 := #C		;CLEAR EOT/BOT FLAG.

5633	027412	005002			ENDIF	CLR R2
5634	027414				ELSE	50502\$: ;ELSE IF CMD IS NOT REVERSE: JMP 50503\$
5635	027420	000137	027446		IF #X0.EOT NOTSETIN EOTFLG(R5) OR #CMD.CO NOTSETIN CMDWRD THEN	50501\$: BIT #X0.EOT,EOTFLG(R5) BEQ 50504\$ BIT #CMD.CO,CMDWRD BEQ .+6 JMP 50505\$
	027420	032765	000001	003506		
	027426	001406				
	027430	032737	000001	003426		
	027436	001402				
	027440	000137	027446			
	027444					50504\$: ;IF NOT AT EOT OR NOT A MOTION CMD THEN: ;CLEAR EOT/BOT FLAG. CLR R2
5636	027444				LET R2 := #0	
5637	027444	005002			ENDIF	50505\$:
5638	027446				ENDIF	50503\$:
5639	027446				JSR PC,NEXTU	
5640	027446	004737	017346		ENDDO	BR 50477\$
5641	027452					50500\$: ;IF ALL UNIT ARE AT EOT/BOT THEN: CMP R2,#1 BEQ .+6 JMP 50506\$
	027452	000735			IF R2 EQ #1 THEN	
5642	027454					MOV NCNT,NCNT1 INC NCNT1
	027454	020227	000001		LET NCNT1 := NCNT + #1	50506\$: ;FORCE TERMINATION OF COMMAND. MOV NCNT,NCNT1 INC NCNT1
	027460	001402				INCB ALLEOT
	027462	000137	027530		LET ALLEOT :B= ALLEOT + #1	50507\$: ;FLAG ALL UNITS AT EOT/BOT TO ALLOW VERIFY OF DATA INCB ALLEOT
5643	027466	013737	003420	003422		50506\$: ;WHEN WRITING IS CURRENT COMMAND CMP CMDLG,#2 BEQ .+6 JMP 50507\$
	027474	005237	003422		IF CMDLG EQ #2 THEN	
5644	027500				JSR PC,T5WEOT	50507\$: ;GO WRITE/READ REV ONE RECORD BEYOND EOT JMP 50510\$
	027500	105237	003531		ENDIF	50506\$: ;WHEN NOT ALL #EOT, CLEAR FLAG CLR ALLEOT
5645	027504					50510\$: ;UPDATE RECORD COUNT. INC NCNT
	027504	023727	003434	000002	IF CMDLG EQ #2 THEN	50506\$: ;WHEN WRITING IS CURRENT COMMAND CMP CMDLG,#2 BEQ .+6 JMP 50507\$
	027512	001402				50507\$: ;GO WRITE/READ REV ONE RECORD BEYOND EOT JMP 50510\$
	027514	000137	027524		JSR PC,T5WEOT	50506\$: ;WHEN NOT ALL #EOT, CLEAR FLAG CLR ALLEOT
5646	027520	004737	027640		ENDIF	50510\$: ;UPDATE RECORD COUNT. INC NCNT
5647	027524				ELSE	50506\$: ;WHEN WRITING IS CURRENT COMMAND CMP CMDLG,#2 BEQ .+6 JMP 50507\$
5648	027524	000137	027534			50507\$: ;GO WRITE/READ REV ONE RECORD BEYOND EOT JMP 50510\$
	027530				LET ALLEOT :B= #0	50506\$: ;WHEN NOT ALL #EOT, CLEAR FLAG CLR ALLEOT
5649	027530				LET ALLEOT :B= #0	50506\$: ;WHEN NOT ALL #EOT, CLEAR FLAG CLR ALLEOT
5650	027534	105037	003531		ENDIF	50510\$: ;UPDATE RECORD COUNT. INC NCNT
5651	027534				LET NCNT := NCNT + #1	50506\$: ;WHEN WRITING IS CURRENT COMMAND CMP CMDLG,#2 BEQ .+6 JMP 50507\$
5652	027534	005237	003420		LET PCMDWD := CMDWRD	50506\$: ;WHEN WRITING IS CURRENT COMMAND CMP CMDLG,#2 BEQ .+6 JMP 50507\$
5653	027540	013737	003426	003432	ENDDO	50506\$: ;WHEN WRITING IS CURRENT COMMAND CMP CMDLG,#2 BEQ .+6 JMP 50507\$
	027546	000647				50506\$: ;WHEN WRITING IS CURRENT COMMAND CMP CMDLG,#2 BEQ .+6 JMP 50507\$
5654	027550	004737	016142		JSR PC,VFYDAT	50506\$: ;WHEN WRITING IS CURRENT COMMAND CMP CMDLG,#2 BEQ .+6 JMP 50507\$
5655						50506\$: ;WHEN WRITING IS CURRENT COMMAND CMP CMDLG,#2 BEQ .+6 JMP 50507\$
5656						50506\$: ;WHEN WRITING IS CURRENT COMMAND CMP CMDLG,#2 BEQ .+6 JMP 50507\$

SEQ 0167

5657	027554			ENDDO			JMP	50472\$
	027554	000137	027110				50473\$:	
	027560							
5658								
5659		177777		\$BR.JMP= 1	;TURN OFF JMP SUBSTITUTION (SPMACJ CONTROL, ONLY).			
5660	027560			LET ALLEOT :B= #0	;NOT AT EOT			
	027560	105037	003531			CLRB	ALLEOT	
5661	027564			LET RPTFLG :B= #1	;REQUEST A REPORT			
	027564	112737	000001	003523		MOVB	#1,RPTFLG	
5662	027572			LET R1 := #CMDSEQ	;SET UP THE COMMAND TABLE ADDRESS	MOV	#CMDSEQ,R1	
	027572	012701	003542					
5663	027576	004737	006724	JSR PC,SETRW	;STORE REWIND COMMAND			
5664	027602			LET (R1) := #END	;TERMINATE THE TABLE			
	027602	012711	177777			MOV	#END.(R1)	
5665	027606	004737	006744	JSR PC,EXALL	;ISSUE THE REWINDS			
5666								
5667								
5668	027612			EXIT TST				
	027612	104432						
	027614	000130				TRAP	C\$EXIT	
5669						.WORD	L10040-	

```

5671
5672      ;      SUBROUTINE TO MOVE A COMMAND FROM THE SOFTWARE P TABLE TO
5673      ;      THE COMMAND SEQUENCE TABLE.
5674      ;      INPUTS:      R2 = POINTER TO SOFT "P" TABLE
5675      ;      OUTPUTS:
5676      ;      REGISTERS:    R3.
5677      ;      CALLS:
5678
5679      PTCMDS: LET R3 := (R2)+ #1 SHIFT +1 ;R3 = COMMAND TABLE INDEX.
5680      027616 012203      MOV      (R2)+,R3
5681      027620 005303      DEC      R3
5682      027622 006303      ASL      R3
5683      027624 016321 003644 LET (R1)+ := CMDTBL(R3) ;MOVE COMMAND WORD.
5684      027630 012221      MOV      CMDTBL(R3),(R1)+
5685      027632 012221      ;MOVE # OF BYTES.
5686      027634 012221      MOV      (R2)+,(R1)+
5687      027636 000207      ;MOVE # OF OPERATIONS.
5688      ;MOVE PATTERN CODE.
5689      ;MOVE (R2)+,(R1)+
5690      RTS PC
5691
5692      ;      SUBROUTINE TO WRITE THEN READ REVERSE ONE RECORD BEYOND EOT
5693      ;      INPUTS:
5694      ;      OUTPUTS:
5695      ;      REGISTERS:
5696      ;      CALLS:      CMDAC,EXSUB,CKHAE
5697
5698      TSWEOT: NOP
5699      NOP
5700      JSR PC,EXSUB ;WRITE ONE RECORD BEYOND EOT
5701      JSR PC,CKHAE ;SO THAT READ SHORTER STOP DISTANCE
5702      ;SHALL POSITION HEAD IN CLEAN IRG GAP
5703      LET PCMDWD := CMDWRD ;REPOSITION TAPE
5704      LET CMDWRD := #RDR ;BEFORE EXTRA RECORD
5705      LET CMDLG := #4 ;BY READING REVERSE
5706      LET CMDPKT := CMDWRD CLR.BY #BRF.C
5707      MOV      CMDWRD,PCMDWD
5708      MOV      #RDR,CMDWRD
5709      MOV      #4,CMDLG
5710      MOV      CMDWRD,CMDPKT
5711      BIC      #BRF.C,CMDPKT
5712      LET CMDSAV := CMDPKT ;THAT RECORD TO ALLOW
5713      LET CMDPKT+CP.ADL := DATARD ;NEXT COMMAND IN THE
5714      MOV      CMDPKT,CMDSAV
5715      MOV      DATARD,CMDPKT+CP.ADL
5716      JSR PC,CMDAC ;TABLE TO BE EXECUTED
5717      JSR PC,EXSUB
5718      JSR PC,CKHAE
5719      RTS PC
5720      .EVEN
5721      ENDTST
5722
5723      L10040:
5724      TRAP      C$ETST
5725
5726      ENDMOD

```

PARAMETER CODING MACRO M1200 21 MAR 84 16:45 PAGE 114
 TEST 6: EXECUTE OPERATOR SELECTED COMMAND SEQUENCE.

SEQ 0169

```

5712          .TITLE PARAMETER CODING
5723
5724          .SBTTL  HARDWARE PARAMETER CODING SECTION
5733
5734 027746          BGNMOD
5735
5736          ;**
5737          ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
5738          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
5739          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
5740          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
5741          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
5742          ; WITH THE OPERATOR.
5743          ;--
5744
5745 027746          BGNHRD
5746 027746 000024
5747 027750          L$HARD::
5748
5752 027750          GPRMA  TS4ADR,0,0,160002,177564,YES
5753 027750 000031          .WORD  L10041-L$HARD/2
5754 027752 027774          .WORD  TS4ADR
5755 027754 160002          .WORD  T$LOLIM
5756 027756 177564          .WORD  T$HILIM
5757
5758 027760          GPRMD  TS4VCT,2,0,777,60,776,YES
5759 027760 001032          .WORD  T$CODE
5760 027762 030011          .WORD  TS4VCT
5761 027764 000777          .WORD  777
5762 027766 000060          .WORD  T$LOLIM
5763 027770 000776          .WORD  T$HILIM
5764
5765 027772          EXIT HRD
5766 027772 013004          .WORD  T$CODE
5767
5768 027774          NLIST  BEX
5769 030011 124 123 123 TS4ADR: .ASCIZ /TSSR ADDRESS/
5770 030011 126 105 103 TS4VCT: .ASCIZ /VECTOR/
5771          .LIST  BEX
5772          .EVEN
5773
5774 030020          ENDHRD
5775
5776 030020          L10041:
5777          .EVEN

```

```

5772 .SBTTL SOFTWARE PARAMETER CODING SECTION
5773
5774
5775 ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
5776 ; THAT ARE USED BY THE SUPERVISOR TO BUILD P TABLES. THE
5777 ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
5778 ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
5779 ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
5780 ; WITH THE OPERATOR.
5781 ;
5782
5783 030020      BGNSFT
5784 030020      000531
5785 030022      L$SOFT::
5786
5791
5792 030022      GPRML   CLRM,0,1,YES
5793 030022      000130
5794 030024      030600
5795 030026      000001
5796 030030      GPRML   RRVM,0,400,YES
5797 030030      000130
5798 030032      030617
5799 030034      000400
5800 030036      GPRML   HAEM,2,1,YES
5801 030036      001130
5802 030040      030646
5803 030042      000001
5804 030044      GPRML   RCVERM,2,400,YES
5805 030044      001130
5806 030046      030672
5807 030050      000400
5808 030052      GPRML   IRECM,4,1,YES
5809 030052      002130
5810 030054      030714
5811 030056      000001
5812 030060      XFERT   NEXTSP
5813 030060      004024
5814 030062      GPRML   BADTM,4,400,YES
5815 030062      002130
5816 030064      030735
5817 030066      000400
5818 030070      NEXTSP: GPRML   DINTM,6,1,YES
5819 030070      003130
5820 030072      030762
5821 030074      000001
5822 030076      GPRML   IREM,6,400,YES
5823 030076      003130
5824 030100      031005
5825 030102      000400
5826 030104      GPRML   RAMM,10,1,YES
5827 030104      004130
5828 030106      031040
5829 030110      000001
5830 030112      GPRML   EWPT,12,1,YES
5831 030112      005130

```

.WORD L10042 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

	030114	031064			.WORD	EMPTY
	030116	000001			.WORD	1
5804	030120		GPRM	CHGM,12,400,YES		
	030120	005130			.WORD	TSCODE
	030122	031122			.WORD	CHGM
	030124	000400			.WORD	400
5805	030126		XFERF	ENDSP1		
	030126	127044			.WORD	TSCODE
5806	030130		GPRMD	CHARM,14,0,377,0,177,YES		
	030130	006032			.WORD	TSCODE
	030132	031141			.WORD	CHARM
	030134	000377			.WORD	377
	030136	000000			.WORD	TSLOLIM
	030140	000777			.WORD	TSMILIM
5807	030142		GPRMD	CMD2M,16,D,37,1,33,YES		
	030142	007052			.WORD	TSCODE
	030144	031166			.WORD	CMD2M
	030146	000037			.WORD	37
	030150	000001			.WORD	TSLOLIM
	030152	000033			.WORD	TSMILIM
5808	030154		GPRMD	BPCRM,20,D,1,1,DATCNT,YES		
	030154	010052			.WORD	TSCODE
	030156	031174			.WORD	BPCRM
	030160	177777			.WORD	1
	030162	000001			.WORD	TSLOLIM
	030164	010000			.WORD	TSMILIM
5809	030166		GPRMD	NUMBM,22,D,-1,1,77777,YES		
	030166	011052			.WORD	TSCODE
	030170	031206			.WORD	NUMBM
	030172	177777			.WORD	-1
	030174	000001			.WORD	TSLOLIM
	030176	077777			.WORD	TSMILIM
5810	030200		GPRMD	PATTM,24,D,17,0,10,YES		
	030200	012052			.WORD	TSCODE
	030202	031226			.WORD	PATTM
	030204	000017			.WORD	17
	030206	000000			.WORD	TSLOLIM
	030210	000010			.WORD	TSMILIM
5811	030212		GPRMD	CMD3M,26,D,37,1,33,YES		
	030212	013052			.WORD	TSCODE
	030214	031240			.WORD	CMD3M
	030216	000037			.WORD	37
	030220	000001			.WORD	TSLOLIM
	030222	000033			.WORD	TSMILIM
5812	030224		GPRMD	BPCRM,30,D,1,1,DATCNT,YES		
	030224	014052			.WORD	TSCODE
	030226	031174			.WORD	BPCRM
	030230	177777			.WORD	1
	030232	000001			.WORD	TSLOLIM
	030234	010000			.WORD	TSMILIM
5813	030236		GPRMD	NUMBM,32,D,-1,1,77777,YES		
	030236	015052			.WORD	TSCODE
	030240	031206			.WORD	NUMBM
	030242	177777			.WORD	1
	030244	000001			.WORD	TSLOLIM
	030246	077777			.WORD	TSMILIM
5814	030250		GPRMD	PATTM,34,D,17,0,10,YES		

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

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

```

030544 177777
030546 000001
030550 010000
5835 030552          GPRMD  NUMBM,!02.D,-1,1,77777,YES
030552 041052
030554 031206
030556 177777
030560 000001
030562 077777
5836 030564          GPRMD  PATTM,104.D,17,0,10,YES
030564 042052
030566 031226
030570 000017
030572 000000
030574 000010
5837 030576          ENDS:
5838 030576          XFER   JMPMSG
030576 120004

```

```

.WORD -1
.WORD T$LOLIM
.WORD T$HILIM

.WORD T$CODE
.WORD NUMBM
.WORD -1
.WORD T$LOLIM
.WORD T$HILIM

.WORD T$CODE
.WORD PATTM
.WORD 17
.WORD T$LOLIM
.WORD T$HILIM

.WORD T$CODE

```

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

```

031231      124      105      122
031234      116      000
5868
5869
5870 031236
5871 031236
031236 023004
5872
5873
5874 031240      103      115      104  CMD3M:
5875 031246      103      115      104  CMD4M:
5876 031254      103      115      104  CMD5M:
5877 031262      103      115      104  CMD6M:
5878 031270      103      115      104  CMD7M:
5879 031276      103      115      104  CMD8M:
5880
5881 031304
031304
5882
5883
5884
5885
5886
5887 031304
5888
5889
5890
5891 031504
031504 031520
031506 000004
031510
5892 031510

.LIST    BEX
.EVEN
JMPMS2:
EXIT SFT
.WORD    T$CODE

.NLIST    BEX
CMD3M: .ASCIZ "CMD/3"
CMD4M: .ASCIZ "CMD/4"
CMD5M: .ASCIZ "CMD/5"
CMD6M: .ASCIZ "CMD/6"
CMD7M: .ASCIZ "CMD/7"
CMD8M: .ASCIZ "CMD/8"
.LIST    BEX
ENDSFT
.EVEN
L10042:
;*****
;*****
;        PATCH AREA
PATCH:: .BLKW    64.
;*****
;*****
LASTAD
.EVEN
.WORD T$FREE
.WORD T$SIZE
L$LAST::
ENDMOD
```

PARAMETER CODING
HARD CODED P TBL

MACRO M1200 21-MAR 84 16:45 PAGE 118

SEQ 0177

```
5894 .SBTTL HARD CODED P-TBL
5895
5896 ***
5897 ;DIAG IS PRE-PARAMETERIZED PER TBL
5898 ;
5899
5900 BGNSETUP 1
5901 BGNPTAB
      031510 000000
      031512 000002
      031514
5902 031514 172522
5903 031516 000224
5904 031520
      031520
5905 031520
5906
5907 000001

      L10043:
          172522
          224
          ENDP TAB
      L10045:
          ENDSETUP

      .END

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

PARAMETER CODING
SYMBOL TABLE

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

SEQ 0178

ACK.C = 100000 G	BTMSG2 015021	CP.CMD= 000000 G	C\$SVEC= 000037	EXSUB 007240 G
ADR = 000020 G	BTMSG3 015076	CP.CNT= 000006 G	C\$TPRI= 000013	E\$END = 002100
ALLEOT 003531 G	BTPT 003516 G	CRLF 005275 G	DATAWD 003416 G	E\$LOAD= 000035
ASSEMB= 000010	BTRPT 020446	CRLFSP 005300 G	DATCNT= 010000 G	FATSM 004455 G
ATTNM 004417 G	BT0 003000 G	CTCC 003456 G	DATRAT 003412 G	FIRSTU 017300 G
AUDRPM 004727 G	BT1 003052 G	CVC.C = 040000 G	DEVTBL 002536 G	FMT.CO= 000040 G
AUTODM 023450	BT2 003124 G	C\$AU = 000052	DFPTBL 002176 G	FMT.C1= 000100 G
BADTM 030735	BT3 003176 G	C\$AUTO= 000061	DFTSCH= 000040 G	FTLCNT 003320 G
BADTSW 002211 G	BYTECO= 000403	C\$BRK = 000022	DIA = 100006 G	FUNRM 004435 G
BFSEQ 024616	CHAR 002220 G	C\$BSEG= 000004	DIABLK= 003414 G	F\$AU = 000015
BFSEQ0 024642	CHARM 031141	C\$BSUB= 000002	DIACNT= 000020 G	F\$AUTO= 000020
BFSEQ1 024714	CHGFLG 002217 G	C\$CEFG= 000045	DIAGMC= 000000	F\$BGN = 000040
BFSEQ2 024726	CHGM 031122	C\$CLCK= 000062	DINT 002212 G	F\$CLEA= 000007
BFSEQ3 025020	CHKERR 012160 G	C\$CLEA= 000012	DINTM 030762	F\$DU = 000016
BFSEQ4 025072	CH.EAI= 000040 G	C\$CLOS= 000035	DLY = 000020 G	F\$END = 000041
BFSEQ5 025134	CH.ERI= 000020 G	C\$CLP1= 000006	DLY.C = 000020 G	F\$HARD= 000004
BFSEQ6 025206	CH.ESS= 000200 G	C\$CVEC= 000036	DRI = 100013 G	F\$HW = 000013
BFSEQ7 025240	CKDATA 016664 G	C\$DCLN= 000044	DROPDN 004700 G	F\$INIT= 000006
BFSEQ8 025272	CKDCNT 017274	C\$DODU= 000051	DROPEP 003527 G	F\$JMP = 000050
BFSEQ9 025324	CKDFF 017276	C\$DRPT= 000024	DROPN 017622	F\$MOD = 000000
BFSE10 025346	CKHAE 017706 G	C\$DU = 000053	DROPU 017402 G	F\$MSG = 000011
BGNFLG= 003464	CKHRTN 017774	C\$EDIT= 000003	DROPUA 017532	F\$PROT= 000021
BINC 015722	CLN = 101012 G	C\$ERDF= 000055	DRORTN 017610	F\$PWR = 000017
BIT0 = 000001 G	CLRERR 011656 G	C\$ERHR= 000056	DTAERM 005460 G	F\$RPT = 000012
BIT00 = 000001 G	CLRFLG 002204 G	C\$ERRO= 000060	DTAER2 004761 G	F\$SEG = 000003
BIT01 = 000002 G	CLRM 030600	C\$ERSF= 000054	DTAER3 005030 G	F\$SOFT= 000005
BIT02 = 000004 G	CMDAC 007600 G	C\$ERSO= 000057	DTAER4 005072 G	F\$SRV = 000010
BIT03 = 000010 G	CMDASC 003732 G	C\$ESCA= 000010	DTAER5 005113 G	F\$SUB = 000002
BIT04 = 000020 G	CMD0D 002222 G	C\$ESEG= 000005	EF.CON= 000036 G	F\$SW = 000014
BIT05 = 000040 G	CMDLG 003434 G	C\$ESUB= 000003	EF.NEW= 000035 G	F\$TEST= 000001
BIT06 = 000100 G	CMDPKM 004164 G	C\$ETST= 000001	EF.PWR= 000034 G	GCMOA 007654 G
BIT07 = 000200 G	CMDPKT 002314 G	C\$EXIT= 000032	EF.RES= 000037 G	GENPAT 010264 G
BIT08 = 000400 G	CMDSAV 003430 G	C\$GETB= 000026	EF.STA= 000040 G	GES = 100017 G
BIT09 = 001000 G	CMDSEQ 003542 G	C\$GETW= 000027	EINC 015730	GETSTM 005241 G
BIT1 = 000002 G	CMDSE2 003552 G	C\$GMAN= 000043	END = 177777 G	GIT 010642
BIT10 = 002000 G	CMDTBL 003644 G	C\$GPHR= 000042	ENDERF= 003476	GOWAIT 011274 G
BIT11 = 004000 G	CMDWRD 003426 G	C\$GPLO= 000030	ENDFLG= 003534	GSCPCK 002324 G
BIT12 = 010000 G	CMD.CO= 000001 G	C\$GPRI= 000040	ENDSP 030576	G\$CNT0= 000200
BIT13 = 020000 G	CMD.C1= 000002 G	C\$INIT= 000011	ENDSP1 030404	G\$DELM= 000372
BIT14 = 040000 G	CMD.C2= 000004 G	C\$INLP= 000020	ENDSP2 030406	G\$DISP= 000003
BIT15 = 100000 G	CMD.C3= 000010 G	C\$MANI= 000050	EOTFLG 003506 G	G\$EXCP= 000400
BIT2 = 000004 G	CMD.C4= 000020 G	C\$MEM = 000031	ERCVER 002207 G	G\$HILI= 000002
BIT3 = 000010 G	CMD2M 031156	C\$MSG = 000023	ERLOG 003472 G	G\$LOLI= 000001
BIT4 = 000020 G	CMD3M 031240	C\$OPEN= 000034	ERRREC 003475 G	G\$NO = 000000
BIT5 = 000040 G	CMD4M 031246	C\$PNTB= 000014	ERS = 100411 G	G\$OFFS= 000400
BIT6 = 000100 G	CMD5M 031254	C\$PNTF= 000017	ERSFLG 003533 G	G\$OF SI= 000376
BIT7 = 000200 G	CMD6M 031262	C\$PNTS= 000016	EVL = 000004 G	G\$PRMA= 000001
BIT8 = 000400 G	CMD7M 031270	C\$PNTX= 000015	EWMSG 012526	G\$PRMD= 000002
BIT9 = 001000 G	CMD8M 031276	C\$QIO = 000377	EWPRNT 002216 G	G\$PRML= 000000
BOE = 000400 G	CMPDAT 003410 G	C\$RDBU= 000007	EWPT 031064	G\$RADA= 000140
BORERS 015146 G	CNTBGN= 002560	C\$REFG= 000047	EWTRY 012632 G	G\$RAD8= 000000
BPCRM 031174	CNTEND= 003330	C\$RESE= 000033	EXALL 006744 G	G\$RADD= 000040
BRCPK 002330 G	CNTLEN= 000550 G	C\$REVI= 000003	EXARTN 007236	G\$RADL= 000120
BRFCNT 003424 G	CODELM 004054 G	C\$RFLA= 000021	EXCRTN 011272	G\$RADO= 000020
BRF.C = 004000 G	COUNTE= 050416	C\$RPT = 000025	EXCUTE 010654 G	G\$XFER= 000004
BTADDR 002550 G	CP.ADM= 000004 G	C\$SEFG= 000046	EXPBOT 003520 G	G\$YES = 000010
BTMSG1 014734	CP.ADL= 000002 G	C\$SPRI= 000041		HAE 002206 G

PARAMETER CODING
SYMBOL TABLE

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

SEQ 0179

HAEM	030646	L\$CLEA	023602 G	L10013	023446	OFLINH	005211 G	RANDOM	003521 G
HALTM	004124 G	L\$CO	002032 G	L10014	023642	ONEFIL	000001	RAMP	000007 G
HELP	000000	L\$DEPO	002011 G	L10015	023706	OPFLAG	003540 G	RAN	025712
HERE	003340	L\$DESC	002140 G	L10016	024002	OPP.C	020000 G	RANS	000000 G
HOE	010000 G	L\$DESP	002076 G	L10017	025370	O\$APTS	000000	RANSC	000000
HRDCNT	003310 G	L\$DEVP	002060 G	L10020	024136	O\$AU	000001	RANW	026012
IBE	010000 G	L\$DISP	002124 G	L10021	024162	O\$BGNR	000001	RANWR	025766
IDU	000040 G	L\$DLY	002116 G	L10022	024202	O\$BGNS	000001	RANWV	026000
IER	020000 G	L\$DTP	002040 G	L10023	024222	O\$DU	000001	RAN1	010644 G
IE.C	000200 G	L\$DTYP	002034 G	L10024	024242	O\$ERRT	000000	RAN2	010646 G
INIT10	021664	L\$DU	023644 G	L10025	024262	O\$GNSW	000001	RAN3	010650 G
INIT15	022104	L\$DUT	002072 G	L10026	024302	O\$POIN	000001	RCVERM	030672
INIT16	022124	L\$DVTY	002166 G	L10027	024322	O\$SETU	000001	RDF	0104001 G
INTFLG	003476 G	L\$EF	002052 G	L10030	024342	PASCNT	003260 G	RDR	0104401 G
INTPRI	000340 G	L\$ENVI	002044 G	L10031	024362	PATCH	031304 G	RECCNT	003330 G
IRE	003526 G	L\$ETP	002102 G	L10032	024420	PATERN	003454 G	RECLOG	003471 G
IREC	002210 G	L\$EXP1	002046 G	L10033	024604	PATRO	010350 G	RECRED	006546
IRECM	030714	L\$EXP4	002064 G	L10034	026044	PATR0	010406 G	RECTAP	006602 G
IREM	031005	L\$EXP5	002066 G	L10035	026522	PATR2	010426 G	RECUD	012012 G
ISR	000100 G	L\$HARD	027750 G	L10036	026666	PATR3	010436 G	REPEAT	050236
IXE	004000 G	L\$HIME	002120 G	L10037	027000	PATR4	010462 G	RERM	004632 G
I\$AU	000041	L\$HPCP	002016 G	L10040	027744	PATR5	010474 G	RETRY	050234
I\$AUTO	000041	L\$HPTP	002022 G	L10041	030020	PATR6	010506 G	RETRYC	003464 G
I\$CLN	000041	L\$HW	002176 G	L10042	031304	PATR7	010526 G	REWRT	015322
I\$DU	000041	L\$ICP	002104 G	L10043	031514	PATR8	010652 G	RFBC	002660 G
I\$HRD	000041	L\$INIT	021664 G	L10045	031520	PATTLB	010326	RFCERM	004336 G
I\$INIT	000041	L\$LADP	002026 G	MBR	010012 G	PATTM	031226	RFREC	002760 G
I\$MOD	000041	L\$LAST	031510 G	MEMOM	023054	PCMDWD	003432 G	RFUNR	002770 G
I\$MSG	000041	L\$LOAD	002100 G	MISCFG	003537 G	PIRE	002213 G	RLEXM	004372 G
I\$PROT	000040	L\$LUN	002074 G	MOD.CO	000400 G	PNT	001000 G	RNF	0125401 G
I\$PTAB	000041	L\$MREV	002050 G	MOD.C1	001000 G	PRI	002000 G	RNOPSC	0177740 G
I\$PWR	000041	L\$NAME	002000 G	MOD.C2	002000 G	PRI00	000000 G	RNR	0105401 G
I\$RPT	000041	L\$PRIO	002042 G	MOD.C3	004000 G	PRI01	000040 G	RNYM	004506 G
I\$SEG	000041	L\$PROT	021656 G	MOVMSG	011726 G	PRI02	000100 G	RPF	0105001 G
I\$SETU	000041	L\$PRT	002112 G	MSGCNT	000016 G	PRI03	000140 G	RPR	0125001 G
I\$SFT	000041	L\$REPP	002062 G	MSGPKA	002506 G	PRI04	000200 G	RPTCNT	003466 G
I\$SRV	000041	L\$REV	002010 G	MSGPKT	002340 G	PRI05	000240 G	RPTFLG	003523 G
I\$SUB	000041	L\$RPT	020052 G	MSGPK0	002356 G	PRI06	000300 G	RPT1A	020714
I\$TST	000041	L\$SOFT	030022 G	MSGPK1	002374 G	PRI07	000340 G	RPT1B	020771
JLOC	003452 G	L\$SPC	002056 G	MSGPK2	002412 G	PRXST	017624 G	RPT1C	021042
JLOOP	003450 G	L\$SPCP	002020 G	MSGPK3	002430 G	PTCMDS	027616	RPT1D	021113
JMP	000040 G	L\$SPTP	002024 G	MS.RFC	000004 G	PWRFLG	003535 G	RPT1E	021341
JMPMSG	031036	L\$STA	002030 G	MS.XS0	000006 G	RAMDAT	003346 G	RPT1F	021217
JMPMS2	031236	L\$SW	002204 G	MS.XS1	000010 G	RAMDUM	014050 G	RPT1G	021270
JMP.C	000040 G	L\$TEST	002114 G	MS.XS2	000012 G	RAMER	015736 G	RPT1I	021460
J\$JMP	000167	L\$TIML	002014 G	MS.XS3	000014 G	RAMFHR	005306	RPT1J	021371
LENMSK	003436 G	L\$UNIT	002012 G	NCMD.C	0177740 G	RAMHLD	003342	RPT1K	021451
LOE	040000 G	L10000	002202	NCNT	003420 G	RAMIOP	005365	RRANV	002205 G
LOG	015436 G	L10001	002312	NCNT1	003422 G	RAMLIN	005450	RRBC	002620 G
LOOPCT	003525 G	L10002	005624	NEXTSP	030070	RAMM	031040	RRECL	000020 G
LOT	000010 G	L10003	006550	NEXTU	017346 G	RAMPD	005436	RRREC	002740 G
L\$ACP	002110 G	L10004	006556	NINUSE	017774 G	RAMRSH	003344	RRUNR	002750 G
L\$APT	002036 G	L10005	006564	NOINTM	004503 G	RAMSIZ	003406 G	RRVM	030617
L\$AU	023710 G	L10006	006572	NRDYM	023544	RAMWRT	002214 G	RS1	001233 G
L\$AUT	002070 G	L10007	006600	NSSRM	004353 G	RANB	003440 G	RS2	007622 G
L\$AUTO	023160 G	L10010	021654	NUMBM	031206	RANBC	0153624 G	RS3	000000 G
L\$CCP	002106 G	L10012	023156	NURTY1	005155 G	RANCMD	025652	RTLE	014212 G

PARAMETER CODING
SYMBOL TABLE

MACRO M1200 21-MAR-84 16:45 PAGE 118-3

SEQ 0180

RTLRTN 014346	TSBA = 002456 G	T\$AUT= 010013	WRTYER 003470 G	\$ISK0 = 000001
RWCPK 002334 G	TSC.FC= 177717 G	T\$CLE= 010014	WRTYFG 003467 G	\$ISK1 = 000001
RWD = 102010 G	TSC.TC= 177761 G	T\$DAT= 010045	WRUNR 002730 G	\$ISK2 = 000001
RWERR 003473 G	TSDB 002456 G	T\$DU = 010015	WSSR 011672 G	\$ISK3 = 000001
RSSAVE 003460 G	TSSR 002466 G	T\$HAR= 010041	WTM = 100011 G	\$ISK4 = 000001
SCCNT 003270 G	TSSREG 003462 G	T\$HW = 010000	WTR = 101011 G	\$ISK5 = 000001
SCERM 004312 G	TSVCT 002476 G	T\$INI= 010012	WTV = 104105 G	\$ISK6 = 000001
SCH = 140004 G	TS.A16= 000400 G	T\$MSG= 010003	WTVERM 004246 G	\$LOCTA= 177777
SCHBK 002446 G	TS.A17= 001000 G	T\$PC = 000001	WTYBRF 014732	\$LSTIN= 000001
SCHCNT= 000010 G	TS.NBA= 002000 G	T\$PRO= 010011	WTYCMD 014726	\$LSTTA= 000001
SEQEND 003632 G	TS.NXM= 004000 G	T\$PTA= 010044	WTYWRD 014730	\$LVTAG= 050416
SETCH 006700 G	TS.OFL= 000100 G	T\$RPT= 010010	X\$ALWA= 000000	\$NESTL= 177777
SETRW 006724 G	TS.RMR= 010000 G	T\$SOF= 010042	X\$FALS= 000040	\$NSK0 = 000120
SETUP 007706 G	TS.SC = 100000 G	T\$SRV= 010007	X\$OFFS= 000400	\$NSK1 = 000120
SFF = 105010 G	TS.SPE= 020000 G	T\$SUB= 010033	X\$TRUE= 000020	\$NSK2 = 000110
SFP TBL 002204 G	TS.SSR= 000200 G	T\$SW = 010001	X0.BOT= 000002 G	\$NSK3 = 000110
SFR = 105410 G	TS.UPE= 040000 G	T\$TES= 010040	X0.EOT= 000001 G	\$NSK4 = 000110
SRF = 104010 G	TS4ADR 027774	T1 024004 G	X0.LET= 020000 G	\$NSK5 = 000110
SRR = 104410 G	TS4CL 002526 G	T1SWB 003530 G	X0.ONL= 000100 G	\$NSK6 = 000110
STAERM 005626 G	TS4INT 002516 G	T1.1 024014	X0.RLL= 010000 G	\$SAVE = 000001
STAER1 006140	TS4INO 006552 G	T1.10 024344	X0.RLS= 040000 G	\$SAVLE= 177777
STAER2 006316	TS4IN1 006560 G	T1.11 024364	X0.TMK= 100000 G	\$SELLE= 177777
STAER3 006375	TS4IN2 006566 G	T1.12 024450	X1.EWN= 000010 G	\$SSK0 = 050473
STAER4 006433	TS4IN3 006574 G	T1.2 024140	X2.OPM= 100000 G	\$TAGLE= 177777
STAER5 006453	TS4VCT 030011	T1.3 024164	X3.DCK= 000010 G	\$TAGNU= 050511
STAER6 006262	T\$ARGC= 000001	T1.4 024204	X3.RNY= 157400 G	\$TEMP = 000402
STAER7 006232	T\$CODE= 023004	T1.5 024224	ZROPAT 010412	\$TSK0 = 050472
STAFLG 003534 G	T\$ERRN= 000001	T1.6 024244	\$BGNLE= 000002	\$TSK1 = 050473
STREAM 003532 G	T\$EXCP= 000000	T1.7 024264	\$BRJMP= 177777	\$TSK2 = 050474
SVCGBL= 000000	T\$FLAG= 000041	T1.8 024304	\$BSK0 = 050234	\$TSK3 = 050475
SVCINS= 000001	T\$FREE= 031520	T1.9 024324	\$BSK1 = 050236	\$TSK4 = 050510
SVCSUB= 000000	T\$GMAN= 000000	T2 025372 G	\$BSK2 = 050416	\$TSK5 = 050507
SVCTAG= 000000	T\$HILI= 000010	T3 026046 G	\$ERFLG= 000400	\$TSK6 = 050503
SVCTST= 000000	T\$LAST= 000001	T4 026524 G	\$F\$AND= 000310	\$TSK7 = 050505
SWBFLG 003524 G	T\$LOLI= 000000	T5 026670 G	\$F\$BAD= 000401	\$U = 000403
SWB.C = 010000 G	T\$LSYM= 010000	TSWEOT 027640	\$F\$BLA= 000170	\$ARGC= 000000
SYMD = 000007	T\$LTNO= 000006	T6 027002 G	\$F\$CAS= 000150	\$BYTE= 000403
SYMS = 000007	T\$NEST= 177777	UAM = 000200 G	\$F\$DEC= 000220	\$CASE= 000000
S\$LSYM= 010000	T\$NS0 = 000000	UNL = 100412 G	\$F\$DO = 000340	\$DST = 000000
TAPCAP 021571	T\$NS1 = 000005	UNREC 003474 G	\$F\$FAL= 000405	\$ELOC= 000402
TCCRA 012612	T\$NS2 = 000002	URERM 004654 G	\$F\$GOO= 000400	\$ERFL= 000000
TCC0 012740 G	T\$PCNT= 000000	VFEXC 016254 G	\$F\$IF = 000110	\$FLAG= 000001
TCC1 012762 G	T\$PTAB= 010044	VFISU 016540 G	\$F\$INC= 000210	\$FROM= 000000
TCC2 013000 G	T\$PTHV= 000001	VFYCNT 003300 G	\$F\$LOO= 000200	\$INH = 000403
TCC3 013146 G	T\$PTNU= 000001	VFYDAT 016142 G	\$F\$NAM= 000160	\$LOC = 027155
TCC4 013170 G	T\$SAVL= 177777	VFYFLG 003522 G	\$F\$NO = 000403	\$LOCN= 000000
TCC5 013670 G	T\$SEGL= 177777	VFY.C = 000100 G	\$F\$OR = 000320	\$REG = 177777
TCC6 013772 G	T\$SIZE= 000004	WLKZRO 010442	\$F\$RTI= 000350	\$RETU= 000000
TCC7 014026 G	T\$SUBN= 000000	WRBC 002560 G	\$F\$RTN= 000300	\$RTN1= 000000
TC2RTN 013144	T\$TAGL= 177777	WRECL = 000020 G	\$F\$SEL= 000140	\$RTN2= 000000
TIME1 003444 G	T\$TAGN= 010046	WRITEM 017776 G	\$F\$THE= 000330	\$SRC = 000000
TIME2 003446 G	T\$TEMP= 000000	WPR = 105005 G	\$F\$TRU= 000404	\$TGSV= 000000
TOERM 004271 G	T\$TEST= 000006	WRREC 002720 G	\$F\$UNT= 000130	\$TGS1= 000000
TOOMM 004542 G	T\$TSTM= 177777	WRT = 104005 G	\$F\$WHI= 000120	\$TGS2= 000000
TRAPD4 003536 G	T\$TSTS= 000001	WRTY 014350 G	\$F\$YES= 000402	\$TO = 000000
TRAP4 023574 G	T\$AU = 010016	WRTYCT 003250 G	\$IFLEV= 177777	\$TAG= 050000
TSAM 004520 G				

M14

PARAMETER CODING
SYMBOL TABLE

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

SEQ 0181

. ABS 031520 000
000000 001

ERRORS DETECTED: 0

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