

Table with multiple columns and rows of technical data, including various parameters and values.

100 1000 10000
100 1000 10000
100 1000 10000

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

.REM @

IDENTIFICATION

PRODUCT CODE:	AC - T771A - MC
PRODUCT NAME:	CZTKAAO TK50 FRONT END FUNCTION
PRODUCT DATE:	11 - APRIL - 1985
MAINTAINER:	TAPE OPTICAL DIAGNOSTIC ENGINEERING
AUTHOR:	WILLIAM KERR

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) 1985 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103

1 GENERAL INFORMATION

1.1 Product Description

The TK50 Functional Diagnostic is intended to provide confidence in the basic functionality of the TK50 subsystem. As such, this should be the first host level diagnostic run on the TK50 subsystem to verify installation, or for troubleshooting. Throughout the program, emphasis is placed on isolating faults to the Field Replaceable Unit (FRU).

The program runs in standalone mode in conjunction with the PDP-11 family Diagnostic Supervisor. In addition to host level testing, the program will implicitly invoke the TK50's controller resident Level 1 self-test microdiagnostics as well as explicitly invoking the controller's Level 2 microdiagnostics.

1.2 Product Users And Uses

1. DMT testing
2. As appropriate at various manufacturing facilities
3. Field service personnel
4. DEC customers who choose to provide their own maintainance

1.3 Performance Goals

This program will test up to four TK50's in a sequential manner. To run a full pass of the program, a scratch tape must be mounted in the transport and an operator must be present to perform manual intervention. However, appropriate subsets of the program can be run if there is no scratch tape, or the operator inhibits manual intervention tests. Furthermore, the first pass of the program will run in "quick verify" mode; i.e., a single iteration of each test will be performed. If multiple passes are specified by the operator, the second and all subsequent passes will run with each test executed with multiple iterations. First pass execution time will be approximately 20 minutes while second pass execution time will be approximately 24 minutes. These pass times are based on a single unit under test.

1.4 Pass/Fail Criteria

This program employs a bottom-up approach to testing the TK50; that is, Test 1 will attempt to verify the simplest level of host-to-controller communication as outlined in UQSSP. Each subsequent test builds upon the functionality already verified in previous tests. Hence, most errors encountered by the program will be considered as fatal device errors and the failing unit will be dropped from the rest of the test sequence.

104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157

1.5 Failsoft Goals

Unit specific problems will be handled by the program. CPU faults (i.e., illegal traps or interrupts) will be handled by the Diagnostic Supervisor. System faults will be handled by the Diagnostic Supervisor, fault dependent.

1.6 Restrictions

Although basic read/write testing is performed, this program is not interested in measuring the subsystem's data reliability. While recoverable data errors will be reported by the program, no attempt will be made to determine the subsystem's compliance with error rates. Unrecoverable data errors will be considered as fatal device errors, although the media could be the causative factor.

1.7 Non-Goals

This program is intended to verify the gross functionality of host-to-controller communications, the integrity of the controller hardware, controller-to-drive communication and the basic functionality of the drive. It is not intended as a verification of TMSCP protocol as implemented in the controller firmware, and no testing of TMSCP commands is provided.

1.8 Runtime Environment Requirements

Runtime environment requirements include:

1. XXDP+ Diagnostic Supervisor
2. PDP-11 Q-bus family CPU
3. 28 KW memory
4. Console Terminal
5. Load Device
6. 1 to 4 TK50 tape drives with controllers
7. 1 to 4 TK50 scratch cartridges (optional)
8. LCP-5 UFD software (optional)

159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208

2 USER INTERFACE

2.1 User Dialogue

The following user dialogue will be provided at program start-time to allow the user to establish certain operational parameters of the program.

2.1.1 Hardware Questions -

This set of questions must be answered when the program is first started.

CHANGE HARDWARE (L)? no default

NUMBER OF UNITS (D)? enter number from 1-4

UNIT x

BASE ADDRESS (O) 774500?

VECTOR (O) 224?

UNIT NUMBER (O)?

2.1.2 Definition Of Hardware Questions -

CHANGE HARDWARE - This question merely wants to know if you want to reconfigure the units under test. It must be answered "yes" on the first pass of the program.

NUMBER OF UNITS - Enter the number of TK50's to be tested.

BASE ADDRESS - Enter the IO address of the unit to be tested.

VECTOR - Enter the vector location to be used for the unit.

UNIT NUMBER - Enter the MSCP-specified unit number for the unit.

This entire set of questions will be repeated up to four times, depending on the user's response to the "number of units" question.

2.1.3 Software Questions -

Most of the optional functionality of the program is either handled automatically by the program or through established procedures provided by the Diagnostic Supervisor hence there are no software questions.

210
211
212
213
214
215
216
217
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

3 ERROR REPORTS

Error reports will have two basic formats as described below. It is anticipated that, due to program partitioning, it will be possible to unambiguously define a single FRU as the cause of any error condition.

3.1 Error Format 1

This basic format will be used by all host level testing.

```
CZTKA DVC FTL ERR eee ON UNIT ll TST tt SUB ss PC: xxxxxx
SA REG CONTENTS INCORRECT
IN INIT SEQUENCE STEP #: n
SA EXPECTED: yyyyyy SA RECEIVED: zzzzzz
```

*****FAILING FRU: CNTL

In this example, the fields have the following meanings:

- eee = discrete error number as defined by program
- ll = logical unit number assigned to unit-in-error during hardware questions
- tt = test number during which error occurred
- ss = subtest number
- xxxxxx = program location of error call
- n = step number of the UQSSP initialization sequence which detected the error condition
- yyyyyy = expected contents of SA register for this step
- zzzzzz = actual SA register contents

3.2 Error Format 2

This format will be used for errors detected by the Level 2 microdiagnostics.

The DUP "Receive Data" command is used to monitor the status of the Level 2 diagnostic and to collect results. It is valid to issue a "Receive Data" command at any time after the Level 2s have been started.

```
CZTKA DEV FTL ERR eee ON UNIT ll TST 09 SUB ss PC: xxxxxx
DRIVE ERROR
```

267
268
269
270
271
272
273
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

**** LEVEL 2 MICRODIAGNOSTIC DUMP ****

Program Status *****
Drive Status ***
Test Number ***
Track Number ***
TMSCP Command ***
TMSCP Response Flag ***
TMSCP Unit Flags *****

Error Log (Tape Transfer Error)
Error Format ***
Error Flags ***
Event Code *****
Error Retry Level ***
Number of Reties ***
Position (Low Order) *****
Position (High Order) *****
Controller Status ***
Drive Error Code ***
Drive Flags ***
Track number ***
Physical Block Number *****
Logical Block Number ***
Tape Count 0 ***
Tape Count 1 ***
Tape Count 2 ***
Drive State *****
Read/Write State *****
Operation Flags *****

Blocks Written Channel 1 *****
Blocks Written Channel 2 *****
Blocks Read Channel 1 *****
Blocks Read Channel 2 *****
Soft Write Channel 1 *****
Soft Write Channel 2 *****
Ecc Corrected Channel 1 *****
Ecc Corrected Channel 2 *****
Read Repositions Channel 1 *****
Read Repositions Channel 2 *****

3.2.1 Program Status -

The program status word is only valid when read with a message number of 1. Upon an error, it will contain the status field of the TMSCP command which produced the error. Note that there are several diagnostic error codes which will be returned with a "ST_DIA" diagnostic status.

If the test completed without a fatal or hard error, this word will be zero (Normal Successful Completion).

324
325
326
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
379
380

3.2.2 Drive Status -

The drive status word is only updated during the level 3 idle loop. Therefore, this word is only valid with a message number of 2. This word will contain the status returned for the TMSCP "Get Unit Status" command. The unit flags for the "Get Unit Status" command will be in the Unit Flag Field.

3.2.3 Test Number -

This byte will contain the number of the test during which an error occurred (only valid when message number is 1 and program status is non-zero). This is only used for the Level 2 diagnostic.

3.2.4 Track Number -

This byte will contain the track number of track most recently written/read. Valid only for message number 1.

3.2.5 TMSCP Command -

This byte will contain the opcode of the command in error (only valid when message number is 1 and program status is non-zero). TMSCP opcodes used include WRITE, READ, REPOSITION, ONLINE, AVAILABLE, GET UNIT STATUS, and SET CONTROLLER CHARACTERISTICS.

3.2.6 TMSCP Response Flags -

This byte will contain the flag field of the TMSCP response packet of the command in error. It will only be valid when the command in error field is valid.

3.2.7 TMSCP Unit Flags -

This word will contain the unit flags for the "Get Unit Status" command issued during the Level 2 Idle Loop. From this word, the host program will be able to tell if the drive is write protected. Unit Flags are only valid when Drive Status is valid.

3.2.8 Error Format -

This byte will contain the format code for any hard error for which an error log was received. If no error log was received, this byte will contain 255 decimal (OFF Hex, 377 Octal, 11111111 Binary).

381 In this case, all information which follows may be ignored (i.e.
382 it's undefined). This is also only valid for message 1.
383
384
385
386 3.2.9 Error Flags -
387
388 This byte will contain the flags from the error log. It is
389 only valid when the error log format code is not equal to 255(10).
390
391
392
393 3.2.10 Event Code -
394
395 This word will contain the event code of the error log. For
396 more detail on this and other fields of the error log, please refer to
397 "TK50 Magnetic Tape Sub-system Functions", 11 January 1985, Ric
398 Perron.
399
400
401
402 3.2.11 Error Retry Level -
403
404 This byte will contain the level field for a Tape Transfer
405 error. Only valid when Error Log Format Code is equal to Tape
406 Transfer error.
407
408
409
410 3.2.12 Number Of Retries -
411
412 This byte will contain the retry field for a Tape Transfer
413 error. Only valid when Error Log Format Code is equal to Tape
414 Transfer error.
415
416
417
418 3.2.13 Position -
419
420 This double word will contain the tape position for a Tape
421 Transfer error. Only valid when Error Log Format Code is equal to
422 Tape Transfer error.
423
424
425
426 3.2.14 Controller Status -
427
428 This byte will contain the controller status code and is only
429 valid when Error Log Format Code is equal to Tape Transfer error.
430
431
432
433 3.2.15 Drive Error Code -
434
435 This byte will contain the drive error code as reported for a
436 Drive error. Only valid when Error Log Format Code is equal to Tape
437 Transfer error.

438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494

3.2.16 Drive Error Code -

This byte will contain the drive error code as reported for a Drive error. Only valid when Error Log Format Code is equal to Tape Transfer error.

3.2.17 Drive Flags -

This byte will contain the drive flags. Only valid when Error Log Format Code is equal to Tape Transfer error.

3.2.18 Track Number -

This byte will contain the track number as reported for a Tape Transfer error. Only valid when Error Log Format Code is equal to Tape Transfer error.

3.2.19 Physical Block Number -

This word will contain the physical block number for a Tape Transfer error. Only valid when Error Log Format Code is equal to Tape Transfer error.

3.2.20 Logical Block Number -

This byte will contain the logical block number for a Tape Transfer error. Only valid when Error Log Format Code is equal to Tape Transfer error.

3.2.21 Tape Counts 0, 1, And 2 -

These bytes will contain the tape counts as reported for a Drive error. Only valid when Error Log Format Code is equal to Tape Transfer error.

3.2.22 Drive State -

This word will contain the drive state as reported in the error log. Only valid when Error Log Format Code is equal to Tape Transfer error.

495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551

3.2.23 Read/Write State -

3.2.24 Operation Flags -

This word will contain the Operation Flags as reported in the error log. Only valid when Error Log Format Code is equal to Tape Transfer error.

3.2.25 Controller Error Code -

This word will contain the controller error code as reported for a Controller error. Only valid when Error Log Format Code is equal to Controller error.

3.3 Get DUST Status

The DUP "Get DUST Status" command can be issued to verify that the Level 2 diagnostics are still running. If the progress count received for two consecutive "Get DUST Status" commands is identical, then progress has stopped. The Level 2 diagnostic should be aborted and restarted.

4 STOPPING THE LEVEL 2 DIAGNOSTIC

The Level 2 diagnostic will never terminate itself. If it is desired to stop the test in progress, the host must issue the DUP "Abort Program" command.

4.0.1 Blocks Written/Read -

These double words will contain the current number of block written/read on a per channel basis.

4.0.2 Soft Write Errors -

These words will contain a count of the number of soft write errors on a per channel basis.

4.0.3 ECC Corrected Error -

These words will contain a count of the number of CRC mismatches on a data block during a read operation which were corrected by an ECC group. This count is maintained on a per channel basis.

552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608

4.0.4 Read W/Reposition -

These words will contain a count of the number of CRC mismatches on data blocks during a read operations which could only be corrected by repositioning. That is, the ECC group could not correct the read error for the data block without repositioning.

4.0.5 Data Compare Error Information -

This buffer area will contain the following information: byte number in error, expected byte, and actual byte read. The first 10 bytes will be the byte value expected. The next 10 bytes will be the byte value read (actual). The final 10 words will contain the byte offset into the block in error (0 to 511).

4.0.6 Number Of Data Compare Bytes -

This word will contain the number of bytes which mismatched in the block.

4.1 Diagnostic Error Codes

In most cases, the Program Status Word will contain the error code returned by the controller firmware for the failing command. In the case where an error is detected by the diagnostic and not by the firmware, the Program Status Word will contain a diagnostic status code.

The diagnostic status code is indicated by the "1F" hex status (ST.DIA : internal diagnostic status code - see MSCP specification). The subcodes will be as follows:

4.2 Data Compare Error

This indicates that the compare operation between the data pattern which was written and read failed. The data compare buffer area will contain up to 10 bytes of the failing record, including byte location within the block, expected and actual data read. In addition, a total count of the bytes in error is available.

4.3 Controller Error

This code will be set in response to the diagnostic internally receiving an error log indicating a controller error. The controller error code field will contain the error code received.

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

4.4 Bad Parameters Error

This error code indicates that the parameters passed to the Level 2 diagnostic were invalid.

4.5 Bad Pattern

This subcode indicates an internal diagnostic error and should never be seen.

4.6 Response Address Error

This subcode indicates an internal firmware or diagnostic problem and should never be seen.

4.7 Host Address Error

This subcode indicates an internal firmware problem and should never be seen.

4.8 Unknown Error Log

This subcode indicates an internal firmware problem and should never be seen.

642
643
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
694
695
696
697
698

5 FUNCTIONAL DESCRIPTION

The following test descriptions all have certain points in common. All errors specified below will cause the unit to be dropped from the test, unless specifically noted to the contrary. Furthermore, if the operator has chosen loop-on-error (LOE flag set) scope loops will return to the beginning of the test containing the failure. Exceptions to this will also be noted explicitly below. To understand the normal four step initialization sequence, refer to the UQSSP; the descriptions of tests that use this sequence will only highlight unique features utilized by that specific test.

5.1 Test 1 - Existence Verification Test

This test verifies the existence of the IP and SA registers by simply addressing them. Failure here could be caused by incorrect address setting in the controller DIP switch, faulty controller logic or operator error in specifying base address.

5.2 Test 2 - SA Register Wrap Test

This test commences the UQ-port initialization sequence, but sets the "WR" bit in its Step 1 response to the controller. Upon seeing this bit set, the controller should immediately enter the Diagnostic Wrap Mode. The program will now write and read a floating 1 pattern to and from the SA register. The process will then be repeated with a floating 0 pattern.

A failure to echo the written data will result in a callout to the controller. If loop on error is set, the program will loop on the failing write and read.

5.3 Test 3 - Initialization Test And Power Up Microdiagnostics

This test commences the UQ-port initialization sequence with interrupts disabled. It will verify that all step transitions occur within the allotted time, and that all host supplied information is correctly echoed by the controller. The program further verifies that no interrupts occur as a result of the step transitions.

NOTE

The diagnostic will verify that interrupts do not occur at the vector assigned to the Unit Under Test (UUT). Inappropriate interrupts to an unassigned vector will be trapped by the Diagnostic Supervisor and will abort the program.

It should be noted that, in accordance with the UQSSP, the controller's power-up microdiagnostics will be executed during Step 1

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

of the sequence. Controller specific failure codes will be posted in the SA register by the self-tests; these could be correlated to a failing test number as an aid in troubleshooting. All failures in this test, however, will indicate the controller as the failing FRU. The specific codes will be 10 consecutive numbers in the range of 600-699 (decimal); exact values have not been designated.

5.4 Test 4 - Vector And Interrupt Test

Test 3 is repeated, this time with interrupts enabled. The checks of test 3 will also be performed, but now the program will also verify that an interrupt occurs at each step transition. No interrupt should occur at the end of step 4. Failures in this test will be attributed to the controller.

5.5 Test 5 - BR Level Test

This test insures that the TK50 cannot interrupt when the CPU priority is set to 7. The test goes through only the first step of the init sequence, since the controller will "hang" waiting for the interrupt acknowledge. Failures in this test will be attributed to the controller.

5.6 Test 6 - Purge And Poll Test

The initialization sequence will again be started by the host, but in step 3, the host will set the "PP" (Purge and Poll) bit. The first part of this test will simulate the UQSSP handshake for a bus adapter purge. The host will then request the controller to commence "poll" testing. The controller will now begin DMA activity in both directions to the host-identified communication area. The controller must end this test by leaving the communication area cleared and by transitioning to step 4. The host will verify that the entire

communication area is cleared. (Note: At the start of this test the host will have filled the communication area with a non-zero data pattern.) Ring depth in this test will be set to the minimal value(i.e., one), reducing potential impact on host memory in event of failure. Failures encountered will again be attributed to the controller.

5.7 Test 7 - Maximum Ring Buffer Test

This test is similar to test 5, but it will utilize the maximum allowable ring depth as specified in UQSSP. This value is equal to 128 command and 128 response slots of 32 bits per slot.

756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812

5.8 Test 8 - Extended Address Test

The format of this test is similar to test 6, but the program will establish the communication area in the highest available memory locations. This will allow testing of the upper six bits of address logic on the controller board.

NOTE

The ability of this test to comprehensively test the upper six bits of address is dependent upon memory size of the run-time system. This test will be completely bypassed if the system has only 28K words of memory.

5.9 Test 9 - Get DUST Status

The Get DUST Status test will request and test the DUST status of each unit under test for two specific cases:

1. No command modifiers set
2. Illegal command modifiers set

DUST status will be received from the unit under test after the program issues the Get DUST Status command available in DUP. The response packet received from the unit will be tested against a known good mask. If the expected and received bits do not match, excluding variable bits which will be discounted, an error will be reported.

5.10 Test 10 - Execute Level 2 Microdiagnostics

This test will invoke, via the DUP Execute Local Program command, the controller resident Level 2 microdiagnostics. These tests are aimed at establishing full communication between the controller and drive, as well as performing functional testing of the drive. Refer to the Level 2 Microdiagnostic Functional Specification for a detailed description of these tests. Test progress will be monitored from the host with the Get DUST Status command. Most errors encountered in this test will be attributed to the drive.

While the Level 2 Microdiagnostics are executing, TMSCP commands will be rejected by the controller. The microdiagnostics, in conjunction with those portions of operational microcode utilized by them, will have responsibility for detecting and dealing with drive errors, including command timeouts and other communication failures of the drive. The host program will be responsible for monitoring overall execution of the microdiagnostics, via DUP commands, and detecting "hung" controller conditions, as well as any spurious interrupts generated by the subsystem.

It is not possible for the Level 2 microdiagnostics themselves to guarantee that a scratch medium is mounted. They are reliant upon

813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869

host level software to protect user data from accidental destruction.

The following sequence of tests will be performed twice, thus ensuring that both channels of the head receive equal testing. To reduce execution time, the entire sequence is performed first at the physical BOT end of tape, utilizing head channel 1. The tape will then be positioned to physical EOT end of tape and the entire sequence repeated, utilizing channel 2. Finally the tape is rewound and left at physical BOT.

5.11 Micro Test Descriptions

5.11.1 Test 1 - Simple Write/Read Test -

This test will write several blocks of 1F and 2F data to the tape. The tape will be rewound and the blocks read back. The test will perform a compare operation of the data read back. This test verifies basic write/read capability.

5.11.2 Test 2 - Streaming Write/Read Test -

This test will write approximately 20 feet of tape on a single track with blocks of various data patterns. The tape will be kept streaming throughout the write. The tape will then be rewound and all records are read, again while streaming. Data compare is performed on all blocks. This test is intended to verify that the drive can write and read in a normal operational.

5.11.3 Test 3 - Thrashing Write/Read Test -

This test will again write 40 feet of tape on a single track; in this test delays will be inserted between the write blocks to force tape thrashing. After rewinding, the blocks will be read with forced delays between reads. This test is intended to verify that tape tracking and tensioning are not affected by frequent direction changes.

5.11.4 Test 4 - Peak Shift Test -

Test 2 is repeated, this time using the worst case MFM peak shift patterns. This test is intended to verify that the write precompensation logic is functioning properly.

5.11.5 Test 5 - Signal Sag Test -

Ten blocks will be written with random data. The tape will be rewound, then moved back and forth 10 times. The tape will then be

870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920

positioned at logical BOT for the recorded track and read back. This test is intended to detect self-erasure of recorded data.

5.11.6 Test 6 - Overwrite Test -

One hundred blocks of data will be written on a single track, using an "MW" data pattern. (A binary representation of the MW pattern is 1110111, although the actual number of ones appearing in a group is subject to further worst case testing by the Drive Engineering group.) The tape will then be rewound and rewritten. (The data pattern used for rewrite consists of 50 blocks of worst-case MFM followed by 50 blocks of random data.) The purpose of the test is to guarantee that the drive is capable of overwriting previously recorded data.

5.11.7 Test 7 - Track Access Test -

Fifty blocks of data will be written on a single track. The tape will then be rewound, the head stepped to the next track and the same number of blocks will be written. This process will be repeated until all the tracks for that head have been written. The test will then reread the blocks from all data. This test is intended to verify tape tracking and the ability to successfully record and retrieve data on adjacent tracks.

5.11.8 Test 8 - Positioning Test -

The primary purpose of this test is simply to perform the correct positioning of the tape for what is to follow; that is, the first time through the test sequence, this test will cause the tape to be positioned at the physical EOT end of tape, thus allowing the entire test cycle to be repeated for testing the drive's operation with channel 2 of the head. Upon second entry to this test, it will cause the tape to be positioned back at physical BOT. The test will include a coarse watch-dog timer to guard against a "hung" drive condition.

5.12 Retry Algorithms

The Level 2 microdiagnostics will make extensive use of portions of the controller's operational microcode, including retry algorithms intended to recover from data errors. Please refer to the TK50 Microcode Functional Specification for detailed descriptions of the retry algorithms.

@

```

932      .TITLE PROGRAM HEADER AND TABLES
933      .SBTTL PROGRAM HEADER
959
961 000000      .ENABL ABS,AMA
962      002000      = 2000
963      .NLIST BEX
965
966 002000      BGNMOD
967
968      ;**
969      ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
970      ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
971      ;--
972
973 002000      POINTER BGNDU,ERRTBL,BGNRPT
974
982
983 002000      HEADER CZTKA,A,0,120.,0,PRI00
      L$NAME::      ;DIAGNOSTIC NAME
      .ASCII /C/
      .ASCII /Z/
      .ASCII /T/
      .ASCII /K/
      .ASCII /A/
      .BYTE 0
      .BYTE 0
      .BYTE 0
      L$REV::      ;REVISION LEVEL
      .ASCII /A/
      L$DEPO::      ;0
      .ASCII /0/
      L$UNIT::      ;NUMBER OF UNITS
      .WORD 0
      L$TIML::      ;LONGEST TEST TIME
      .WORD 120.
      L$HPCP::      ;POINTER TO H.W. QUES.
      .WORD L$HARD
      L$SPCP::      ;POINTER TO S.W. QUES.
      .WORD 0
      L$HPTP::      ;PTR. TO DEF. H.W. PTABLE
      .WORD L$HW
      L$SPTP::      ;PTR. TO S.W. PTABLE
      .WORD 0
      L$LADP::      ;DIAG. END ADDRESS
      .WORD L$LAST
      L$STA::      ;RESERVED FOR APT STATS
      .WORD 0
      L$CO::      .WORD 0
      L$DTYP::      ;DIAGNOSTIC TYPE
      .WORD 0
      $APT::      ;APT EXPANSION
      .WORD 0
      L$DTP::      ;PTR. TO DISPATCH TABLE
      .WORD L$DISPATCH
      L$PRIO::      ;DIAGNOSTIC RUN PRIORITY
      .WORD PRI00

```

002044		L\$ENVI::		;FLAGS DESCRIBE HOW IT WAS SETUP
002044	000000		.WORD 0	
002046		L\$EXP1::		;EXPANSION WORD
002046	000000		.WORD 0	
002050		L\$MREV::		;SVC REV AND EDIT #
002050	004		.BYTE C\$REVISION	
002051	000		.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	023412		.WORD L\$DVTYP	
002062		L\$REPP::		;PTR. TO REPORT CODE
002062	035002		.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	000000		.WORD 0	
002072		L\$DUT::		;PTR. TO DROP UNIT CODE
002072	041104		.WORD L\$DU	
002074		L\$LUN::		;LUN FOR EXERCISERS TO FILL
002074	000000		.WORD 0	
002076		L\$DESP::		;POINTER TO DIAG. DESCRIPTION
002076	002150		.WORD L\$DESC	
002100		L\$LOAD::		;GENERATE SPECIAL AUTOLOAD EMT
002100	104035		EMT E\$LOAD	
002102		L\$ETP::		;POINTER TO ERR_TBL
002102	023052		.WORD L\$ERR_TBL	
002104		L\$ICP::		;PTR. TO INIT CODE
002104	040574		.WORD L\$INIT	
002106		L\$CCP::		;PTR. TO CLEAN-UP CODE
002106	041054		.WORD L\$CLEAN	
002110		L\$ACP::		;PTR. TO AUTO CODE
002110	000000G		.WORD L\$AUTO	
002112		L\$PRT::		;PTR. TO PROTECT TABLE
002112	023044		.WORD L\$PROT	
002114		L\$TEST::		;TEST NUMBER
002114	000000		.WORD 0	
002116		L\$DLY::		;DELAY COUNT
002116	000000		.WORD 0	
002120		L\$HIME::		;PTR. TO HIGH MEM
002120	000000		.WORD 0	

991
992
993
994
995
996
997
998

.SBTTL DISPATCH TABLE

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

DISPATCH 10.

002122
002122 000012
002124
002124 041126
002126 041624
002130 042406
002132 043060
002134 043640
002136 044302
002140 045106
002142 045712
002144 046622
002146 047032

.WORD 10
L\$DISPATCH: :
.WORD T1
.WORD T2
.WORD T3
.WORD T4
.WORD T5
.WORD T6
.WORD T7
.WORD T8
.WORD T9
.WORD T10

999
1006
1007
002150
002150
002150

103 132 124

DESCRIPT <CZTKAA0 TK50 FUNCTIONAL >
L\$DESC: :
.ASCIZ /CZTKAA0 TK50 FUNCTIONAL/
.EVEN

1008

```

1010          .SBTTL  DEFAULT HARDWARE P-TABLE
1011
1012          ;++
1013          ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
1014          ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
1015          ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES.
1016          ;--
1017
1018 002200          BGNHW  DFPTBL
          002200 000003  .WORD  L10000-L$HW/2
          002202          L$HW::
          002202          DFPTBL::

1019
1025 002202 174500          .WORD  174500          ;TKIP BASE ADDRESS
1026 002204 000260          .WORD  260           ;VECTOR
1027 002206 000000          .WORD  0           ;T/MSCP UNIT NUMBER
1028 002210          ENDDW
          002210          L10000:

```

```
1031 .SBTTL SOFTWARE P-TABLE
1032
1033 ;++
1034 ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
1035 ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
1036 ;--
1037
1038 002210          BGNSW  SFPTBL
      002210 000000  .WORD  L10001-L$SW/2
      002212
      002212 L$SW::
              SFPTBL::
1039
1046
1047 002212          ENDSW
      002212 L10001:
1048
1049 002212          ENDMOD
```

1052
1063
1064
1092
1093
1094 002212
1095
1096
1097
1098
1099
1100
1101 002212

.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.	; BIT POSITION IN SECOND STATUS WORD
000037	EF.RESTART==	31.	; (100000) START COMMAND WAS ISSUED
000036	EF.CONTINUE==	30.	; (040000) RESTART COMMAND WAS ISSUED
000035	EF.NEW==	29.	; (020000) CONTINUE COMMAND WAS ISSUED
000034	EF.PWR==	28.	; (010000) A NEW PASS HAS BEEN STARTED
			; (004000) A POWER-FAIL/POWER UP OCCURRED

; PRIORITY LEVEL DEFINITIONS

000340	PRI07==	340
000300	PRI06==	300
000240	PRI05==	240
000200	PRI04==	200
000140	PRI03==	140
000100	PRI02==	100
000040	PRI01==	40
000000	PRI00==	0
	:	
	; OPERATOR FLAG BITS	
	:	
000004	EVL==	4
000010	LOT==	10
000020	ADR==	20
000040	IDU==	40
000100	ISR==	100
000200	UAM==	200
000400	BOE==	400
001000	PNT==	1000
002000	PRI==	2000
004000	IXE==	4000
010000	IBE==	10000
020000	IER==	20000
040000	LOE==	40000
100000	HOE==	100000

```

1106      ;:*****
1107      ;:*****
1108      ;
1109      ;LUN_BLOCK OFFSETS
1110      ;   THESE LITERALS ARE USED AS WORD OFFSETS INTO THE LUNBLK, WHICH
1111      ;   IS POINTED TO THROUGHOUT THE PROGRAM BY R4.
1112      ;
1113      ;:*****
1114      ;:*****
1115
1116      000000      TKIP      ==      0      ;TKIP REGISTER ADDRESS
1117      000002      TKSA      ==      2      ;TKSA REGISTER ADDRESS
1118      000004      TKVEC     ==      4      ;TK INTERRUPT VECTOR
1119      000006      MSCPUN    ==      6      ;T/MSCP UNIT NUMBER
1120      000010      TKIPSV    ==     10      ;SAVE LOCATION FOR IP CONTENTS
1121      000012      TKSASV    ==     12      ;SAVE LOCATION FOR SA CONTENTS
1122      000014      LUNFLG    ==     14      ;BIT-SPECIFIC MEANINGS AS DEFINED BELOW
1123
1124
1125      ;:*****
1126      ;:*****
1127      ;
1128      ;LUNFLG
1129      ;   THIS WORD IN LUNBLK IS USED TO CONVEY VARIOUS INFORMATION
1130      ;   IN A BIT-SPECIFIC MANNER. BITS USED BY THE PROGRAM ARE
1131      ;   DEFINED AS FOLLOWS.
1132      ;
1133      ;:*****
1134      ;:*****
1135
1136      000001      DRPFLG    ==      BIT0      ;=0 UUT AVAILABLE FOR TEST
1137      ;=1 UUT HAS BEEN DROPPED
1138      000002      INTFLG    ==      BIT1      ;=1 EXPECTED INTERRUPT OCCURRED
1139
1140      000004      BRFLAG     ==      BIT2      ;=1 INTERRUPT PRIORITY TEST
1141
1142      000010      DUPFLG     ==      BIT3      ;=1 DUP COMMAND
1143
1144      000020      ABTFLG     ==      BIT4      ;=1 ABORT LOCAL PROGRAM
1145
1146      000040      CNTRLC     ==      BIT5      ;=1 RETURN TO DRS
1147

```

```

1149 ;*****
1150 ;*****
1151 ;
1152 ;UQ-PORT EQUATES
1153 ; THIS SECTION DEFINES THOSE LITERALS USED
1154 ; BY THE DIAGNOSTIC IN THE UQ-PORT PROTOCOL.
1155 ; IN GENERAL THEY HAVE BEEN FORMED BY USING
1156 ; THE TWO LETTER MNEMONIC DEFINED IN UQSSP,
1157 ; PRECEDED BY "B." INDICATING THEY ARE BITS.
1158 ;
1159 ;*****
1160 ;*****
1161 ;
1162 ;READ-ONLY BITS
1163
1164 004000 B.S1 == BIT11 ;STEP 1
1165 010000 B.S2 == BIT12 ;STEP 2
1166 020000 B.S3 == BIT13 ;STEP 3
1167 040000 B.S4 == BIT14 ;STEP 4
1168
1169 100000 B.ER == BIT15 ;ERROR INDICATION
1170 002000 B.NV == BIT10 ;=0 VECTOR IS HOST SETTABLE
1171 001000 B.QB == BIT9 ;=1 SUPPORTS 22 BIT HOST BUS
1172 000400 B.DI == BIT8 ;=1 SUPPORTS ENHANCED DIAGNOSTICS
1173 000200 B.OD == BIT7 ;=1 SUPPORTS ODD BUFFER ADDRESSES
1174 000100 B.PP == BIT6 ;=1 SUPPORTS ADDRESS MAPPING
1175
1176 ;WRITE-ONLY BITS
1177
1178 100000 B.PP == BIT15 ;PERFORM PURGE AND POLL TESTS
1179 040000 B.WR == BIT14 ;ENTER DIAGNOSTIC WRAP MODE
1180 000002 B.LF == BIT1 ;LAST FAIL REQUEST
1181 000001 B.PI == BIT0 ;ENABLE ADAPTER PURGE INTERRUPTS
1182 000001 B.GO == BIT0 ;GO BIT - START RUNNING
1183
1184 ;READ/WRITE BITS
1185
1186 000200 B.IE == BIT7 ;STEP X-TION INTERRUPT ENABLE
1187
    
```

```

1189      ;:*****
1190      ;:*****
1191      ;
1192      ;GENERAL PURPOSE EQUATES
1193      ;
1194      ;:*****
1195      ;:*****
1196
1197      000004      VEC4      ==      4      ;VECTOR FOUR - NXM TIMEOUTS, ETC.
1198      177560      RCSR      ==      177560      ;TERMINAL RECEIVE CONTROL/STATUS REGISTER ADDRESS
1199      177562      RBUF      ==      177562      ;TERMINAL RECEIVE BUFFER ADDRESS
1200

```

```

1202      ;:*****
1203      ;:*****
1204      ;
1205      ;MEMORY MANAGEMENT EQUATES
1206      ;
1207      ;:*****
1208      ;:*****
1209
1210      177572      MMUSRO ==      177572 ;STATUS REG 0
1211      177574      MMUSR1 ==      177574
1212      177576      MMUSR2 ==      177576
1213      172516      MMUSR3 ==      172516 ;SHOULD ONLY BE PRESENT ON 22 BIT CPU'S
1214
1215      172340      KPAR0  ==      172340 ;KERNEL MODE PAGE ADDRESS REG 0
1216      172342      KPAR1  ==      172342
1217      172344      KPAR2  ==      172344
1218      172346      KPAR3  ==      172346
1219      172350      KPAR4  ==      172350
1220      172352      KPAR5  ==      172352
1221      172354      KPAR6  ==      172354
1222      172356      KPAR7  ==      172356 ;ALWAYS FOR I/O PAGE
1223
1224      172300      KPDR0   ==      172300 ;KERNEL MODE PAGE DESCRIPTOR REG 0
1225      172302      KPDR1   ==      172302
1226      172304      KPDR2   ==      172304
1227      172306      KPDR3   ==      172306
1228      172310      KPDR4   ==      172310
1229      172312      KPDR5   ==      172312
1230      172314      KPDR6   ==      172314
1231      172316      KPDR7   ==      172316
1232
1233      000001      MMON    ==      BIT0   ;ENABLE MMU - MMUSRO
1234      000020      MM220N ==      BIT4   ;ENABLE 22 BIT MMU - MMUSR3
1235

```

```

1237 ;:*****
1238 ;:*****
1239 ;
1240 ;COMMAND PACKET OPCODES
1241 ;
1242 ;:*****
1243 ;:*****
1244
1245 000004 OP.SCC == 04 ;SET CONTROLLER CHARACTERISTICS OPCODE
1246 000011 OP.ONL == 11 ;ONLINE OPCODE
1247 000001 OP.GDS == 01 ;GET DUST STATUS OPCODE
1248 000003 OP.ELP == 03 ;EXECUTE LOCAL PROGRAM OPCODE
1249 000005 OP.REC == 05 ;RECEIVE DATA OPCODE
1250 000006 OP.ABT == 06 ;ABORT PROGRAM OPCODE
1251 000200 OP.END == 200 ;END MESSAGE FLAG OPCODE
1252
1253
1254 ;:*****
1255 ;:*****
1256 ;
1257 ;GENERIC COMMAND PACKET OFFSETS
1258 ;
1259 ;:*****
1260 ;:*****
1261
1262 000000 P.CRF == 0 ;COMMAND REFERENCE NUMBER
1263 000004 P.UNIT == 4 ;UNIT NUMBER
1264 000010 P.OPCD == 10 ;OPCODE
1265 000011 P.FLGS == 11 ;END MESSAGE FLAGS
1266 000012 P.MOD == 12 ;MODIFIERS / STATUS
1267 000012 P.STS == 12 ;MODIFIERS / STATUS
1268 000014 P.BCNT == 14 ;BYTE COUNT
1269 000020 P.BUFF == 20 ;BUFFER DESCRIPTOR
1270
1271
1272 ;:*****
1273 ;:*****
1274 ;
1275 ;GET DUST STATUS END PACKET OFFSETS
1276 ;
1277 ;:*****
1278 ;:*****
1279
1280 000014 P.EXT1 == 14 ;1ST BYTE OF PROGRAM EXTENSION
1281 000015 P.EXT2 == 15 ;2ND BYTE OF PROGRAM EXTENSION
1282 000016 P.EXT3 == 16 ;3RD BYTE OF PROGRAM EXTENSION
1283 000017 DUSTFL == 17 ;FLAGS
1284 000020 P.IND1 == 20 ;1ST WORD OF PROGRESS INDICATOR
1285 000022 P.IND2 == 22 ;2ND WORD OF PROGRESS INDICATOR
1286 000024 TIMEOUT == 24 ;TIMEOUT VALUE
1287

```

```

1289      ;*****
1290      ;*****
1291      ;
1292      ;TKSA BIT DEFINITIONS
1293      ;
1294      ;*****
1295      ;*****
1296
1297      100000      ERR      ==      100000      ;ERROR
1298      004000      S1      ==      004000      ;STEP 1
1299      000001      GO      ==      000001      ;GO
1300
1301
1302      ;*****
1303      ;*****
1304      ;
1305      ;U/Q PORT LITERALS
1306      ;
1307      ;*****
1308      ;*****
1309
1310      100000      OWN      ==      100000      ;DESCRIPTOR OWNERSHIP BIT
1311      040000      FLAG     ==      040000      ;DESCRIPTOR INTERRUPT FLAG BIT
1312      000200      IMM      ==      000200      ;IMMEDIATE COMMAND FLAG
1313      000010      TF.BLK   ==      10          ;TAPE FORMAT
1314      000000      HSTIMO   ==      0          ;HOST TIMEOUT VALUE
1315      000000      MSCPVR   ==      0          ;MSCP VERSION NUMBER
1316      000004      RNGSTP   ==      4.         ;DESCRIPTOR RING STEP
1317      000104      RSPSTP   ==      68.        ;RESPONCE BUFFER STEP
1318
1319

```

```

1321      ;*****
1322      ;*****
1323      ;
1324      ;TMSCP DRIVER BUFFER OFFSETS
1325      ;
1326      ;*****
1327      ;*****
1328
1329      000002      HIADDR ==      2.      ;DESCRIPTOR ADDRESS OFFSET
1330      177777      CONID  ==      -1.     ;COMMAND/RESPONSE CONNECTION TYPE I.D.
1331      177776      CRD   ==      -2.     ;COMMAND/RESPONSE CREDIT LIMIT OFFSET
1332      177774      MSGLEN ==      4.     ;COMMAND/RESPONSE MESSAGE LENGTH
1333      000005      TXFER ==      5.     ;ERROR FORMAT FOR "TAPE TRANSFER" ERROR LOG
1334      000011      DRIVER ==      9.     ;ERROR FORMAT FOR "DRIVE ERROR" ERROR LOG
1335      000000      CNTER ==      0.     ;ERROR FORMAT FOR "CONTROLLER ERROR" ERROR LOG
1336

```



```

1338 ;*****
1339 ;*****
1340 ;
1341 ;OFFSET VALUES INTO LEVEL 2 MICRODIAGNOSTIC RECEIVE DATA BUFFER
1342 ;
1343 ;*****
1344 ;*****
1345
1346 000002 L2STA == 2 ;OFFSET OF PROGRAM STATUS
1347 000004 L2DRV == 4 ;OFFSET OF DRIVE STATUS
1348 000006 L2TST == 6 ;OFFSET OF TEST NUMBER
1349 000007 L2TRK == 7 ;OFFSET OF TRACK NUMBER
1350 000010 L2CMD == 10 ;OFFSET OF COMMAND IN ERROR
1351 000011 L2RSP == 11 ;OFFSET OF RESPONSE FLAG
1352 000012 L2UNT == 12 ;OFFSET OF UNIT FLAGS
1353 000014 L2BWR1 == 14 ;OFFSET OF BLOCKS WRITTEN CH. 1
1354 000020 L2BWR2 == 20 ;OFFSET OF BLOCKS WRITTEN CH. 2
1355 000024 L2BRD1 == 24 ;OFFSET OF BLOCKS READ CH. 1
1356 000030 L2BRD2 == 30 ;OFFSET OF BLOCKS READ CH. 2
1357 000034 L2SWR1 == 34 ;OFFSET OF SOFT WRITE ERRORS CH. 1
1358 000036 L2SWR2 == 36 ;OFFSET OF SOFT WRITE ERRORS CH. 2
1359 000040 L2ECC1 == 40 ;OFFSET OF ECC CORRECTED ERRORS CH. 1
1360 000042 L2ECC2 == 42 ;OFFSET OF ECC CORRECTED ERRORS CH. 2
1361 000044 L2CRC1 == 44 ;OFFSET OF CRC ERRORS ON ECC BLOCK CH. 1
1362 000046 L2CRC2 == 46 ;OFFSET OF CRC ERRORS ON ECC BLOCK CH. 2
1363 000050 L2REP1 == 50 ;OFFSET OF READ REPOSITIONS CH. 1
1364 000052 L2REP2 == 52 ;OFFSET OF READ REPOSITIONS CH. 2
1365 000134 L2ELFM == 134 ;OFFSET OF ERROR LOG ERROR FORMAT
1366 000135 L2ELFL == 135 ;OFFSET OF ERROR LOG FLAGS
1367 000136 L2ELEV == 136 ;OFFSET OF ERROR LOG EVENT CODE
1368 000144 L2ELRL == 144 ;OFFSET OF ERROR LOG RETRY LEVEL
1369 000145 L2ELRT == 145 ;OFFSET OF ERROR LOG NUMBER OF RETRIES
1370 000146 L2ELP1 == 146 ;OFFSET OF ERROR LOG POSITION (low order)
1371 000150 L2ELP2 == 150 ;OFFSET OF ERROR LOG POSITION (high order)
1372 000154 L2ELST == 154 ;OFFSET OF CONTROLLER STATUS
1373 000155 L2ELDC == 155 ;OFFSET OF ERROR LOG DRIVE ERROR CODE
1374 000156 L2ELDF == 156 ;OFFSET OF ERROR LOG DRIVE FLAGS
1375 000157 L2ELTN == 157 ;OFFSET OF ERROR LOG TRACK NUMBER
1376 000160 L2ELPB == 160 ;OFFSET OF ERROR LOG PHYSICAL BLOCK NUMBER
1377 000162 L2ELLB == 162 ;OFFSET OF ERROR LOG LOGICAL BLOCK NUMBER
1378 000163 L2ELT0 == 163 ;OFFSET OF ERROR LOG LSB OF TAPE LENGTH COUNT
1379 000164 L2ELT1 == 164 ;OFFSET OF ERROR LOG BYTE 1 OF TAPE LENGTH COUNT
1380 000165 L2ELT2 == 165 ;OFFSET OF ERROR LOG MSB OF TAPE LENGTH COUNT
1381 000166 L2ELDS == 166 ;OFFSET OF ERROR LOG DRIVE STATE
1382 000170 L2ELRW == 170 ;OFFSET OF ERROR LOG READ/WRITE STATE
1383 000172 L2ELOF == 172 ;OFFSET OF ERROR LOG OPERATION FLAGS
1384 000140 L2ELEC == 140 ;OFFSET OF ERROR LOG CONTROLLER ERROR CODE

```

1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431

.SBTTL GLOBAL DATA SECTION

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

;LUNBLK
; THIS BLOCK OF MEMORY IS USED TO STORE VARIABLE INFORMATION
; PERTAINING TO THE CURRENT LOGICAL UNIT UNDER TEST. LUNBLK
; IS POINTED TO THROUGHOUT THE PROGRAM BY R4 AND INDIVIDUAL
; LOCATIONS ARE ACCESSED VIA LITERALS DEFINED ABOVE.

002212 000000 000000 000000 LUNBLK: .WORD 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

;UQ-PORT NECESSITIES
; THESE TABLES ARE SET UP BY VARIOUS
; TESTS WITH VALUES TO BE WRITTEN TO
; THE PORT, AND COMPARISON VALUES TO
; CHECK THE PORT AFTER EACH STEP TRAN-
; SITION OCCURS, RESPECTIVELY.

002250 STPTBL: .BLKW 4 ;VALUES WRITTEN TO THE PORT

002260 CMPTBL: .BLKW 4 ;COMPARISON VALUES

```

1433 ;*****
1434 ;*****
1435 ;
1436 ;PROGRAM CONTROL VARIABLES
1437 ; THESE GLOBAL VARIABLES ARE GENERALLY USED TO CONTROL THE
1438 ; OVERALL EXECUTION OF THE DIAGNOSTIC.
1439 ;
1440 ;*****
1441 ;*****
1442
1443 002270 000000 PASCNT:: .WORD 0 ;CUMULATIVE PROGRAM PASS COUNTER
1444 002272 000000 ITRCNT:: .WORD 0 ;LOADED BY EACH TEST TO SPECIFY # OF ITERATIONS
1445 002274 000000 KTFLAG:: .WORD 0 ;=0 MEMORY MANAGEMENT NOT AVAILABLE
1446 ;=1 MEMORY MANAGEMENT IS AVAILABLE
1447 002276 000000 TRP4FG:: .WORD 0 ;=1 TRAP TO VECTOR OCCURRED
1448 002300 000000 PAROFF:: .WORD 0 ;USED IN TEST 7 TO STEP THROUGH UPPER MEMORY
1449 002302 000000 CMMERR:: .WORD 0 ;=0 NO ERROR IN COMMUNICATION AREA
1450 ;=1 ERROR WITHIN COMMUNICATION AREA
1451 ;=-1 ERROR BEYOND BOUNDS OF COMM AREA
1452 002304 000000 CMTBLG:: .WORD 0 ;# OF CONTIGUOUS WORDS IN ERROR IN COMM AREA
1453 002306 000000 CMARLG:: .WORD 0 ;LENGTH OF COMM AREA FOR TEST N
1454 002310 000000 FRUIS:: .WORD 0 ;POINTER TO FAULTY FRU ASCII FOR PRINTOUT
1455 002312 000000 LOGUNT:: .WORD 0 ;LOGICAL UNIT # OF CURRENT UUT
1456 002314 000000 SAEXP:: .WORD 0 ;LOADED WITH EXPECTED SA FOR ERROR CHECKING
1457 002316 000000 INISTP:: .WORD 0 ;CURRENT STEP OF INIT SEQUENCE
1458 002320 000000 STEPST:: .WORD 0 ;SUCCESS/FAIL STATUS FROM STEP SUBROUTINES
1459 002322 000000 WRDATA:: .WORD 0 ;LOADED WITH DATA FRO WRAP MODE TEST
1460 002324 000000 INNER:: .WORD 0 ;COUNTER FOR PDELAY ROUTINE
1461 002326 000000 OUTER:: .WORD 0 ;OTHER COUNTER FOR PDELAY
1462 002330 000000 TOUT:: .WORD 0 ;TIMEOUT INDICATOR FOR PDELAY
1463 002332 000000 TEMP:: .WORD 0 ;TEMPORARY STORAGE LOCATION
1464 002334 000000 FLAGS:: .WORD 0 ;WORK LOCATION FOR SUPERVISOR FLAGS

```

```

1466 ;*****
1467 ;*****
1468 ;
1469 ;TMSCP/DUP COMMAND PACKETS
1470 ;
1471 ;*****
1472 ;*****
1473 ;
1474 ;*****
1475 ;
1476 ;TMSCP SET CONTROLLER CHARACTERISTICS COMMAND PACKET
1477 ;
1478 ;*****
1479 ;
1480 002336 000040 000000          SCTRLC: .WORD 32.,0
1481 002342 000000 000000          .WORD 0,0
1482 002346 000000 000000          .WORD 0,0
1483 002352 000004 000000          .WORD OP.SCC,0
1484 002356 000000 000000          .WORD 0,0
1485 002362 000000 000000          .WORD 0,0
1486 002366 000000 000000          .WORD 0,0
1487 002372 000000 000000          .WORD 0,0
1488 002376 000000 000000          .WORD 0,0
1489 ;
1490 ;
1491 ;STARS
1492 ;
1493 ;TMSCP ONLINE COMMAND PACKET
1494 ;
1495 ;*****
1496 ;
1497 002402 000044 000000          ONLINE: .WORD 36.,0
1498 002406 000000 000000          .WORD 0,0
1499 002412 000000 000000          .WORD 0,0
1500 002416 000011 000000          .WORD OP.ONL,0
1501 002422 000000 000000          .WORD 0,0
1502 002426 000000 000000          .WORD 0,0
1503 002432 000000 000000          .WORD 0,0
1504 002436 000000 000000          .WORD 0,0
1505 002442 000000 000000          .WORD 0,0
1506 002446 000000 000000          .WORD 0,0
1507 ;
1508 ;
1509 ;STARS
1510 ;
1511 ;DUP GET DUST STATUS COMMAND PACKET
1512 ;
1513 ;*****
1514 ;
1515 002452 000020 000000          GDUST: .WORD 16.,0
1516 002456 000000 000000          .WORD 0,0
1517 002462 000000 000000          .WORD 0,0
1518 002466 000001 000000          .WORD OP.GDS,0
1519 ;
1520 ;
1521 ;STARS
1522 ;

```

```

1523 ;DUP EXECUTE LOCAL PROGRAM COMMAND PACKET
1524 ;
1525 ;:*****
1526
1527 002472 000022 000000 XLOCPR: .WORD 18.,0
1528 002476 000000 000000 .WORD 0,0
1529 002502 000000 000000 .WORD 0,0
1530 002506 000003 000000 .WORD OP.ELP,0
1531 002512 104 111 101 .ASCII /DIAGL2/
1532
1533 ;STARS
1534 ;
1535 ;DUP RECEIVE DATA COMMAND PACKET
1536 ;
1537 ;:*****
1538
1539
1540 002520 000024 000000 RCVDAT: .WORD 20.,0
1541 002524 000000 000000 .WORD 0,0
1542 002530 000000 000000 .WORD 0,0
1543 002534 000005 000000 .WORD OP.REC,0
1544 002540 000226 000000 .WORD 150.,0
1545 002544 060000 000000 .WORD RDBUF,0
1546
1547
1548 ;STARS
1549 ;
1550 ;DUP ABORT COMMAND PACKET
1551 ;
1552 ;:*****
1553
1554 002550 000014 000000 ABORT: .WORD 12.,0
1555 002554 000000 000000 .WORD 0,0
1556 002560 000000 000000 .WORD 0,0
1557 002564 000006 000000 .WORD OP.ABT,0
1558

```

```

1560      ;*****
1561      ;*****
1562      ;
1563      ;CLASS DRIVER BUFFERS
1564      ;
1565      ;*****
1566      ;*****
1567      ;
1568 002570 RESPBF::      .BLKW  2.      ;TOP 4 LOCATIONS OF RESPONSE BUFFER
1569 002574 RSPBUF::      .BLKW  66.      ;DRIVER RESPONSE BUFFER
1570      ;
1571      ;
1572      ;*****
1573      ;*****
1574      ;
1575      ;U/Q PORT DESCRIPTOR RINGS
1576      ;
1577      ;*****
1578      ;*****
1579      ;
1580 003000 DSCRNG::      .BLKW  2.      ;DESCRIPTOR RING
1581 003004 RSPEND::      .BLKW  2.      ;END OF RESPONSE BUFFER
1582 003004 RSPRNG::      .BLKW  4.      ;RESPONSE DESCRIPTOR RING
1583 003014 CMDRNG::      .BLKW  4.      ;COMMAND DESCRIPTOR RING
1584 003024 DSCEND::      .BLKW  4.      ;END OF DESCRIPTOR RING
1585      ;
1586      ;
1587      ;*****
1588      ;*****
1589      ;
1590      ;CLASS AND PORT DRIVER VARIABLES
1591      ;
1592      ;*****
1593      ;*****
1594      ;
1595 003024 000000 CNTHI::      .WORD  0      ;VALUE OF THE HIGH TIMEOUT
1596 003026 000000 CNTFLG::      .WORD  0      ;CONTROLLER FLAGS
1597 003030 000000 PCKSIZ::      .WORD  0      ;PACKET SIZE IN BYTES
1598 003032 000000 CMDREF::      .WORD  0      ;COMMAND REFERENCE NUMBER
1599 003034 000000 CMDCNT::      .WORD  0      ;COMMAND COUNT
1600 003036 000000 WRBUF::      .BLKW  4096.      ;WRITE BUFFER
1601 023036 000000 CMDSAV::      .WORD  0      ;COMMAND DESCRIPTOR SAVE
1602 023040 000000 RSPSAV::      .WORD  0      ;RESPONSE DESCRIPTOR SAVE
1603 023042 000000 CURCMD::      .WORD  0      ;POINTER TO CURRENT COMMAND ASCII
1604

```

```
1606 ;:*****  
1607 ;:*****  
1608 ;  
1609 ;PROTECTION TABLE  
1610 ;  
1611 ;:*****  
1612 ;:*****  
1616 ;  
1617 023044 BGNPROT  
      023044 L$PROT::  
1618 023044 000000 .WORD 0  
1619 023046 177777 .WORD -1  
1620 023050 177777 .WORD -1  
1621 ;  
1622 023052 ENDPROT  
1623 ;
```

```

1625 023052 STARS
1626 023052 ;*****
STARS
;*****
1627 ;
1628 ;LEVEL 2 MICRODIAGNOSTIC ERROR TABLE
1629 ;
1630 023052 STARS
1631 023052 ;*****
STARS
;*****
1632
1633 023052 ERRTABL
023052 L$ERRTABL::
023052 000000 ERRRTYP:: .WORD 0
023054 000000 ERRNBR:: .WORD 0
023056 000000 ERRMSG:: .WORD 0
023060 000000 ERRBLK:: .WORD 0

1634
1635 023062 L2ETBL:: ;START OF TMSCP COMMAND ERROR MESSAGES
1636 ;
1637 023062 000001 .WORD 1. ;INVALID COMMAND
1638 023064 000145 .WORD 101.
1639 023066 030763 .WORD L2ER1
1640 023070 033544 .WORD L2DUMP
1641 ;
1642 023072 000001 .WORD 1. ;COMMAND ABORTED
1643 023074 000146 .WORD 102.
1644 023076 031003 .WORD L2ER2
1645 023100 033544 .WORD L2DUMP
1646 ;
1647 023102 000001 .WORD 1. ;UNIT-OFFLINE
1648 023104 000147 .WORD 103.
1649 023106 031023 .WORD L2ER3
1650 023110 033544 .WORD L2DUMP
1651 ;
1652 023112 000001 .WORD 1. ;UNIT-AVAILABLE
1653 023114 000150 .WORD 104.
1654 023116 031040 .WORD L2ER4
1655 023120 033544 .WORD L2DUMP
1656 ;
1657 023122 000001 .WORD 1. ;INVALID STATUS
1658 023124 000151 .WORD 105.
1659 023126 031057 .WORD L2ER5
1660 023130 033544 .WORD L2DUMP
1661 ;
1662 023132 000001 .WORD 1. ;WRITE PROTECTED
1663 023134 000152 .WORD 106
1664 023136 031107 .WORD L2ER6
1665 023140 033544 .WORD L2DUMP
1666 ;
1667 023142 000001 .WORD 1. ;COMPARE ERROR
1668 023144 000153 .WORD 107.
1669 023146 031127 .WORD L2ER7
1670 023150 033544 .WORD L2DUMP
1671 ;
1672 023152 000001 .WORD 1. ;DATA ERROR

```


1673	023154	000154	.WORD	108.	
1674	023156	031145	.WORD	L2ER8	
1675	023160	033544	.WORD	L2DUMP	
1676					
1677	023162	000001	.WORD	1.	;HOST BUFFER ACCESS ERROR
1678	023164	000155	.WORD	109.	
1679	023166	031160	.WORD	L2ER9	
1680	023170	033544	.WORD	L2DUMP	
1681					
1682	023172	000001	.WORD	1.	;CONTROLLER ERROR
1683	023174	000156	.WORD	110.	
1684	023176	031211	.WORD	L2ER10	
1685	023200	033544	.WORD	L2DUMP	
1686					
1687	023202	000001	.WORD	1.	;DRIVE ERROR
1688	023204	000157	.WORD	111.	
1689	023206	031232	.WORD	L2ER11	
1690	023210	033544	.WORD	L2DUMP	
1691					
1692	023212	000001	.WORD	1.	;FORMATTER ERROR
1693	023214	000160	.WORD	112.	
1694	023216	031246	.WORD	L2ER12	
1695	023220	033544	.WORD	L2DUMP	
1696					
1697	023222	000001	.WORD	1.	;BOT ENCOUNTERED
1698	023224	000161	.WORD	113.	
1699	023226	031266	.WORD	L2ER13	
1700	023230	033544	.WORD	L2DUMP	
1701					
1702	023232	000001	.WORD	1.	;TAPE MARK ENCOUNTERED
1703	023234	000162	.WORD	114.	
1704	023236	031306	.WORD	L2ER14	
1705	023240	033544	.WORD	L2DUMP	
1706					
1707	023242	000001	.WORD	1.	;INVALID STATUS
1708	023244	000163	.WORD	115.	
1709	023246	031334	.WORD	L2ER15	
1710	023250	033544	.WORD	L2DUMP	
1711					
1712	023252	000001	.WORD	1.	;RECORD DATA TRUNCATED
1713	023254	000164	.WORD	116.	
1714	023256	031364	.WORD	L2ER16	
1715	023260	033544	.WORD	L2DUMP	
1716					
1717	023262	000001	.WORD	1.	;POSITION LOST
1718	023264	000165	.WORD	117.	
1719	023266	031412	.WORD	L2ER17	
1720	023270	033544	.WORD	L2DUMP	
1721					
1722	023272	000001	.WORD	1.	;SERIOUS EXCEPTION
1723	023274	000166	.WORD	118.	
1724	023276	031430	.WORD	L2ER18	
1725	023300	033544	.WORD	L2DUMP	
1726					
1727	023302	000001	.WORD	1.	;LEOT DETECTED
1728	023304	000167	.WORD	119.	
1729	023306	031451	.WORD	L2ER19	

1730	023310	033544	.WORD	L2DUMP	
1731					
1732	023312		L2MSG:		;START OF MICRODIAGNOSTIC DETECTED ERROR MESSAGES
1733					
1734	023312	000001	DCERR:	.WORD 1.	;DATA COMPARE ERROR
1735	023314	000170		.WORD 120.	
1736	023316	031467		.WORD L2ER20	
1737	023320	033544		.WORD L2DUMP	
1738					
1739	023322	000001	CNTERR:	.WORD 1.	;CONTROLLER ERROR
1740	023324	000171		.WORD 121.	
1741	023326	031551		.WORD L2ER21	
1742	023330	033544		.WORD L2DUMP	
1743					
1744	023332	000001	INVSTA:	.WORD 1.	;INVALID STATUS
1745	023334	000172		.WORD 122.	
1746	023336	031631		.WORD L2ER22	
1747	023340	033544		.WORD L2DUMP	
1748					
1749	023342	000001	BPERR:	.WORD 1.	;BAD PATTERN NUMBER
1750	023344	000173		.WORD 123.	
1751	023346	031720		.WORD L2ER23	
1752	023350	033544		.WORD L2DUMP	
1753					
1754	023352	000001	RSPADD:	.WORD 1.	;RESPONSE ADDRESS ERROR
1755	023354	000174		.WORD 124.	
1756	023356	032010		.WORD L2ER24	
1757	023360	033544		.WORD L2DUMP	
1758					
1759	023362	000001	HBFAADD:	.WORD 1.	;HOST BUFFER ADDRESS ERROR
1760	023364	000175		.WORD 125.	
1761	023366	032076		.WORD L2ER25	
1762	023370	033544		.WORD L2DUMP	
1763					
1764	023372	000001	UNERLG:	.WORD 1.	;UNKNOWN ERROR LOG RECEIVED
1765	023374	000176		.WORD 126.	
1766	023376	032167		.WORD L2ER26	
1767	023400	033544		.WORD L2DUMP	
1768					
1769	023402	000001	RSPTO:	.WORD 1.	;RESPONSE TIMEOUT ERROR
1770	023404	000177		.WORD 127.	
1771	023406	032261		.WORD L2ER27	
1772	023410	033544		.WORD L2DUMP	

```

1774 .SBTTL GLOBAL TEXT SECTION
1778 ;*****
1779 ;*****
1780 ;
1781 ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
1782 ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
1783 ; MORE THAN ONE TEST.
1784 ;
1785 ;*****
1786 ;*****
1787 ;
1788 ;*****
1789 ;*****
1790 ;
1791 ; NAMES OF DEVICES SUPPORTED BY PROGRAM
1792 ;
1793 ;*****
1797 ;*****
1798 023412          DEVTYP <TK50>
      023412      L$DVTYP::
      023412      .ASCIZ *TK50*
1799              .EVEN

```

124 113 065

```

1804
1805
1806
1807
1808
1809
1810
1811 023420      045    101    111  LINE1:: .ASCIZ  ?%AINIT SEQUENCE STEP #: %D1?
1812 023454      045    116    045  LINE2:: .ASCIZ  ?%N%ASA REG: %06%A EXPCTD: %06%A  ACTUAL SA: %06?
1813 023534      045    116    045  LINE3:: .ASCIZ  ?%N%AIP REG ADDRESS: %06?
1814 023564      045    116    045  LINE4:: .ASCIZ  ?%N%A****FAILING FRU: %T%A****%N%N?
1815 023626      045    101    122  LINE5:: .ASCIZ  ?%ARELOCATION CONSTANT: %06%A  VIRT. ADD: %06?
1816 023703      045    116    045  LINE6:: .ASCIZ  ?%N%AEXPECTED: %06%A  RECEIVED: %06?
1817 023746      045    101    120  LINE7:: .ASCIZ  ?%APHYSICAL ADD: %06?
1818
1819 023772      045    116    045  WR1::   .ASCIZ  ?%N%ASA REG: %06%A  SA CONTENTS: %06?
1820 024035      045    116    045  WR2::   .ASCIZ  ?%N%AFAILING COMMAND : %T%N?
1821 024071      045    116    045  WR3::   .ASCIZ  ?%N%A  RESPONSE PACKET%N?
1822 024122      045    116    045  WR4::   .ASCIZ  ?%N%A  %06%A  %06?
1823 024147      045    116    045  WR5::   .ASCIZ  ?%N%ACMD EXP: %06%A  CMD REC: %06?
1824 024211      045    116    000  WR6::   .ASCIZ  ?%N?
1825 024214      045    116    045  WR7::   .ASCIZ  ?%N%AFAILING COMMAND : %T%N%AUNIT UNKNOWN?
1826 024265      045    116    045  WR8::   .ASCIZ  ?%N%AFAILING COMMAND : %T%N%AAND MEDIA MOUNTED?
1827 024342      045    116    045  WR9::   .ASCIZ  ?%N%AFAILING COMMAND : %T%N%AUNIT INOPERATIVE?
1828
1829
1830 024417      045    116    062  L21::   .ASCIZ  ?%N2%A**** LEVEL 2 MICRODIAGNOSTIC DUMP ****?
1831 024473      045    116    062  L22::   .ASCIZ  ?%N2%APROGRAM STATUS: %06?
1832 024531      045    116    045  L23::   .ASCIZ  ?%N%ADRIVE STATUS: %03?
1833 024571      045    116    045  L24::   .ASCIZ  ?%N%AATEST NUMBER: %03?
1834 024631      045    116    045  L25::   .ASCIZ  ?%N%ATRACK NUMBER: %03?
1835 024671      045    116    045  L26::   .ASCIZ  ?%N%ATMSCP COMMAND: %03?
1836 024731      045    116    045  L27::   .ASCIZ  ?%N%ATMSCP RESPONSE FLAG: %03?
1837 024771      045    116    045  L28::   .ASCIZ  ?%N%ATMSCP UNIT FLAG: %06?
1838 025026      045    116    062  L29::   .ASCIZ  ?%N2%ABLOCKS WRITTEN CHANNEL 1: %D6?
1839 025073      045    116    045  L210::  .ASCIZ  ?%N%ABLOCKS WRITTEN CHANNEL 2: %D6?
1840 025137      045    116    045  L211::  .ASCIZ  ?%N%ABLOCKS READ CHANNEL 1: %D6?
1841 025203      045    116    045  L212::  .ASCIZ  ?%N%ABLOCKS READ CHANNEL 2: %D6?
1842 025247      045    116    045  L213::  .ASCIZ  ?%N%AASOFT WRITE CHANNEL 1: %D6?
1843 025313      045    116    045  L214::  .ASCIZ  ?%N%AASOFT WRITE CHANNEL 2: %D6?
1844 025357      045    116    045  L215::  .ASCIZ  ?%N%AECC CORRECTED CHANNEL 1: %D6?
1845 025423      045    116    045  L216::  .ASCIZ  ?%N%AECC CORRECTED CHANNEL 2: %D6?
1846 025467      045    116    045  L219::  .ASCIZ  ?%N%AREAD REPOSITIONS CHANNEL 1: %D6?
1847 025533      045    116    045  L220::  .ASCIZ  ?%N%AREAD REPOSITIONS CHANNEL 2: %D6%N?
1848 025601      045    116    062  L221::  .ASCIZ  ?%N2%AERROR LOG (TAPE TRANSFER ERROR)?
1849 025646      045    116    045  L222::  .ASCIZ  ?%N%AERROR FORMAT: %03?
1850 025710      045    116    045  L223::  .ASCIZ  ?%N%AERROR FLAGS: %03?
1851 025752      045    116    045  L224::  .ASCIZ  ?%N%AEVENT CODE: %06?
1852 026011      045    116    045  L225::  .ASCIZ  ?%N%AERROR RETRY LEVEL: %03?
1853 026053      045    116    045  L226::  .ASCIZ  ?%N%ANUMBER OF RETRIES: %03?
1854 026115      045    116    045  L227::  .ASCIZ  ?%N%APOSITION (LOW ORDER): %06?
1855 026154      045    116    045  L228::  .ASCIZ  ?%N%APOSITION (HIG4 ORDER): %06?
1856 026213      045    116    045  L229::  .ASCIZ  ?%N%ACONTROLLER STATUS: %03?
1857 026255      045    116    045  L230::  .ASCIZ  ?%N%ADRIVE ERROR CODE: %03?
1858 026317      045    116    045  L231::  .ASCIZ  ?%N%ADRIVE FLAGS: %03?
1859 026361      045    116    045  L232::  .ASCIZ  ?%N%ATRACK NUMBER: %D3?
1860 026423      045    116    045  L233::  .ASCIZ  ?%N%APHYSICAL BLOCK NUMBER: %D6?

```

1861	026462	045	116	045	L234::	.ASCIZ	?N#ALOGICAL BLOCK NUMBER:	#D3?
1862	026524	045	116	045	L235::	.ASCIZ	?N#ATAPE COUNT 0 :	#03?
1863	026566	045	116	045	L236::	.ASCIZ	?N#ATAPE COUNT 1 :	#03?
1864	026630	045	116	045	L237::	.ASCIZ	?N#ATAPE COUNT 2 :	#03?
1865	026672	045	116	045	L238::	.ASCIZ	?N#ADRIIVE STATE:	#06?
1866	026731	045	116	045	L239::	.ASCIZ	?N#AREAD/WRITE STATE:	#06?
1867	026770	045	116	045	L240::	.ASCIZ	?N#AOPERATION FLAGS:	#06?
1868	027027	045	116	062	L241::	.ASCIZ	?N#AERROR LOG (CONTROLLER ERROR)?	
1869	027071	045	116	045	L242::	.ASCIZ	?N#ACONTROLLER ERROR CODE:	#06?
1870								
1871	027130	045	116	062	L244::	.ASCIZ	?N#A#ABYTE READ: #03#A	BYTE WRITTEN: #03#A
1872	027224	045	116	045	L245::	.ASCIZ	?N#A	#03#A
1873	027320	045	116	045	L246::	.ASCIZ	?N#A	#03#A
1874	027414	045	116	045	L247::	.ASCIZ	?N#A	#03#A
1875	027510	045	116	045	L248::	.ASCIZ	?N#A	#03#A
1876	027604	045	116	045	L249::	.ASCIZ	?N#A	#03#A
1877	027700	045	116	045	L250::	.ASCIZ	?N#A	#03#A
1878	027774	045	116	045	L251::	.ASCIZ	?N#A	#03#A
1879	030070	045	116	045	L252::	.ASCIZ	?N#A	#03#A
1880	030164	045	116	045	L253::	.ASCIZ	?N#A	#03#A
1881	030260	045	116	062	L254::	.ASCIZ	?N#A#ATOTAL NUMBER OF MISCOMPARED BYTES:	#06?
1882							.EVEN	
1883								
1884								
1885								
1886								
1887								
1888								
1889								
1890								
1891	030334	116	130	115	MSG5::	.ASCIZ	?NXM ON READ TKIP?	
1892	030355	124	113	111	MSG6::	.ASCIZ	?TKIP NOT 0 ON FIRST READ?	
1893	030406	116	130	115	MSG7::	.ASCIZ	?NXM ON READ TKSA?	
1894	030427	123	101	040	MSG8::	.ASCIZ	?SA REG IN ERROR ON FIRST READ?	
1895	030465	123	101	040	MSG9::	.ASCIZ	?SA CONTENTS IN ERROR?	
1896	030512	123	101	040	MSG10::	.ASCIZ	?SA WRONG IN DATA WRAP?	
1897	030540	105	130	120	MSG11::	.ASCIZ	?EXPECTED INTERRUPT DID NOT OCCUR?	
1898	030601	111	116	124	MSG12::	.ASCIZ	?INTRRPT OCCURRED WITH CPU PRIORITY = 7?	
1899	030650	123	101	040	MSG13::	.ASCIZ	?SA NOT 0 IN PURGE/POLL?	
1900	030677	120	125	122	MSG14::	.ASCIZ	?PURGE/POLL TEST FAILED?	
1901	030726	105	130	124	MSG15::	.ASCIZ	?EXTEJDED ADDRESS TEST FAILED?	
1902								
1903	030763	111	116	126	L2ER1::	.ASCIZ	?INVALID COMMAND?	
1904	031003	103	117	115	L2ER2::	.ASCIZ	?COMMAND ABORTED?	
1905	031023	125	116	111	L2ER3::	.ASCIZ	?UNIT-OFFLINE?	
1906	031040	125	116	111	L2ER4::	.ASCIZ	?UNIT-AVAILABLE?	
1907	031057	111	116	126	L2ER5::	.ASCIZ	?INVALID STATUS RECEIVED?	
1908	031107	127	122	111	L2ER6::	.ASCIZ	?WRITE PROTECTED?	
1909	031127	103	117	115	L2ER7::	.ASCIZ	?COMPARE ERROR?	
1910	031145	104	101	124	L2ER8::	.ASCIZ	?DATA ERROR?	
1911	031160	110	117	123	L2ER9::	.ASCIZ	?HOST BUFFER ACCESS ERROR?	
1912	031211	103	117	116	L2ER10::	.ASCIZ	?CONTROLLER ERROR?	
1913	031232	104	122	111	L2ER11::	.ASCIZ	?DRIVE ERROR?	
1914	031246	106	117	122	L2ER12::	.ASCIZ	?FORMATTER ERROR?	
1915	031266	102	117	124	L2ER13::	.ASCIZ	?BOT ENCOUNTERED?	
1916	031306	124	101	120	L2ER14::	.ASCIZ	?TAPE MARK ENCOUNTERED?	
1917	031334	111	116	126	L2ER15::	.ASCIZ	?INVALID STATUS RECEIVED?	

```

;*****
;
;ERROR MESSAGES
;
;*****
    
```

```

1918 031364      122      105      103 L2ER16::ASCIZ ?RECORD DATA TRUNCATED?
1919 031412      120      117      123 L2ER17::ASCIZ ?POSITION LOST?
1920 031430      123      105      122 L2ER18::ASCIZ ?SERIOUS EXEPTION?
1921 031451      114      105      117 L2ER19::ASCIZ ?LEOT DETECTED?
1922 031467      115      111      103 L2ER20::ASCIZ ?MICRODIAGNOSTIC PROGRAM ERROR: DATA COMPARE ERROR?
1923 031551      115      111      103 L2ER21::ASCIZ ?MICRODIAGNOSTIC PROGRAM ERROR: CONTROLLER ERROR?
1924 031631      115      111      103 L2ER22::ASCIZ ?MICRODIAGNOSTIC PROGRAM ERROR: INVALID STATUS RECEIVED?
1925 031720      115      111      103 L2ER23::ASCIZ ?MICRODIAGNOSTIC PROGRAM ERROR: BAD PATTERN NUMBER ERROR?
1926 032010      115      111      103 L2ER24::ASCIZ ?MICRODIAGNOSTIC PROGRAM ERROR: RESPONSE ADDRESS ERROR?
1927 032076      115      111      103 L2ER25::ASCIZ ?MICRODIAGNOSTIC PROGRAM ERROR: HOST BUFFER ADDRESS ERROR?
1928 032167      115      111      103 L2ER26::ASCIZ ?MICRODIAGNOSTIC PROGRAM ERROR: UNKNOWN ERROR LOG RECEIVED?
1929 032261      115      111      103 L2ER27::ASCIZ ?MICRODIAGNOSTIC PROGRAM ERROR: RESPONSE TIMEOUT ERROR?
1930
1931 032347      124      111      115 WRER1::ASCIZ ?TIME OUT DURING PORT INIT?
1932 032401      120      117      122 WRER2::ASCIZ ?PORT INIT FAILED?
1933 032422      124      115      123 WRER3::ASCIZ ?TMSCP COMMAND FAILURE?
1934 032450      120      117      122 WRER4::ASCIZ ?PORT DETECTED ERROR?
1935 032474      122      105      123 WRER5::ASCIZ ?RESPONSE OUT OF SEQUENCE?
1936 032525      103      117      115 WRER6::ASCIZ ?COMMAND TIME OUT?
1937 032546      125      116      105 WRER7::ASCIZ ?UNEXPECTED STATUS FOR ONLINE RECEIVED?
1938 032614      104      125      120 WRER8::ASCIZ ?DUP COMMAND FAILURE?
1939
1940
1941
1942
1943
1944
1945
1946
1947 032640      103      116      124 CTRL::ASCIZ ?CNTRLR/CABLE?
1948 032655      104      122      111 DRVE::ASCIZ ?DRIVE?
1949 032663      123      103      103 SCCCMD::ASCIZ ?SCC ?
1950 032670      117      116      114 ONLCMD::ASCIZ ?ONL ?
1951 032675      122      105      127 REWCMD::ASCIZ ?REW ?
1952 032702      127      122      040 WRCMD::ASCIZ ?WR ?
1953 032707      122      104      040 RDCMD::ASCIZ ?RD ?
1954 032714      107      104      123 GDSCMD::ASCIZ ?GDS ?
1955 032721      105      114      120 ELPCMD::ASCIZ ?ELP ?
1956 032726      122      103      126 RCVCMD::ASCIZ ?RCV ?
1957 032733      101      102      124 ABTCMD::ASCIZ ?ABT ?
1958
1959

```

```

;*****
;
;MISCELLANEOUS ERROR MESSAGES
;
;*****

```

1961
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1979
1980 032740
1981
1982 032740
1983 032740
032740 013746 002316
032744 012746 023420
032750 012746 000002
032754 010600
032756 104415
032760 062706 000006
1984
1985 032764
1986 032764
032764 016446 000012
032770 013746 002314
032774 016446 000002
033000 012746 023454
033004 012746 000004
033010 010600
033012 104415
033014 062706 000012
1987 033020 000452
1988
1989 033022
1990 033022
033022 010246
033024 012746 023746
033030 012746 000002
033034 010600
033036 104415
033040 062706 000006
1991 033044 000413
1992
1993 033046
1994 033046
033046 010246
033050 013746 172346
033054 012746 023626
033060 012746 000003
033064 010600
033066 104415
033070 062706 000010
1995
1996 033074

```
.SBTTL GLOBAL ERROR REPORT SECTION
;*****
;*****
;GLOBAL ERROR REPORTS
;THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB
;AND PRINTX CALLS THAT ARE USED IN MORE THAN ONE TEST.
;IT ALSO INCLUDES THE ASCII MESSAGES THAT ARE USED BY
;THE PRINTB AND PRINTX CALLS.
;*****
;*****

          BGNMSG

PRIINI::
PRINTX   #LINE1,INISTP
MOV      INISTP,-(SP)
MOV      #LINE1,-(SP)
MOV      #2,-(SP)
MOV      SP,R0
TRAP    C$PNTX
ADD      #6,SP

PRISA::
PRINTX   #LINE2,TKSA(R4),SAEXP,TKSASV(R4)
MOV      TKSASV(R4),-(SP)
MOV      SAEXP,-(SP)
MOV      TKSA(R4),-(SP)
MOV      #LINE2,-(SP)
MOV      #4,-(SP)
MOV      SP,R0
TRAP    C$PNTX
ADD      #12,SP
BR       PRIERR

PRIPAD::
PRINTX   #LINE7,R2
MOV      R2,-(SP)
MOV      #LINE7,-(SP)
MOV      #2,-(SP)
MOV      SP,R0
TRAP    C$PNTX
ADD      #6,SP
BR       PRIDAT

PRIVAD::
PRINTX   #LINE5,KPAR3,R2
MOV      R2,-(SP)
MOV      KPAR3,-(SP)
MOV      #LINE5,-(SP)
MOV      #3,-(SP)
MOV      SP,R0
TRAP    C$PNTX
ADD      #10,SP

PRIDAT::
```

1997	033074			PRINTX	#LINE6,R1,(R2)
	033074	011246		MOV	(R2),-(SP)
	033076	010146		MOV	R1, -(SP)
	033100	012746	023703	MOV	#LINE6, -(SP)
	033104	012746	000003	MOV	#3, -(SP)
	033110	010600		MOV	SP,R0
	033112	104415		TRAP	C\$PNTX
	033114	062706	000010	ADD	#10,SP
1998	033120	000412		BR	PRIERR
1999					
2000	033122			PRIIP::	
2001	033122			PRINTX	#LINE3,TKIP(R4)
	033122	016446	000000	MOV	TKIP(R4),-(SP)
	033126	012746	023534	MOV	#LINE3, -(SP)
	033132	012746	000002	MOV	#2, -(SP)
	033136	010600		MOV	SP,R0
	033140	104415		TRAP	C\$PNTX
	033142	062706	000006	ADD	#6,SP
2002					
2003	033146			PRIERR::	
2004	033146			PRINTB	#LINE4,FRUIS
	033146	013746	002310	MOV	FRUIS, -(SP)
	033152	012746	023564	MOV	#LINE4, -(SP)
	033156	012746	000002	MOV	#2, -(SP)
	033162	010600		MOV	SP,R0
	033164	104414		TRAP	C\$PNTB
	033166	062706	000006	ADD	#6,SP
2005	033172	000137	033536	JMP	PRIEX
2006					
2007	033176			WRINTO::	
2008	033176			PRINTX	#LINE1,INISTP
	033176	013746	002316	MOV	INISTP, -(SP)
	033202	012746	023420	MOV	#LINE1, -(SP)
	033206	012746	000002	MOV	#2, -(SP)
	033212	010600		MOV	SP,R0
	033214	104415		TRAP	C\$PNTX
	033216	062706	000006	ADD	#6,SP
2009					
2010	033222			WRPRTE::	
2011	033222			PRINTX	#WR1,TKSA(R4),TKSASV(R4)
	033222	016446	000012	MOV	TKSASV(R4),-(SP)
	033226	016446	000002	MOV	TKSA(R4),-(SP)
	033232	012746	023772	MOV	#WR1, -(SP)
	033236	012746	000003	MOV	#3, -(SP)
	033242	010600		MOV	SP,R0
	033244	104415		TRAP	C\$PNTX
	033246	062706	000010	ADD	#10,SP
2012	033252	000137	033512	JMP	WREX
2013					
2014	033256			WRCMDE::	
2015	033256			PRINTX	R2,CURCMD
	033256	013746	023042	MOV	CURCMD, -(SP)
	033262	010246		MOV	R2, -(SP)
	033264	012746	000002	MOV	#2, -(SP)
	033270	010600		MOV	SP,R0
	033272	104415		TRAP	C\$PNTX
	033274	062706	000006	ADD	#6,SP

2016	033300			PRINTX	#WR3	
	033300	012746	024071	MOV	#WR3,-(SP)	
	033304	012746	000001	MOV	#1,-(SP)	
	033310	010600		MOV	SP,R0	
	033312	104415		TRAP	C\$PNTX	
	033314	062706	000004	ADD	#4,SP	
2017	033320	010301		MOV	R3,R1	
2018	033322			1\$: PRINTX	#WR4,2(R1),(R1)	
	033322	011146		MOV	(R1),-(SP)	
	033324	016146	000002	MOV	2(R1),-(SP)	
	033330	012746	024122	MOV	#WR4,-(SP)	
	033334	012746	000003	MOV	#3,-(SP)	
	033340	010600		MOV	SP,R0	
	033342	104415		TRAP	C\$PNTX	
	033344	062706	000010	ADD	#10,SP	
2019	033350	062701	000004	ADD	#4,R1	
2020	033354	162763	000004	177774	SUB	#4,MSGLEN(R3)
2021	033362	001357		BNE	1\$	
2022	033364			PRINTX	#WR6	
	033364	012746	024211	MOV	#WR6,-(SP)	
	033370	012746	000001	MOV	#1,-(SP)	
	033374	010600		MOV	SP,R0	
	033376	104415		TRAP	C\$PNTX	
	033400	062706	000004	ADD	#4,SP	
2023	033404	000137	033512	JMP	WREX	
2024						
2025	033410			WRSEQE::		
2026	033410			PRINTX	#WR2,CURCMD	
	033410	013746	023042	MOV	CURCMD,-(SP)	
	033414	012746	024035	MOV	#WR2,-(SP)	
	033420	012746	000002	MOV	#2,-(SP)	
	033424	010600		MOV	SP,R0	
	033426	104415		TRAP	C\$PNTX	
	033430	062706	000006	ADD	#6,SP	
2027	033434			PRINTX	#WR5,P.CRF(R3),P.CRF(R5)	
	033434	016546	000000	MOV	P.CRF(R5),-(SP)	
	033440	016346	000000	MOV	P.CRF(R3),-(SP)	
	033444	012746	024147	MOV	#WR5,-(SP)	
	033450	012746	000003	MOV	#3,-(SP)	
	033454	010600		MOV	SP,R0	
	033456	104415		TRAP	C\$PNTX	
	033460	062706	000010	ADD	#10,SP	
2028	033464	000412		BR	WREX	
2029						
2030	033466			WRTOE::		
2031	033466			PRINTX	#WR2,CURCMD	
	033466	013746	023042	MOV	CURCMD,-(SP)	
	033472	012746	024035	MOV	#WR2,-(SP)	
	033476	012746	000002	MOV	#2,-(SP)	
	033502	010600		MOV	SP,R0	
	033504	104415		TRAP	C\$PNTX	
	033506	062706	000006	ADD	#6,SP	
2032						
2033	033512			WREX::		
2034	033512			PRINTB	#LINE4,FRUIS	
	033512	013746	002310	MOV	FRUIS,-(SP)	
	033516	012746	023564	MOV	#LINE4,-(SP)	

033522	012746	000002	MOV	#2,-(SP)
033526	010600		MOV	SP,R0
033530	104414		TRAP	C\$PNTB
033532	062706	000006	ADD	#6,SP
2035				
2036	033536		PRIEX: EXIT	MSG
	033536	000167	.WORD	J\$JMP
	033540	000000	.WORD	L10003-2-.
2037				
2038	033542		ENDMSG	
	033542		L10003:	
	033542	104423	TRAP	C\$MSG
2039				
2040	033544		BGNMSG	L2DUMP
	033544		L2DUMP::	
2041				
2042	033544		PRINTB	#LINE4,FRUIS
	033544	013746	MOV	FRUIS,-(SP)
	033550	012746	MOV	#LINE4,-(SP)
	033554	012746	MOV	#2,-(SP)
	033560	010600	MOV	SP,R0
	033562	104414	TRAP	C\$PNTB
	033564	062706	ADD	#6,SP
2043	033570		PRINTB	#L21
	033570	012746	MOV	#L21,-(SP)
	033574	012746	MOV	#1,-(SP)
	033600	010600	MOV	SP,R0
	033602	104414	TRAP	C\$PNTB
	033604	062706	ADD	#4,SP
2044	033610		PRINTB	#L22,L2STA(R1)
	033610	016146	MOV	L2STA(R1),-(SP)
	033614	012746	MOV	#L22,-(SP)
	033620	012746	MOV	#2,-(SP)
	033624	010600	MOV	SP,R0
	033626	104414	TRAP	C\$PNTB
	033630	062706	ADD	#6,SP
2045	033634		PRINTB	#L23,L2DRV(R1)
	033634	016146	MOV	L2DRV(R1),-(SP)
	033640	012746	MOV	#L23,-(SP)
	033644	012746	MOV	#2,-(SP)
	033650	010600	MOV	SP,R0
	033652	104414	TRAP	C\$PNTB
	033654	062706	ADD	#6,SP
2046	033660		PRINTB	#L24,<B,L2TST(R1)>
	033660	005046	CLR	-(SP)
	033662	156116	BISB	L2TST(R1),(SP)
	033666	012746	MOV	#L24,-(SP)
	033672	012746	MOV	#2,-(SP)
	033676	010600	MOV	SP,R0
	033700	104414	TRAP	C\$PNTB
	033702	062706	ADD	#6,SP
2047	033706		PRINTB	#L25,<B,L2TRK(R1)>
	033706	005046	CLR	-(SP)
	033710	156116	BISB	L2TRK(R1),(SP)
	033714	012746	MOV	#L25,-(SP)
	033720	012746	MOV	#2,-(SP)
	033724	010600	MOV	SP,R0

	033726	104414			TRAP	C\$PNTB
	033730	062706	000006		ADD	#6,SP
2048	033734				PRINTB	#L26,<B,L2CMD(R1)>
	033734	005046			CLR	-(SP)
	033736	156116	000010		BISB	L2CMD(R1),(SP)
	033742	012746	024671		MOV	#L26,-(SP)
	033746	012746	000002		MOV	#2,-(SP)
	033752	010600			MOV	SP,R0
	033754	104414			TRAP	C\$PNTB
	033756	062706	000006		ADD	#6,SP
2049	033762				PRINTB	#L27,<B,L2RSP(R1)>
	033762	005046			CLR	-(SP)
	033764	156116	000011		BISB	L2RSP(R1),(SP)
	033770	012746	024731		MOV	#L27,-(SP)
	033774	012746	000002		MOV	#2,-(SP)
	034000	010600			MOV	SP,R0
	034002	104414			TRAP	C\$PNTB
	034004	062706	000006		ADD	#6,SP
2050	034010				PRINTB	#L28,L2UNT(R1)
	034010	016146	000012		MOV	L2UNT(R1),-(SP)
	034014	012746	024771		MOV	#L28,-(SP)
	034020	012746	000002		MOV	#2,-(SP)
	034024	010600			MOV	SP,R0
	034026	104414			TRAP	C\$PNTB
	034030	062706	000006		ADD	#6,SP
2051	034034	126127	000134	000005	CMPB	L2ELFM(R1),#TXFER
2052	034042	001402			BEQ	1\$
2053	034044	000137	034720		JMP	10\$
2054	034050			1\$:	PRINTB	#L221
	034050	012746	025601		MOV	#L221,-(SP)
	034054	012746	000001		MOV	#1,-(SP)
	034060	010600			MOV	SP,R0
	034062	104414			TRAP	C\$PNTB
	034064	062706	000004		ADD	#4,SP
2055	034070				PRINTB	#L222,<B,L2ELFM(R1)>
	034070	005046			CLR	-(SP)
	034072	156116	000134		BISB	L2ELFM(R1),(SP)
	034076	012746	025646		MOV	#L222,-(SP)
	034102	012746	000002		MOV	#2,-(SP)
	034106	010600			MOV	SP,R0
	034110	104414			TRAP	C\$PNTB
	034112	062706	000006		ADD	#6,SP
2056	034116				PRINTB	#L223,<B,L2ELFL(R1)>
	034116	005046			CLR	-(SP)
	034120	156116	000135		BISB	L2ELFL(R1),(SP)
	034124	012746	025710		MOV	#L223,-(SP)
	034130	012746	000002		MOV	#2,-(SP)
	034134	010600			MOV	SP,R0
	034136	104414			TRAP	C\$PNTB
	034140	062706	000006		ADD	#6,SP
2057	034144				PRINTB	#L224,L2ELEV(R1)
	034144	016146	000136		MOV	L2ELEV(R1),-(SP)
	034150	012746	025752		MOV	#L224,-(SP)
	034154	012746	000002		MOV	#2,-(SP)
	034160	010600			MOV	SP,R0
	034162	104414			TRAP	C\$PNTB
	034164	062706	000006		ADD	#6,SP

```

; "TAPE TRANSFER" ERROR LOG ?
; YES, DUMP "TAPE TRANSFER" ERROR LOG
; NO, CHECK "CONTROLLER ERROR" ERROR LOG

```

2058	034170			PRINTB	#L225,<B,L2ELRL(R1)>
	034170	005046		CLR	-(SP)
	034172	156116	000144	BISB	L2ELRL(R1),(SP)
	034176	012746	026011	MOV	#L225,-(SP)
	034202	012746	000002	MOV	#2,-(SP)
	034206	010600		MOV	SP,R0
	034210	104414		TRAP	C#PNTB
	034212	062706	000006	ADD	#6,SP
2059	034216			PRINTB	#L226,<B,L2ELRT(R1)>
	034216	005046		CLR	-(SP)
	034220	156116	000145	BISB	L2ELRT(R1),(SP)
	034224	012746	026053	MOV	#L226,-(SP)
	034230	012746	000002	MOV	#2,-(SP)
	034234	010600		MOV	SP,R0
	034236	104414		TRAP	C#PNTB
	034240	062706	000006	ADD	#6,SP
2060	034244			PRINTB	#L227,L2ELP1(R1)
	034244	016146	000146	MOV	L2ELP1(R1),-(SP)
	034250	012746	026115	MOV	#L227,-(SP)
	034254	012746	000002	MOV	#2,-(SP)
	034260	010600		MOV	SP,R0
	034262	104414		TRAP	C#PNTB
	034264	062706	000006	ADD	#6,SP
2061	034270			PRINTB	#L228,L2ELP2(R1)
	034270	016146	000150	MOV	L2ELP2(R1),-(SP)
	034274	012746	026154	MOV	#L228,-(SP)
	034300	012746	000002	MOV	#2,-(SP)
	034304	010600		MOV	SP,R0
	034306	104414		TRAP	C#PNTB
	034310	062706	000006	ADD	#6,SP
2062	034314			PRINTB	#L229,<B,L2ELST(R1)>
	034314	005046		CLR	-(SP)
	034316	156116	000154	BISB	L2ELST(R1),(SP)
	034322	012746	026213	MOV	#L229,-(SP)
	034326	012746	000002	MOV	#2,-(SP)
	034332	010600		MOV	SP,R0
	034334	104414		TRAP	C#PNTB
	034336	062706	000006	ADD	#6,SP
2063	034342			PRINTB	#L230,<B,L2ELDE(R1)>
	034342	005046		CLR	-(SP)
	034344	156116	000155	BISB	L2ELDE(R1),(SP)
	034350	012746	026255	MOV	#L230,-(SP)
	034354	012746	000002	MOV	#2,-(SP)
	034360	010600		MOV	SP,R0
	034362	104414		TRAP	C#PNTB
	034364	062706	000006	ADD	#6,SP
2064	034370			PRINTB	#L231,<B,L2ELDF(R1)>
	034370	005046		CLR	-(SP)
	034372	156116	000156	BISB	L2ELDF(R1),(SP)
	034376	012746	026317	MOV	#L231,-(SP)
	034402	012746	000002	MOV	#2,-(SP)
	034406	010600		MOV	SP,R0
	034410	104414		TRAP	C#PNTB
	034412	062706	000006	ADD	#6,SP
2065	034416			PRINTB	#L232,<B,L2ELTN(R1)>
	034416	005046		CLR	-(SP)
	034420	156116	000157	BISB	L2ELTN(R1),(SP)

	034424	012746	026361	MOV	#L232,-(SP)
	034430	012746	000002	MOV	#2,-(SP)
	034434	010600		MOV	SP,R0
	034436	104414		TRAP	C#PNTB
	034440	062706	000006	ADD	#6,SP
2066	034444			PRINTB	#L233,L2ELPB(R1)
	034444	016146	000160	MOV	L2ELPB(R1),-(SP)
	034450	012746	026423	MOV	#L233,-(SP)
	034454	012746	000002	MOV	#2,-(SP)
	034460	010600		MOV	SP,R0
	034462	104414		TRAP	C#PNTB
	034464	062706	000006	ADD	#6,SP
2067	034470			PRINTB	#L234,<B,L2ELLB(R1)>
	034470	005046		CLR	-(SP)
	034472	156116	000162	BISB	L2ELLB(R1),(SP)
	034476	012746	026462	MOV	#L234,-(SP)
	034502	012746	000002	MOV	#2,-(SP)
	034506	010600		MOV	SP,R0
	034510	104414		TRAP	C#PNTB
	034512	062706	000006	ADD	#6,SP
2068	034516			PRINTB	#L235,<B,L2ELT0(R1)>
	034516	005046		CLR	-(SP)
	034520	156116	000163	BISB	L2ELT0(R1),(SP)
	034524	012746	026524	MOV	#L235,-(SP)
	034530	012746	000002	MOV	#2,-(SP)
	034534	010600		MOV	SP,R0
	034536	104414		TRAP	C#PNTB
	034540	062706	000006	ADD	#6,SP
2069	034544			PRINTB	#L236,<B,L2ELT1(R1)>
	034544	005046		CLR	-(SP)
	034546	156116	000164	BISB	L2ELT1(R1),(SP)
	034552	012746	026566	MOV	#L236,-(SP)
	034556	012746	000002	MOV	#2,-(SP)
	034562	010600		MOV	SP,R0
	034564	104414		TRAP	C#PNTB
	034566	062706	000006	ADD	#6,SP
2070	034572			PRINTB	#L237,<B,L2ELT2(R1)>
	034572	005046		CLR	-(SP)
	034574	156116	000165	BISB	L2ELT2(R1),(SP)
	034600	012746	026630	MOV	#L237,-(SP)
	034604	012746	000002	MOV	#2,-(SP)
	034610	010600		MOV	SP,R0
	034612	104414		TRAP	C#PNTB
	034614	062706	000006	ADD	#6,SP
2071	034620			PRINTB	#L238,L2ELDS(R1)
	034620	016146	000166	MOV	L2ELDS(R1),-(SP)
	034624	012746	026672	MOV	#L238,-(SP)
	034630	012746	000002	MOV	#2,-(SP)
	034634	010600		MOV	SP,R0
	034636	104414		TRAP	C#PNTB
	034640	062706	000006	ADD	#6,SP
2072	034644			PRINTB	#L239,L2ELRW(R1)
	034644	016146	000170	MOV	L2ELRW(R1),-(SP)
	034650	012746	026731	MOV	#L239,-(SP)
	034654	012746	000002	MOV	#2,-(SP)
	034660	010600		MOV	SP,R0
	034662	104414		TRAP	C#PNTB

```

2073 034664 062706 000006      ADD    #6,SP
      034670      PRINTB  #L240,L2ELOF(R1)
      034670 016146 000172      MOV    L2ELOF(R1),-(SP)
      034674 012746 026770      MOV    #L240,-(SP)
      034700 012746 000002      MOV    #2,-(SP)
      034704 010600      MOV    SP,RO
      034706 104414      TRAP   C$PNTB
      034710 062706 000006      ADD    #6,SP
2074 034714 000137 034774      JMP    25$
2075 034720 126127 000134 000000 10$:  CMPB   L2ELFM(R1),#CINTER
2076 034726 001022      BNE    25$
2077 034730      PRINTB  #L241
      034730 012746 027027      MOV    #L241,-(SP)
      034734 012746 000001      MOV    #1,-(SP)
      034740 010600      MOV    SP,RO
      034742 104414      TRAP   C$PNTB
      034744 062706 000004      ADD    #4,SP
2078 034750      PRINTB  #L242,L2ELEC(R1)
      034750 016146 000140      MOV    L2ELEC(R1),-(SP)
      034754 012746 027071      MOV    #L242,-(SP)
      034760 012746 000002      MOV    #2,-(SP)
      034764 010600      MOV    SP,RO
      034766 104414      TRAP   C$PNTB
      034770 062706 000006      ADD    #6,SP
2079 034774      EXIT    MSG
      034774 000167      .WORD  J$JMP
      034776 000000      .WORD  L10004-2-.
2080
2081 035000      ENDMSG
      035000      L10004:
      035000 104423      TRAP   C$MSG

```

```

;"CONTROLLER ERROR" ERROR LOG ?
:NO, NO ERROR LOG TO PRINT, EXIT
:DUMP "CONTROLLER ERROR" ERROR LOG

```

```

2083 .SBTTL STATISTICAL REPORT SECTION
2084 035002 STARS
      ;:*****
2085 035002 STARS
      ;:*****
2086 ;
2087 ;STATISTICAL REPORT SECTION
2088 ; THESE ARE THE "PRINTS" STATEMENTS USED TO DUMP THE LEVEL 2
2089 ; MICRODIAGNOSTIC TEST RESULTS
2090 ;
2091 035002 STARS
      ;:*****
2092 035002 STARS
      ;:*****
2093 ;
2094 035002 BGNRPT
      035002 L$RPT::
2095 035002 PRINTS #L29,L2BWR1(R1)
      035002 MOV L2BWR1(R1),-(SP)
      035006 MOV #L29,-(SP)
      035012 MOV #2,-(SP)
      035016 MOV SP,R0
      035020 TRAP C$PNTS
      035022 ADD #6,SP
2096 035026 PRINTS #L210,L2BWR2(R1)
      035026 MOV L2BWR2(R1),-(SP)
      035032 MOV #L210,-(SP)
      035036 MOV #2,-(SP)
      035042 MOV SP,R0
      035044 TRAP C$PNTS
      035046 ADD #6,SP
2097 035052 PRINTS #L211,L2BRD1(R1)
      035052 MOV L2BRD1(R1),-(SP)
      035056 MOV #L211,-(SP)
      035062 MOV #2,-(SP)
      035066 MOV SP,R0
      035070 TRAP C$PNTS
      035072 ADD #6,SP
2098 035076 PRINTS #L212,L2BRD2(R1)
      035076 MOV L2BRD2(R1),-(SP)
      035102 MOV #L212,-(SP)
      035106 MOV #2,-(SP)
      035112 MOV SP,R0
      035114 TRAP C$PNTS
      035116 ADD #6,SP
2099 035122 PRINTS #L213,L2SWR1(R1)
      035122 MOV L2SWR1(R1),-(SP)
      035126 MOV #L213,-(SP)
      035132 MOV #2,-(SP)
      035136 MOV SP,R0
      035140 TRAP C$PNTS
      035142 ADD #6,SP
2100 035146 PRINTS #L214,L2SWR2(R1)
      035146 MOV L2SWR2(R1),-(SP)
      035152 MOV #L214,-(SP)
      035156 MOV #2,-(SP)
      035162 MOV SP,R0

```

	035164	104416			TRAP	C\$PNTS
	035166	062706	000006		ADD	#6,SP
2101	035172				PRINTS	#L215,L2ECC1(R1)
	035172	016146	000040		MOV	L2ECC1(R1),-(SP)
	035176	012746	025357		MOV	#L215,-(SP)
	035202	012746	000002		MOV	#2,-(SP)
	035206	010600			MOV	SP,R0
	035210	104416			TRAP	C\$PNTS
	035212	062706	000006		ADD	#6,SP
2102	035216				PRINTS	#L216,L2ECC2(R1)
	035216	016146	000042		MOV	L2ECC2(R1),-(SP)
	035222	012746	025423		MOV	#L216,-(SP)
	035226	012746	000002		MOV	#2,-(SP)
	035232	010600			MOV	SP,R0
	035234	104416			TRAP	C\$PNTS
	035236	062706	000006		ADD	#6,SP
2103	035242				PRINTS	#L219,L2REP1(R1)
	035242	016146	000050		MOV	L2REP1(R1),-(SP)
	035246	012746	025467		MOV	#L219,-(SP)
	035252	012746	000002		MOV	#2,-(SP)
	035256	010600			MOV	SP,R0
	035260	104416			TRAP	C\$PNTS
	035262	062706	000006		ADD	#6,SP
2104	035266				PRINTS	#L220,L2REP2(R1)
	035266	016146	000052		MOV	L2REP2(R1),-(SP)
	035272	012746	025533		MOV	#L220,-(SP)
	035276	012746	000002		MOV	#2,-(SP)
	035302	010600			MOV	SP,R0
	035304	104416			TRAP	C\$PNTS
	035306	062706	000006		ADD	#6,SP
2105	035312	026127	000124	000000	CMP	124(R1),#0
2106	035320	001002			BNE	1\$
2107	035322	000137	036052		JMP	10\$
2108	035326			1\$:	PRINTS	#L244,<B,66(R1)>,<B,54(R1)>,100(R1)
	035326	016146	000100		MOV	100(R1),-(SP)
	035332	005046			CLR	-(SP)
	035334	156116	000054		BISB	54(R1),(SP)
	035340	005046			CLR	-(SP)
	035342	156116	000066		BISB	66(R1),(SP)
	035346	012746	027130		MOV	#L244,-(SP)
	035352	012746	000004		MOV	#4,-(SP)
	035356	010600			MOV	SP,R0
	035360	104416			TRAP	C\$PNTS
	035362	062706	000012		ADD	#12,SP
2109	035366				PRINTS	#L245,<B,67(R1)>,<B,55(R1)>,102(R1)
	035366	016146	000102		MOV	102(R1),-(SP)
	035372	005046			CLR	-(SP)
	035374	156116	000055		BISB	55(R1),(SP)
	035400	005046			CLR	-(SP)
	035402	156116	000067		BISB	67(R1),(SP)
	035406	012746	027224		MOV	#L245,-(SP)
	035412	012746	000004		MOV	#4,-(SP)
	035416	010600			MOV	SP,R0
	035420	104416			TRAP	C\$PNTS
	035422	062706	000012		ADD	#12,SP
2110	035426				PRINTS	#L246,<B,70(R1)>,<B,56(R1)>,104(R1)
	035426	016146	000104		MOV	104(R1),-(SP)

:ANY DATA COMPARES ?
:YES, LIST THEM
:NO, FINISHED

	035432	005046		CLR	-(SP)
	035434	156116	000056	BISB	56(R1),(SP)
	035440	005046		CLR	-(SP)
	035442	156116	000070	BISB	70(R1),(SP)
	035446	012746	027320	MOV	#L246,-(SP)
	035452	012746	000004	MOV	#4,-(SP)
	035456	010600		MOV	SP,R0
	035460	104416		TRAP	C\$PNTS
	035462	062706	000012	ADD	#12,SP
2111	035466			PRINTS	#L247,<B,71(R1)>,<B,57(R1)>,106(R1)
	035466	016146	000106	MOV	106(R1),-(SP)
	035472	005046		CLR	-(SP)
	035474	156116	000057	BISB	57(R1),(SP)
	035500	005046		CLR	-(SP)
	035502	156116	000071	BISB	71(R1),(SP)
	035506	012746	027414	MOV	#L247,-(SP)
	035512	012746	000004	MOV	#4,-(SP)
	035516	010600		MOV	SP,R0
	035520	104416		TRAP	C\$PNTS
	035522	062706	000012	ADD	#12,SP
2112	035526			PRINTS	#L248,<B,72(R1)>,<B,60(R1)>,110(R1)
	035526	016146	000110	MOV	110(R1),-(SP)
	035532	005046		CLR	-(SP)
	035534	156116	000060	BISB	60(R1),(SP)
	035540	005046		CLR	-(SP)
	035542	156116	000072	BISB	72(R1),(SP)
	035546	012746	027510	MOV	#L248,-(SP)
	035552	012746	000004	MOV	#4,-(SP)
	035556	010600		MOV	SP,R0
	035560	104416		TRAP	C\$PNTS
	035562	062706	000012	ADD	#12,SP
2113	035566			PRINTS	#L249,<B,73(R1)>,<B,61(R1)>,112(R1)
	035566	016146	000112	MOV	112(R1),-(SP)
	035572	005046		CLR	-(SP)
	035574	156116	000061	BISB	61(R1),(SP)
	035600	005046		CLR	-(SP)
	035602	156116	000073	BISB	73(R1),(SP)
	035606	012746	027604	MOV	#L249,-(SP)
	035612	012746	000004	MOV	#4,-(SP)
	035616	010600		MOV	SP,R0
	035620	104416		TRAP	C\$PNTS
	035622	062706	000012	ADD	#12,SP
2114	035626			PRINTS	#L250,<B,74(R1)>,<B,62(R1)>,114(R1)
	035626	016146	000114	MOV	114(R1),-(SP)
	035632	005046		CLR	-(SP)
	035634	156116	000062	BISB	62(R1),(SP)
	035640	005046		CLR	-(SP)
	035642	156116	000074	BISB	74(R1),(SP)
	035646	012746	027700	MOV	#L250,-(SP)
	035652	012746	000004	MOV	#4,-(SP)
	035656	010600		MOV	SP,R0
	035660	104416		TRAP	C\$PNTS
	035662	062706	000012	ADD	#12,SP
2115	035666			PRINTS	#L251,<B,75(R1)>,<B,63(R1)>,116(R1)
	035666	016146	000116	MOV	116(R1),-(SP)
	035672	005046		CLR	-(SP)
	035674	156116	000063	BISB	63(R1),(SP)

	035700	005046		CLR	-(SP)
	035702	156116	000075	BISB	75(R1),(SP)
	035706	012746	027774	MOV	#L251,-(SP)
	035712	012746	000004	MOV	#4,-(SP)
	035716	010600		MOV	SP,R0
	035720	104416		TRAP	C\$PNTS
	035722	062706	000012	ADD	#12,SP
2116	035726			PRINTS	#L252,<B,76(R1)>,<B,64(R1)>,120(R1)
	035726	016146	000120	MOV	120(R1),-(SP)
	035732	005046		CLR	-(SP)
	035734	156116	000064	BISB	64(R1),(SP)
	035740	005046		CLR	-(SP)
	035742	156116	000076	BISB	76(R1),(SP)
	035746	012746	030070	MOV	#L252,-(SP)
	035752	012746	000004	MOV	#4,-(SP)
	035756	010600		MOV	SP,R0
	035760	104416		TRAP	C\$PNTS
	035762	062706	000012	ADD	#12,SP
2117	035766			PRINTS	#L253,<B,77(R1)>,<B,65(R1)>,122(R1)
	035766	016146	000122	MOV	122(R1),-(SP)
	035772	005046		CLR	-(SP)
	035774	156116	000065	BISB	65(R1),(SP)
	036000	005046		CLR	-(SP)
	036002	156116	000077	BISB	77(R1),(SP)
	036006	012746	030164	MOV	#L253,-(SP)
	036012	012746	000004	MOV	#4,-(SP)
	036016	010600		MOV	SP,R0
	036020	104416		TRAP	C\$PNTS
	036022	062706	000012	ADD	#12,SP
2118	036026			PRINTS	#L254,124(R1)
	036026	016146	000124	MOV	124(R1),-(SP)
	036032	012746	030260	MOV	#L254,-(SP)
	036036	012746	000002	MOV	#2,-(SP)
	036042	010600		MOV	SP,R0
	036044	104416		TRAP	C\$PNTS
	036046	062706	000006	ADD	#6,SP
2119	036052			10\$:	ENDRPT
	036052			L10005:	
	036052	104425		TRAP	C\$RPT

2121
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2156
2157
2158
2159
2160
2161
2162

036054
036054
036054 005237 002276
036060
036060
036060 000002

```
.SBTTL GLOBAL SUBROUTINES SECTION
;*****
;*****
;GLOBAL SUBROUTINES SECTION
; THIS SECTION CONTAINS ALL SUBROUTINES AND
; INTERRUPT SERVICE ROUTINES THAT ARE AC-
; CESSSED FROM ANYWHERE IN THE PROGRAM.
;*****
;*****
;*****
;*****
;TRAP4
; THE ADDRESS OF THIS ROUTINE IS LOADED
; INTO VECTOR 4 WHENEVER THE PROGRAM IS
; ATTEMPTING TO ACCESS A PIECE OF HARDWARE
; FOR THE FIRST TIME. IT IS INTENDED TO
; CATCH NON-EXISTENT MEMORY TIMEOUTS IN
; THE EVENT THE HARDWARE IS NOT REALLY PRE-
; SENT OR IS MALFUNCTIONING. IT SIMPLY
; SETS A FLAG, INDICATING THE TRAP OCCURRED.
;*****
;*****
TRAP4: BGNSRV TRAP4
INC TRP4FG ;SET THE FLAG - TRAP OCCURRED
ENDSRV
L10006: RTI
```

```

2167
2168
2169 ;*****
2170 ;*****
2171 ;
2172 ;INTRCV
2173 ; THIS IS THE TK50 INTERRUPT HANDLER USED BY THE PRO-
2174 ; GRAM WHEN INTERRUPTS HAVE BEEN ENABLED. IF THE
2175 ; BRFLAG IS CLEAR, THE ROUTINE SETS A FLAG INDICATING
2176 ; THE EXPECTED INTERRUPT OCCURRED. IF BRFLAG IS SET,
2177 ; IT INDICATES THAT PROCESOR PRIORITY WAS SET TO A
2178 ; LEVEL THAT SHOULD HAVE INHIBITED THE INTERRUPT, SO
2179 ; THE ROUTINE SETS AN ERROR INDICATOR.
2180 ;
2181 ;*****
2182 ;*****
2183
2187 036062 BGNSRV INTRCV
      036062 INTRCV::
2188 ;
2189 ; BIT #BRFLAG,LUNFLG(R4) ;IF NOT PRIORITY LEVEL TESTING
2190 ; BEQ 5$ ; THEN SKIP AROUND
2191 ; MOV #DRPFLG,LUNFLG(R4) ; ELSE SET FAILED BIT
2192 ; BR EXTINT ;RETURN
2193
2194 036062 052764 000002 000014 5$: BIS #INTFLG,LUNFLG(R4) ;SET THE FLAG
2195
2196 036070 EXTINT:
2197 036070 ENDSRV
      036070 L10007:
      036070 RTI
2198

```

```

2203
2204 ;*****
2205 ;*****
2206 ;
2207 ;ILLINT
2208 ; THIS HANDLER ROUTINE'S ADDRESS IS LOADED INTO THE
2209 ; CURRENT UUT'S VECTOR FOR ALL TESTS THAT DO NOT EN-
2210 ; ABLE DEVICE INTERRUPTS.
2211 ;
2212 ;*****
2213 ;*****
2217
2218 036072 BGNSRV ILLINT
      036072 ILLINT::
2219
2220 036072 052764 000001 000014 BIS #DRPFLG,LUNFLG(R4)
2221
2222
2223 036100 ENDSRV
      036100 L10010:
      036100 000002 RTI

```

```

2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2244
2245 036102
2246 036102
      036102 012746 000340
      036106 012746 036054
      036112 012746 000004
      036116 012746 000003
      036122 104437
      036124 062706 000010
2247 036130 005737 177572
2248 036134
      036134 012727 000001
      036140 000000
      036142 013727 002116
      036146 000000
      036150 005367 177772
      036154 001375
      036156 005367 177756
      036162 001367
2249
2250 036164 005737 002276
2251 036170 001026
2252 036172 005237 002274
2253
2254 036176 005737 172516
2255 036202
      036202 012727 000001
      036206 000000
      036210 013727 002116
      036214 000000
      036216 005367 177772
      036222 001375
      036224 005367 177756
      036230 001367
2256
2257 036232 005737 002276
2258 036236 001005
2259 036240 005237 002274
2260 036244 000402
2261
2262 036246 005037 002274
2263
2264 036252
      036252 012700 000004

```

```

;*****
;*****
;KTTEST
;   THIS SUBROUTINE IS USED BY THE INIT CODE TO
;   DETERMINE IF THE MEMORY MANAGEMENT UNIT IS
;   PRESENT.  IF SO, IT RETURNS A FLAG IN THE
;   SET STATE.  OTHERWISE THE FLAG IS CLEAR IN
;   WHICH CASE TEST SEVEN IS BYPASSED.
;*****
;*****
KTTEST:
      SETVEC  #VEC4,#TRAP4,#PRI07      ;SET UP FOR POSSIBLE NXM
      MOV     #PRI07,-(SP)
      MOV     #TRAP4,-(SP)
      MOV     #VEC4,-(SP)
      MOV     #3,-(SP)
      TRAP    C+SVEC
      ADD     #10,SP
      TST     MMUSRO                    ;ARE YOU THERE, MMU?
      DELAY   1                        ;GIVE NXM TIMEOUT A CHANCE
      MOV     #1,(PC)+
      .WORD   0
      MOV     L$DLY,(PC)+
      .WORD   0
      DEC     -6(PC)
      BNE     .-4
      DEC     -22(PC)
      BNE     .-20
      TST     TRP4FG                    ;IF NXM OCCURRED
      BNE     NOKT                      ; THEN NO MMU IS PRESENT
      INC     KTFLAG                    ; ELSE SAY WE FOUND 18 BIT SO FAR
      TST     MMUSR3                    ;NOW LOOK FOR 22 BIT MAPPING
      DELAY   1                        ;GIVE NXM A CHANCE
      MOV     #1,(PC)+
      .WORD   0
      MOV     L$DLY,(PC)+
      .WORD   0
      DEC     -6(PC)
      BNE     .-4
      DEC     -22(PC)
      BNE     .-20
      TST     TRP4FG                    ;IF NXM OCCURRED
      BNE     KTEXT                     ; THEN 18 BIT IS ALL WE'VE GOT
      INC     KTFLAG                    ; ELSE SAY WE'VE GOT 22 BIT
      BR      KTEXT                     ; AND BRANCH AROUND NEXT
NOKT:  CLR     KTFLAG                    ;NO MMU - CLEAR FLAG
KTEXT: CLRVEC #VEC4                    ;RESTORE VECTOR
      MOV     #VEC4,R0

```

2265 036256 104436
2266 036260 005037 002276
2266 036264 000207
2267
2268

TRAP C\$CVEC
CLR TRP4FG
RTS PC

;MORE HOUSEKEEPING

```

2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2292 036266
2293 036266      012746  000000
      036266      012746  036072
      036272      012746  036072
      036276      016446  000004
      036302      012746  000003
      036306      104437
      036310      062706  000010
2294
2295 036314      000207
2296

```

```

;*****
;*****
:RSTVEC
:   THIS ROUTINE IS CALLED FROM VARIOUS PLACES
:   IN THE PROGRAM TO SET THE UUT'S INTERRUPT
:   VECTOR WITH THE ADDRESS OF A HANDLER ROUTINE
:   WHICH WILL CATCH ILLEGAL DEVICE INTERRUPTS,
:   SPECIFICALLY "ILLINT".  INTERRUPT PRIORITY
:   IS SET TO 0.
;*****
;*****

```

```

RSTVEC:
      SETVEC  TKVEC(R4),#ILLINT,#PRI00
      MOV     #PRI00,-(SP)
      MOV     #ILLINT,-(SP)
      MOV     TKVEC(R4),-(SP)
      MOV     #3,-(SP)
      TRAP   C$SVEC
      ADD    #10,SP

      RTS    PC

```



```

2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2324 036316
2325 036316 032764 000004 000014
2326 036324 001014
2327 036326
      036326 012746 000000
      036332 012746 036062
      036336 016446 000004
      036342 012746 000003
      036346 104437
      036350 062706 000010
2328 036354 000413
2329
2330 036356
      036356 012746 000340
      036362 012746 036062
      036366 016446 000004
      036372 012746 000003
      036376 104437
      036400 062706 000010
2331
2332 036404 000207
2333
2334

```

```

:*****
:*****
:VECTOR
:
:   THIS ROUTINE IS CALLED FROM VARIOUS PLACES
:   IN THE PROGRAM TO SET THE UUT'S VECTOR WITH
:   THE ADDRESS OF A HANDLER ROUTINE WHEN DEVICE
:   INTERRUPTS HAVE BEEN ENABLED.  THE ROUTINE HAS
:   TWO MODES OF OPERATION:  WHEN BRFLAG IS CLEAR,
:   PROCESSOR PRIORITY IS SET TO ZERO, ALLOWING
:   DEVICE INTERRUPTS.  IF BRFLAG IS SET, PRIORITY
:   IS SET TO 7.  IF AN INTERRUPT OCCURS IN THIS
:   CASE, AN ERROR IS RETURNED BY THE HANDLER
:   ROUTINE, "INTRCV".
:*****
:*****

```

```

VECTOR::
BIT      @BRFLAG,LUNFLG(R4)      ;IF FLAG IS SET
BNE      5$                      ; THEN SKIP TO SECOND HALF
SETVEC   TKVEC(R4),@INTRCV,@PRI00 ;ELSE LOW PRIORITY
MOV      @PRI00,-(SP)
MOV      @INTRCV,-(SP)
MOV      TKVEC(R4),-(SP)
MOV      @3,-(SP)
TRAP     C$SVEC
ADD      @10,SP
BR       EXTVEC                   ;RETURN

5$:      SETVEC   TKVEC(R4),@INTRCV,@PRI07 ;HIGH PRIORITY
MOV      @PRI07,-(SP)
MOV      @INTRCV,-(SP)
MOV      TKVEC(R4),-(SP)
MOV      @3,-(SP)
TRAP     C$SVEC
ADD      @10,SP

EXTVEC:  RTS      PC

```

2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371

```

;*****
;*****
;PDELAY
; THIS ROUTINE IS USED THROUGHOUT THE PROGRAM TO PROVIDE
; A VARIABLE AMOUNT OF DELAY TIME. THE DELAY WILL BE
; INSTRUCTION EXECUTION TIME DEPENDENT. TWO VALUES MUST
; BE LOADED BY MAINLINE CODE PRIOR TO CALLING PDELAY:
; "INNER" AND "OUTER". IF SUFFICIENT CALLS TO PDELAY ARE
; MADE SUCH THAT THE OUTER COUNT IS EXHAUSTED, THE ROUTINE
; RETURNS "TOUT" EQUAL TO 1, INDICATING TIMEOUT HAS OCCURRED.
; "INNER" SHOULD BE RE-LOADED BY MAINLINE CODE, PRIOR TO
; CALL TO PDELAY WITHIN A TIMING LOOP.
;*****
;*****

```

```

PDELAY: :
        CLR      TOUT      ;CLEAR TIMEOUT INDICATOR
        DEC      INNER     ;IF COUNT NOT EXHAUSTED
        BNE      PDELAY    ; THEN KEEP LOOPING
        DEC      OUTER     ;IF MAJOR COUNT NOT 0
        BNE      PDLYEX   ; THEN LEAVE WITH STATUS = OK
        INC      TOUT
        PDLYEX: RTS      PC ; ELSE SET TIMEOUT

```

```

036406 005037 002330
036406 005337 002324
036416 001373
036420 005337 002326
036424 001002
036426 005237 002330
036432 000207

```

2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413

```

:*****
:*****
:STEP1
:   THIS SUBROUTINE IS RESPONSIBLE FOR PERFORMING
:   STEP 1 OF THE UQ-PORT INIT SEQUENCE. SPECIFI-
:   CALLY, IT WILL INITIALIZE THE UUT BY WRITING
:   TO ITS IP REGISTER. AFTER A BRIEF DELAY, IT
:   WILL READ THE SA REGISTER TO INSURE THAT THE
:   STEP 1 BIT IS SET AND THE ERROR BIT IS CLEAR.
:   IT WILL THEN WRITE THE FIRST LOCATION OF THE
:   STEP TABLE (SET UP BY MAINLINE CODE) TO THE
:   UUT'S SA REG. IF ALL STEPS COMPLETE SUCCESS-
:   FULLY THE ROUTINE RETURNS "STEPST" CLEARED;
:   OTHERWISE "STEPST" IS RETURNED INDICATING A
:   FAILURE OCCURRED.
:*****
:*****

```

```

STEP1::
CLR      STEPST      ;CLEAR THE STATUS INDICATOR
MOV      #0,@TKIP(R4) ;INIT THE UUT
MOV      #1,(PC)+
        .WORD      0
MOV      L$DLY,(PC)+
        .WORD      0
DEC      -6(PC)
BNE      -.4
DEC      -22(PC)
BNE      .-20

MOV      @TKSA(R4),TKSASV(R4) ;GET THE SA REG CONTENTS
CMP      #B.S1!B.QB!B.DI!B.OO!B.MP,TKSASV(R4)

        ;IF ALL THE RIGHT BITS AREN'T SET
        ; THEN TAKE ERROR EXIT
BNE      STP1ER      ;
MOV      STPTBL,@TKSA(R4); ELSE WRITE HOST'S STEP 1 RESPONSE
BR       STP1EX      ; AND LEAVE SHOWING SUCCESS

STP1ER: INC      STEPST      ;SET ERROR INDICATOR

STP1EX: RTS      PC

```

```

2418 ;:*****
2419 ;:*****
2420 ;:
2421 ;BAKPAT
2422 ;   THIS SUBROUTINE WILL FILL THE COMMUNICATION WITH AN
2423 ;   ALL 1'S DATA PATTERN.  THE LENGTH OF THE AREA IN USE
2424 ;   BY THE CURRENT TEST IS CONTAINED IN "CMARLG".
2425 ;:
2426 ;:*****
2427 ;:*****
2431
2432 036532 BAKPAT::
2433 036532 012702 060000      MOV     #COMMBF,R2          ;STARTING ADDRESS OF COMM AREA
2434                                     ; -20 WORDS
2435 036536 012703 000024      MOV     #20.,R3           ;BUFFER LENGTH IN FRONT OF AREA
2436 036542 006303             ASL     R3                ;MULTIPLIED BY 2
2437 036544 063703 002306      ADD     CMARLG,R3        ;ADD COMM AREA LENGTH USED
2438 036550 012722 177777      1$:   MOV     #-1,(R2)+   ;WRITE THE DATA
2439 036554 005303             DEC     R3                ;IF NOT DONE YET
2440 036556 001374             BNE     1$                ; THEN DO IT AGAIN
2441
2442 036560 000207             RTS     PC
2443
    
```

```

2448
2449
2450 ;*****
2451 ;*****
2452 ;
2453 ;CHKCOM
2454 ; THIS ROUTINE IS CALLED BY TESTS DOING THE PURGE/POLL
2455 ; CHECK. IT IS USED TO VERIFY THAT THE PORT LEFT THE
2456 ; COMMUNICATIONS AREA CLEARED. ADDITIONALLY, IT CHECKS
2457 ; THE 20 WORDS PRECEDING AND SUCCEEDING THE COMM AREA
2458 ; TO MAKE SURE THE PORT DIDN'T GO OUTSIDE THE COMM AREA.
2459 ;
2460 ;*****
2461 ;*****
2465
2466 036562
2467 036562 012701 177777
2468 036566 012702 060000
2469 036572 012703 000022
2470 036576 020122
2471 036600 001022
2472 036602 005303
2473 036604 001374
2474
2475 036606 005001
2476 036610 013703 002306
2477 036614 005722
2478 036616 001013
2479 036620 005303
2480 036622 001374
2481
2482 036624 012701 177777
2483 036630 012703 000024
2484 036634 020122
2485 036636 001003
2486 036640 005303
2487 036642 001374
2488 036644 000422
2489
2490 036646 162702 000002
2491 036652 022737 000010 002114
2492 036660 001405
2493 036662
036662 104455
036664 000001
036666 030677
036670 033022
2494 036672 000404
2495
2496 036674
036674 104455
036676 000002
036700 030726
036702 033046
2497
2498 036704
036704 013700 002312

```

```

CHKCOM::
      MOV     #-1,R1           ;TEST DATA
      MOV     #COMMBF,R2      ;STARTING ADDRESS
      MOV     #18.,R3         ;FIRST COUNT
1$:   CMP     R1,(R2)+        ;IF NOT ALL 1'S
      BNE     15$             ; THEN GO REPORT ERROR
      DEC     R3              ;IF NOT ALL DONE
      BNE     1$              ; THEN GO CHECK ANOTHER

      CLR     R1              ;TEST DATA FOR PRINTOUT
      MOV     CMARLG,R3       ;SET UP COUNTER FOR COMM AREA
5$:   TST     (R2)+           ;IF NOT 0
      BNE     15$             ; THEN GO REPORT ERROR
      DEC     R3              ;IF NOT ALL DONE
      BNE     5$              ; THEN GO CHECK ANOTHER

      MOV     #-1,R1         ;TEST DATA FOR PRINTOUT
      MOV     #20.,R3        ;SET UP COUNTER FOR POST COMM AREA
10$:  CMP     R1,(R2)+        ;IF NOT ALL 1'S
      BNE     15$             ; THEN GO REPORT ERROR
      DEC     R3              ;IF NOT ALL DONE
      BNE     10$            ; THEN GO CHECK ANOTHER
      BR      CKCMEX         ; ELSE RETURN

15$:  SUB     #2,R2           ;ADJUST ADDRESS FOR PRINTOUT
      CMP     #8.,L$TEST     ;IF IN TEST 8
      BEQ     20$            ; THEN DO ALTERNATE PRINTOUT
      ERRDF  1,MSG14,PRIPAD ;"PURGE/POLL TEST FAILED"
      TRAP   C$ERDF
      .WORD  1
      .WORD  MSG14
      .WORD  PRIPAD
      BR     25$             ;COMMON EXIT

20$:  ERRDF  2,MSG15,PRIVAD ;"EXTENDED ADDRESS TEST FAILED"
      TRAP   C$ERDF
      .WORD  2
      .WORD  MSG15
      .WORD  PRIVAD

25$:  DODU   LOGUNT
      MOV     LOGUNT,R0

```

2499	036710	104451	TRAP	C\$DODU
2500	036712	000207	CKCMEX: RTS	PC
2501				

```

2506
2507
2508 ;:*****
2509 ;:*****
2510
2511 ;INTMMU
2512 ;
2513 ; THIS SUBROUTINE IS CALLED FROM TEST 8 TO INITIALIZE
2514 ; MEMORY MANAGEMENT REGISTERS. ALL PAR'S EXCEPT ONE
2515 ; ARE SET UP TO MAP VIRTUAL ADDRESSES INTO THE LOWEST
2516 ; 32K OF PHYSICAL MEMORY. KPAR7 IS SET UP TO MAP TO
2517 ; THE I/O PAGE. THE PAR REGISTER THAT CORRESPONDS TO
2518 ; THE VIRTUAL ADDRESS OF THE COMMUNICATION AREA IS SET
2519 ; UP TO POINT TO THE SECOND 32K OF PHYSICAL MEMORY.
2520 ; ALL PDR'S ARE INITIALIZED TO THE SAME VALUE; NAMELY,
2521 ; UPWARD EXPANDABLE, READ/WRITE ACCESS ENABLED, AND THE
2522 ; FULL 8KBYTE PAGE IS ACCESSIBLE.
2523 ;:*****
2524 ;:*****
2528
2529 INTMMU::
2530 036714 012703 172300 MOV #KPDRO,R3 ;START OF PDR ADDRESS RANGE
2531 036720 012702 172340 MOV #KPARO,R2 ;START OF PAR ADDRESS RANGE
2532 036724 005001 CLR R1 ;STARTING RELOCATION VALUE
2533
2534 036726 010122 1$: MOV R1,(R2)+ ;LOAD RELOCATION VALUE
2535 036730 012723 077406 MOV #77406,(R3)+ ;LOAD PDR
2536 036734 062701 000200 ADD #200,R1 ;ADJUST RELOCATION VALUE
2537 036740 022701 002000 CMP #2000,R1 ;IF NOT AT THE END
2538 036744 001370 BNE 1$ ; THEN DO ANOTHER ONE
2539
2540 036746 010137 172346 MOV R1,KPAR3 ; ELSE SET THIS REG TO NEXT 32K
2541 036752 012737 007600 172356 MOV #7600,KPAR7 ;18 BIT I/O PAGE
2542 036760 032737 000002 002274 BIT #BIT1,KTFLAG ;IF 22-BIT BUS NOT AVAILABLE
2543 036766 001406 BEQ 2$ ; THEN GO TURN MMU ON
2544 036770 012737 177600 172356 MOV #177600,KPAR7 ; ELSE SET 22 BIT I/O PAGE
2545 036776 012737 000020 172516 MOV #MM22ON,MMUSR3 ; AND ENABLE 22 BIT MAPPING
2546
2547 037004 012737 000001 177572 2$: MOV #MMON,MMUSRO ;TURN ON THE WHOLE THING
2548 037012 000207 RTS PC
2549
2550
2551 037014 PRTINT::
2552 037014 010174 000000 MOV R1,@TKIP(R4) ;INITIALIZE THE DRIVE
2553 037020 012703 037236 MOV #INTTBL,R3 ;PUT THE TABLE ADDRESS INTO R3
2554 037024 012701 004000 MOV #S1,R1 ;SET UP TO BEGIN AT STEP 1
2555 037030 005037 002316 CLR INISTP ;CLEAR THE STEP TRACKER
2556 037034 012737 000030 003024 LOOP: MOV #24.,CNTHI ;SET UP THE TIME OUT COUNTER
2557 037042 005002 CLR R2 ;CLEAR R2
2558 037044 005202 ILOOP: INC R2 ;INCREMENT HI TIME OUT VALUE ?
2559 037046 001016 BNE 2$ ;IF NOT, BRANCH
2560 037050 005337 003024 DEC CNTHI ;ELSE, DECREMENT LO TIMEOUT
2561 037054 001013 BNE 2$ ;BRANCH IF NO TIME OUT
2562 037056 017464 000002 000012 MOV @TKSA(R4),TKSASV(R4) ;SAVE THE SA FOR THE ERROR PRINTOUT
2563 037064 104455 ERRDF 51.,WRER1,WRINTO ;PRINT PORT INIT FAILURE
037066 000063 TRAP C$ERDF
WORD 51

```

```

037070 032347 .WORD WRER1
037072 033176 .WORD WRINTO
2564 037074 013700 002312 DODU LOGUNT ;DROP THE UNIT
037074 104451 MOV LOGUNT,R0
037100 104451 TRAP C$DODU
2565 037102 000454 BR 100$ ;EXIT ROUTINE
2566 037104 037401 000002 2$: BIT @TKSA(R4),R1 ;TEST FOR STEP BIT FROM DRIVE
2567 037110 001755 BEQ ILOOP ;LOOP UNTIL SOMETHING SETS
2568 037112 032774 100000 000002 BIT #ERR,@TKSA(R4) ;CHECK FOR ERROR
2569 037120 001413 BEQ 3$ ;NO ERROR, KEEP GOING
2570 037122 017464 000002 000012 MOV @TKSA(R4),TKSASV(R4) ;SAVE THE SA CONTENTS
2571 037130 ERRDF 52.,WRER2,WRPRT ;PRINT ERROR
037130 104455 TRAP C$ERDF
037132 000064 .WORD 52
037134 032401 .WORD WRER2
037136 033222 .WORD WRPRT
2572 037140 DODU LOGUNT ;DROP THE UNIT
037140 013700 002312 MOV LOGUNT,R0
037144 104451 TRAP C$DODU
2573 037146 000432 BR 100$ ;EXIT ROUTINE
2574 037150 005237 002316 3$: INC INISTP ;INCREMENT THE STEP TRACKER
2575 037154 012374 000002 MOV (R3)+,@TKSA(R4) ;WRITE WORD FROM TABLE TO CONTROLLER
2576 037160 006301 ASL R1 ;SHIFT TO NEXT STEP
2577 037162 100324 BPL ILOOP ;IF NOT AT LAST STEP LOOP
2578 037164 012702 003004 MOV #RSPRNG,R2 ;PUT THE RESPONSE DESCRIPTOR ADDR IN R2
2579 037170 012703 002574 MOV #RSPBUF,R3 ;PUT THE RESPONSE BUFFER ADDRESS IN R3
2580 037174 010322 5$: MOV R3,(R2)+ ;PUT THE BUFF ADDR IN THE DESCRIPTOR
2581 037176 012722 100000 MOV *#OWN,(R2)+ ;SET THE DESCRIPTOR TO THE CONTROLLER
2582 037202 062703 000104 ADD #RSPSTP,R3 ;STEP TO THE NEXT BUFFER SLOT
2583 037206 022703 003004 CMP #RSPEND,R3 ;ARE WE AT THE END OF THE BUFFER ?
2584 037212 001370 BNE 5$ ;NO, KEEP GOING
2585 037214 012737 003004 023040 MOV #RSPRNG,RSPSAV ;SET UP TO USE FIRST RESPONSE BUFFER
2586 037222 012737 003014 023036 MOV #CMDRNG,CMDSAV ;SET UP TO USE FIRST COMMAND BUFFER
2587 037230 005037 003032 CLR CMDREF ;SET THE COMMAND REFERENCE # TO 0
2588 037234 000207 100$: RTS PC ;RETURN
2589
2590 ;INIT DATA TABLE
2591 037236 104400 INTTBL: .WORD 104400
2592 037240 003004 .WORD RSPRNG
2593 037242 000000 .WORD 0
2594 037244 000001 .WORD GO

```



```

2596 037246          CLSDRV::
2597 037246 005237 003032 5$: INC CMDREF ;ADD 1 TO THE COMMAND REFERENCE NUMBER
2598 037252 004737 037604 JSR PC,PRTDRV ;GO SEND THE COMMAKD
2599 037256 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
2600 037254 001146 BNE 100$ ;GET OUT IF NOT AVAILABLE
2601 037266 004737 037746 JSR PC,CDRECV ;GO CHECK FOR ANY NEW RESPONSES
2602 037272 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
2603 037300 001140 BNE 100$ ;GET OUT IF NOT AVAILABLE
2604 037302 012702 024035 MOV #WR2,R2 ;R2 GETS ERROR MESSAGE ADDRESS
2605 037306 005763 000012 TST P.STS(R3) ;WAS STATUS "NORMAL"?
2606 037312 001463 BEQ 15$ ;YES - BRANCH
2607 037314 032764 000010 000014 7$: BIT #DUPFLG,LUNFLG(R4) ;IS IT A DUP COMMAND?
2608 037322 001047 BNE 12$ ;YES, PRINT DUP ERROR
2609 037324 022705 002406 CMP #ONLINE,R5 ;IS IT AN ONLINE COMMAND
2610 037330 001024 BNE 10$ ;NO, GO PRINT ERROR
2611 037332 122763 000003 000012 CMPB #3,P.STS(R3) ;WAS IT A UNIT UNKNOWN ERROR?
2612 037340 001003 BNE 8$ ;NO, CONTINUE
2613 037342 012702 024214 MOV #WR7,R2 ;R2 GETS ERROR MESSAGE ADDRESS
2614 037346 000415 BR 10$ ;PRINT ERROR MESSAGE
2615 037350 122763 000043 000012 8$: CMPB #43,P.STS(R3) ;WAS IT A NO MEDIA MOUNTED ERROR?
2616 037356 001003 BNE 9$ ;NO, CONTINUE
2617 037360 012702 024265 MOV #WR8,R2 ;R2 GETS ERROR MESSAGE ADDRESS
2618 037364 000406 BR 10$ ;PRINT ERROR MESSAGE
2619 037366 122763 000103 000012 9$: CMPB #103,P.STS(R3) ;WAS IT A UNIT INOPERATIVE ERROR?
2620 037374 001012 BNE 11$ ;NO, MUST BE INVALID STATUS
2621 037376 012702 024265 MOV #WR8,R2 ;R2 GETS ERROR MESSAGE ADDRESS
2622 037402 104$: ERRDF 57.,WRER3,WRCMDE ;GET READY TO PRINT FAILURE
      037402 104455 TRAP C$ERDF
      037404 000071 .WORD 57
      037406 032422 .WORD WRER3
      037410 033256 .WORD WRCMDE
2623 037412 DODU LOGUNT ;DROP THE UNIT
      037412 013700 002312 MOV LOGUNT,R0
      037416 104451 TRAP C$DODU
2624 037420 000470 BR 100$ ;GET OUT ON ERROR
2625 037422 11$: ERRDF 58.,WRER7,WRCMDE ;GET READY TO PRINT FAILURE
      037422 104455 TRAP C$ERDF
      037424 000072 .WORD 58
      037426 032546 .WORD WRER7
      037430 033256 .WORD WRCMDE
2626 037432 DODU LOGUNT ;DROP THE UNIT
      037432 013700 002312 MOV LOGUNT,R0
      037436 104451 TRAP C$DODU
2627 037440 000460 BR 100$ ;GET OUT ON ERROR
2628 037442 12$: ERRDF 59.,WRER8,WRCMDE ;GET READY TO PRINT FAILURE
      037442 104455 TRAP C$ERDF
      037444 000073 .WORD 59
      037446 032614 .WORD WRER8
      037450 033256 .WORD WRCMDE
2629 037452 DODU LOGUNT ;DROP THE UNIT
      037452 013700 002312 MOV LOGUNT,R0
      037456 104451 TRAP C$DODU
2630 037460 000450 BR 100$ ;GET OUT ON ERROR
2631 037462 022705 002456 15$: CMP #GDUST,R5 ;WAS IT A GET DUST STATUS COMMAND?
2632 037466 001035 BNE 20$ ;NO, CONTINUE
2633 037470 126327 000014 000113 CMPB P.EXT1(R3),#113 ;1ST BYTE OF PROGRAM EXTENSION = "K"?
2634 037476 001341 BNE 10$ ;NO, ERROR
    
```

2635	037500	126327	000015	000065		CMPB	P.EXT2(R3),#65		;2ND BYTE OF PROGRAM EXTENSION = "5" ?
2636	037506	001335				BNE	10\$;NO, ERROR
2637	037510	126327	000016	000060		CMPB	P.EXT3(R3),#60		;3RD BYTE OF PROGRAM EXTENSION = "0" ?
2638	037516	001331				BNE	10\$;NO, ERROR
2639	037520	126327	000017	000007		CMPB	DUSTFL(R3),#7		;FLAGS = 7 ?
2640	037526	001325				BNE	10\$;NO, ERROR
2641	037530	026327	000020	000000		CMP	P.IND1(R3),#0		;1ST WORD OF PROGRESS INDICATOR = 0 ?
2642	037536	001321				BNE	10\$;NO, ERROR
2643	037540	026327	000022	000000		CMP	P.IND2(R3),#0		;2ND WORD OF PROGRESS INDICATOR = 0 ?
2644	037546	001315				BNE	10\$;NO, ERROR
2645	037550	026327	000024	000012		CMP	TIMOUT(R3),#12		;TIMEOUT VALUE = 12 ?
2646	037556	001311				BNE	10\$;NO, ERROR
2647	037560	000405				BR	30\$;NO ERRORS, RETURN
2648	037562	022705	002524		20\$:	CMP	#RCVDAT,R5		;WAS IT A RECEIVE DATA COMMAND?
2649	037566	001002				BNE	30\$;NO, CONTINUE
2650	037570	004737	040200			JSR	PC,L2DATA		;YES, GO READ DIAGL2 PROGRAM STATUS
2651	037574	005337	002272		30\$:	DEC	IIRCNT		;SUBTRACT 1 FROM TIMES TO DO
2652	037600	001222				BNE	5\$;GO SEND IT AGAIN
2653	037602	000207			100\$:	RTS	PC		;RETURN
2654									
2655	037604					PRTDRV::			
2656	037604	013701	023036			MOV	CMDSAV,R1		;SET UP COMMAND RING POINTER
2657	037610	013765	003032	000000		MOV	CMDREF,P.CRF(R5)		;PUT COMMAND REFERENCE # INTO PACKET
2658	037616	112765	000002	177776		MOVB	#2,CRD(R5)		;PUT THE CREDIT LIMIT INTO THE PACKET
2659	037624	032764	000010	000014		BIT	#DUPFLG,LUNFLG(R4)		;IS IT A DUP COMMAND?
2660	037632	001404				BEQ	1\$;NO, CONNECTION TYPE = 1
2661	037634	112765	000002	177777		MOVB	#2,CONID(R5)		;YES,CONNECTION TYPE = 2
2662	037642	000403				BR	2\$;CONTINUE
2663	037644	112765	000001	177777	1\$:	MOVB	#1,CONID(R5)		;PUT THE CONNECTION TYPE INTO THE PACKET
2664	037652	010511			2\$:	MOV	R5,(R1)		;PUT THE PACKET ADDRESS INTO THE DESCRIPTOR
2665	037654	012761	100000	000002		MOV	#OWN,HIADDR(R1)		;SET THE OWNERSHIP BIT OF THE DESCRIPTOR
2666	037662	005774	000000			TST	@TKIP(R4)		;READ THE IP REGISTER
2667	037666	005774	000002			TST	@TKSA(R4)		;READ THE SA REGISTER
2668	037672	100013				BPL	10\$;BRANCH IF NO ERRORS
2669	037674	017464	000002	000012		MOV	@TKSA(R4),TKSASV(R4)		;SAVE THE SA FOR THE ERROR PRINTOUT
2670	037702					ERRDF	53.,WRER4,WRPTE		;PRINT PORT DETECTED ERROR
	037702	104455				TRAP	C\$ERDF		
	037704	000065				.WORD	53		
	037706	032450				.WORD	WRER4		
	037710	033222				.WORD	WRPTE		
2671	037712					DODU	LOGUNT		;DROP THE UNIT
	037712	013700	002312			MOV	LOGUNT,R0		
	037716	104451				TRAP	C\$DODU		
2672	037720	000411				BR	100\$;GET OUT
2673	037722	062701	000004		10\$:	ADD	#RNGSTP,R1		;ADJUST RESPONSE POINTER FOR NEXT TIME
2674	037726	022701	003024			CMP	#DSCEND,R1		;ARE WE AT THE END ?
2675	037732	001002				BNE	15\$;NO, GET OUT
2676	037734	012701	003014			MOV	#CMDRNG,R1		;SET R1 TO TOP BUFFER
2677	037740	010137	023036		15\$:	MOV	R1,CMDSAV		;SAVE THE COMMAND RING LOCATION
2678	037744	000207			100\$:	RTS	PC		;RETURN

```

2680 037746          CDRECV:
2681 037746 004737 040054 1$: JSR    PC,PDRECV          ;CALL PORT DRIVER RECEIVE
2682 037752 032764 000001 000014 BIT    #DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
2683 037760 001034          BNE    100$                ;GET OUT IF NOT AVAILABLE
2684 037762 011103          MOV    (R1),R3              ;SET UP RESPONCE BUFFER POINTER
2685 037764 026365 000000 000000 CMP    P.CRF(R3),P.CRF(R5)    ;IS THIS THE COMMAND THAT IS EXPECTED ?
2686 037772 001015          BNE    10$                 ;GET OUT IF WRONG RESPONCE
2687 037774 012761 100000 000002 MOV    #OWN,HIADDR(R1)       ;GIVE THE CONTROLLER THE RING BACK
2688 040002 062701 000004          ADD    #RNGSTP,R1          ;ADJUST RESPONCE POINTER FOR NEXT TIME
2689 040006 022701 003014          CMP    #CMDRNG,R1          ;ARE WE AT THE END ?
2690 040012 001002          BNE    15$                 ;NO, GET OUT
2691 040014 012701 003004          MOV    #RSPRNG,R1          ;SET R1 TO TOP BUFFER
2692 040020 010137 023040 15$: MOV    R1,RSPSAV          ;SAVE THE POINTER FOR NEXT TIME
2693 040024 000412          BR    100$                ;PRINT CONTROLLER ERROR
2694 040026 012737 032640 002310 10$: MOV    #CTRL,FRUIS          ;PRINT COMMAND OUT OF SEQUENCE ERROR
2695 040034          ERRDF 56.,WRER5,WRSEQE
2696 040044          TRAP  C$ERDF
2697 040052          .WORD 56
2698          .WORD WRER5
2699          .WORD WRSEQE
2700          DODU LOGUNT          ;DROP THE UNIT
2701 040054          MOV    LOGUNT,RO
2702 040054 013701 023040          TRAP  C$DODU
2703 040060 012737 000225 003024 1$: RTS    PC          ;RETURN
2704 040066 005002          .WORD 56
2705 040070 005202          .WORD WRER5
2706 040072 001013          .WORD WRSEQE
2707 040074 005337 003024          DODU LOGUNT          ;DROP THE UNIT
2708 040100 001010          MOV    LOGUNT,RO
2709 040102          TRAP  C$DODU
2710 040112          BR    100$                ;GO DROP THE UNIT
2711 040120 000426          DODU LOGUNT
2712 040122 017464 000002 000012 10$: MOV    LOGUNT,RO
2713 040130 032764 100000 000012 5$: TRAP  C$DODU
2714 040136 001413          BR    100$                ;GET OUT ON ERROR
2715 040140 012737 032640 002310 10$: MOV    @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
2716 040146          BIT    #BIT15,TKSASV(R4) ;CHECK FOR SA ERROR
2717 040156          BEQ    20$                ;BRANCH IF NO ERROR
2718 040164          MOV    #CTRL,FRUIS          ;LOAD FAILING FRU
2719          ERRDF 55.,EMSG9,WRP RTE ;PRINT SA CONTENTS IN ERROR MESSAGE
2720          TRAP  C$ERDF
2721          .WORD 55
2722          .WORD EMSG9
2723          .WORD WRP RTE
2724          DODU LOGUNT          ;GO DROP THE UNIT
2725          MOV    LOGUNT,RO
2726          TRAP  C$DODU
2727          BR    100$                ;GET OUT ON ERROR
    
```

```

2719 040166 032761 100000 000002 20$: BIT #OWN,HIADDR(R1) ;IS THE SLOT SET TO US ?
2720 040174 001335 BNE 5$ ;KEEP GOING TILL TIMEOUT OR SUCCESS
2721 040176 000207 100$: RTS PC ;RETURN
2722
2723
2724
2725
2726 040200 L2DATA::
2727 040200 010146 MOV R1,-(SP) ;SAVE REGISTERS
2728 040202 010246 MOV R2,-(SP)
2729 040204 010346 MOV R3,-(SP)
2730 040206 012701 060000 MOV #RDBUF,R1 ;GET START ADDRESS OF RECEIVE DATA
2731 040212 121127 000001 CMPB (R1),#1 ;ANY NEW INFORMATION ?
2732 040216 001162 BNE EXIT ;NO, RETURN
2733 040220 026127 000002 000000 CMP L2STA(R1),#0 ;YES, LOCAL PROGRAM FINISHED ?
2734 040226 001552 BEQ GOABO ;YES, SET ABORT FLAG
2735 040230 016103 000002 MOV L2STA(R1),R3 ;GET LEVEL 2 PROGRAM STATUS
2736 040234 042703 177740 BIC #177740,R3 ;MASK OFF UPPER 11 BITS OF STATUS
2737 040240 022703 000037 CMP #37,R3 ;L2 MESSAGE IF 1ST 5 BITS SET
2738 040244 001113 BNE MSCPER ;TMSCP COMMAND ERROR OTHERWISE
2739 040246 016103 000002 MOV L2STA(R1),R3 ;GET LEVEL 2 PROGRAM STATUS
2740 040252 012737 032640 002310 MOV #CTRL,FRUIS ;NO, SET FAILING FRU TO CONTROLLER
2741 040260 042703 000037 BIC #000037,R3 ;MASK OFF UNUSED BITS OF STATUS
2742 040264 006203 ASR R3
2743 040266 006203 ASR R3
2744 040270 006203 ASR R3
2745 040272 006203 ASR R3
2746 040274 006203 ASR R3
2747
2748 040276 022703 000001 CMP #1.,R3 ;IS IT A DATA COMPARE ERROR ?
2749 040302 001006 BNE 5$ ;NO TRY AGAIN
2750 040304 012702 023312 MOV #DCERR,R2 ;SET UP TO PRINT ERROR
2751 040310 012737 032655 002310 MOV #DRVE,FRUIS ;SET FRU CALLOUT TO DRIVE
2752 040316 000454 BR 40$ ;GO PRINT IT
2753
2754 040320 022703 000002 5$: CMP #2.,R3 ;IS IT A CONTROLLER ERROR ?
2755 040324 001003 BNE 10$ ;NO TRY AGAIN
2756 040326 012702 023322 MOV #CNTERR,R2 ;SET UP TO PRINT ERROR
2757 040332 000446 BR 40$ ;GO PRINT IT
2758
2759 040334 022703 000012 10$: CMP #10.,R3 ;IS IT AN INVAILID STATUS ERROR ?
2760 040340 001003 BNE 15$ ;NO TRY AGAIN
2761 040342 012702 023332 MOV #INVSTA,R2 ;SET UP TO PRINT ERROR
2762 040346 000440 BR 40$ ;GO PRINT IT
2763
2764 040350 022703 000024 15$: CMP #20.,R3 ;IS IT A BAD PATTERN NUMBER ERROR ?
2765 040354 001003 BNE 20$ ;NO TRY AGAIN
2766 040356 012702 023342 MOV #BPNERR,R2 ;SET UP TO PRINT ERROR
2767 040362 000432 BR 40$ ;GO PRINT IT
2768
2769 040364 022703 000025 20$: CMP #21.,R3 ;IS IT A RESPONSE ADDRESS ERROR ?
2770 040370 001003 BNE 25$ ;NO TRY AGAIN
2771 040372 012702 023352 MOV #RSPADD,R2 ;SET UP TO PRINT ERROR
2772 040376 000424 BR 40$ ;GO PRINT IT
2773
2774 040400 022703 000026 25$: CMP #22.,R3 ;IS IT A HOST BUFFER ADDRESS ERROR ?
2775 040404 001003 BNE 30$ ;NO TRY AGAIN
    
```

```

2776 040406 012702 023362      MOV    #HBFADD,R2      ;SET UP TO PRINT ERROR
2777 040412 000416              BR     40$            ;GO PRINT IT
2778
2779 040414 022703 000027      30$:  CMP    #23.,R3    ;IS IT A UNKNOWN ERROR LOG RECIEVED ?
2780 040420 001003              BNE   35$            ;NO TRY AGAIN
2781 040422 012702 023372      MOV    #UNERLG,R2     ;SET UP TO PRINT ERROR
2782 040426 000410              BR     40$            ;GO PRINT IT
2783
2784 040430 022703 000030      35$:  CMP    #24.,R3    ;IS IT A RESPONSE TIME OUT ERROR ?
2785 040434 001003              BNE   36$            ;NO TRY AGAIN
2786 040436 012702 023402      MOV    #RSPT0,R2     ;SET UP TO PRINT ERROR
2787 040442 000402              BR     40$            ;GO PRINT IT
2788
2789 040444 012702 023332      36$:  MOV    #INVSTA,R2  ;SET UP TO PRINT ERROR
2790
2791 040450 012237 023052      40$:  MOV    (R2)+,ERRTYP ;LOAD ERROR TYPE
2792 040454 012237 023054      MOV    (R2)+,ERRNBR  ;LOAD ERROR NUMBER
2793 040460 012237 023056      MOV    (R2)+,ERRMSG  ;LOAD ERROR MESSAGE ADDRESS
2794 040464 012237 023060      MOV    (R2)+,ERRBLK  ;LOAD ERROR SUBROUTINE ADDRESS
2795 040470              ERROR              ;CALL "ERROR" MACRO
2796 040470 104460              TRAP   C$ERROR
2796 040472 000430              BR     GOABO         ;ABORT MICRODIAGNOSTIC
2797 040474 022703 000012      MSCPER: CMP   #12,R3   ;IS IT A CONTROLLER ERROR ?
2798 040500 001406              BEQ   10$            ;YES, SET FAILING FRU TO CONTROLLER
2799 040502 022703 000014      CMP   #14,R3        ;IS IT A FORMATTER ERROR ?
2800 040506 001403              BEQ   10$            ;YES, LEAVE FAILING FRU AS CONTRLLOER
2801 040510 012737 032655 002310 5$:  MOV    #DRVE,FRUIS   ;NO, SET FAILING FRU TO DRIVE
2802 040516 005303      10$:  DEC    R3
2803 040520 070327 000010      MUL   #10,R3        ;R3 IS OFFSET INTO TMSCP ERROR TABLE
2804 040524 012702 023062      MOV   #L2ETBL,R2   ;GET TMSCP ERROR TABLE START ADDRESS
2805 040530 060302      ADD   R3,R2        ;R2 POINTS TO ENTRY IN TMSCP ERROR TABLE.
2806 040532 012237 023052      MOV   (R2)+,ERRTYP  ;LOAD ERROR TYPE
2807 040536 012237 023054      MOV   (R2)+,ERRNBR  ;LOAD ERROR NUMBER
2808 040542 012237 023056      MOV   (R2)+,ERRMSG  ;LOAD ERROR MESSAGE ADDRESS
2809 040546 012237 023060      MOV   (R2)+,ERRBLK  ;LOAD ERROR SUBROUTINE ADDRESS
2810 040552              ERROR              ;CALL "ERROR" MACRO
2810 040552 104460              TRAP   C$ERROR
2811 040554 052764 000020 000014 GOABO: BIS   #ABTFLG,LUNFLG(R4) ;SET ABORT FLAG
2812 040562              DORPT
2812 040562 104424              TRAP   C$DRPT      ;DUMP NORMAL COMPLETION STATISTICS
2813 040564 012603      EXIT: MOV   (SP)+,R3  ;RESTORE REGISTERS
2814 040566 012602      MOV   (SP)+,R2
2815 040570 012601      MOV   (SP)+,R1
2816 040572 000207      RTS   PC
2817
2818 040574              ENDMOD
2829              .TITLE MISCELLANEOUS SECTIONS
2830              .SBTTL REPORT CODING SECTION
2858
2859 040574              BGNMOD
2860
    
```

```

2862          .SBTTL  INITIALIZE SECTION
2863
2864          ;**
2865          ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
2866          ; AT THE BEGINNING OF EACH PASS.
2867          ;--
2868
2869 040574          BGNINIT
040574          L$INIT::
2870
2871
2872 040574          READEF #EF.START          ;IF THIS IS A FRESH START
040574 012700 000040  MOV #EF.START,RO
040600 104447  TRAP C$REFG
2873 040602          BCOMPLETE START          ; THEN GO TO START
040602 103421  BCS START
2874
2875 040604          READEF #EF.RESTART        ;IF THIS IS A RESTART
040604 012700 000037  MOV #EF.RESTART,RO
040610 104447  TRAP C$REFG
2876 040612          BCOMPLETE START          ; THEN GO TO START
040612 103415  BCS START
2877
2878 040614          READEF #EF.PWR           ;IF POWER-FAIL OCCURRED
040614 012700 000034  MOV #EF.PWR,RO
040620 104447  TRAP C$REFG
2879 040622          BCOMPLETE START          ; THEN START FROM THE BEGINNING
040622 103411  BCS START
2880
2881 040624          READEF #EF.NEW           ;IF THIS IS A NEW PASS
040624 012700 000035  MOV #EF.NEW,RO
040630 104447  TRAP C$REFG
2882 040632          BCOMPLETE NUPASS        ; THEN SKIP START CODE
040632 103422  BCS NUPASS
2883
2884 040634          READEF #EF.CONTINUE       ;IF THIS IS A CONTINUE
040634 012700 000036  MOV #EF.CONTINUE,RO
040640 104447  TRAP C$REFG
2885 040642          BCOMPLETE END           ; THEN SKIP ALL INIT CODE
040642 103465  BCS END
2886
2887 040644          BR NEXT                 ;JUST HERE FOR NEXT UUT
2888
2889 040646          START:
2890 040646 012737 000000 002270  MOV #0,PASCNT          ;INITIALIZE PASS COUNT
2891 040654 005037 002274          CLR KTFLAG            ;IN CASE WE'RE STARTED > THAN ONCE
2892 040660 012704 002212          MOV #LUNBLK,R4          ;R4 WILL ALWAYS POINT TO LUNBLK
2893 040664 022737 001400 002120  CMP #1400,L$HIME       ;IF <= 28KWORDS OF MEMORY PRESENT
2894 040672 103002          BHS NUPASS        ; THEN SKIP NEXT
2895 040674 004737 036102          JSR PC,KTTEST        ; ELSE SEE IF MMU IS PRESENT
2896
2897 040700          NUPASS: BRESET           ;CLEAR THE WORLD
040700 104433          TRAP C$RESET
2898 040702 005237 002270          INC PASCNT            ;UPDATE THE PASS COUNT
2899 040706 012737 177777 002312  MOV #-1,LOGUNT        ;INITIALIZE LOGICAL UNIT COUNT
2900
2901 040714 005237 002312          NEXT: INC LOGUNT          ;POINT TO NEXT UUT

```

```

2902 040720 023737 002312 002012      CMP      LOGUNT,L$UNIT      ;IF WE'VE PASSED MAXIMUM UUT'S
2903 040726 001433                      BEQ      END                ; THEN LEAVE INIT
2904
2905 040730                      GPHARD  LOGUNT,RO          ;GET P-TABLE FOR THIS UNIT
      040730 013700 002312      MOV      LOGUNT,RO
      040734 104442                      TRAP    C$GPHRD
2906 040736                      BNCOMPLETE NEXT          ;TRY AGAIN
      040736 103366                      BCC     NEXT
2907
2908 040740 011064 000000      MOV      (R0),TKIP(R4)      ;PUT IP REG ADDRESS IN LUNBLK
2909 040744 012064 000002      MOV      (R0),TKSA(R4)     ; AND ANOTHER COPY IN LUNBLK
2910 040750 062764 000002 000002      ADD     #2,TKSA(R4)        ;MAKE IT THE SA REG ADDRESS
2911 040756 012064 000004      MOV      (R0),TKVEC(R4)    ;GET THE VECTOR INTO THE LUNBLK
2912 040762 011064 000006      MOV      (R0),MSCPUN(R4)   ;PUT THE T/MSCP UNIT # IN LUNBLK
2913 040766 004737 036266      JSR     PC,RSTVEC         ;SET UUT VECTOR FOR ILLEGAL INTRPTS.
2914 040772                      PRINTF  #IMSG,LOGUNT      ;"TESTING UNIT N"
      040772 013746 002312      MOV      LOGUNT,-(SP)
      040776 012746 041022      MOV      #IMSG,-(SP)
      041002 012746 000002      MOV      #2,-(SP)
      041006 010600                      MOV      SP,RO
      041010 104417                      TRAP    C$PNTF
      041012 062706 000006      ADD     #6,SP
2915
2916 041016                      END:
2917 041016                      EXIT   INIT
      041016 104432                      TRAP   C$EXIT
      041020 000032                      .WORD L10011-.
2918
2930 041022 045 116 045  IMGS:  .ASCIZ  ?*N*ATESTING UNIT *D1*N?
2931                      .EVEN
2932
2933 041052                      ENDINIT
      041052                      L10011.
      041052 104411                      TRAP   C$INIT

```

```

2935 .SBTTL CLEANUP CODING SECTION
2936
2937
2938 ;**
2939 ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
2940 ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
2941 ;--
2942 041054 BGNCLN
      041054 L$CLEAN::
2943
2950 041054 032764 000000G 002212 BIT #T9FLAG,LUNBLK(R4) ;IF NOT HERE FROM TEST 9
2951 041062 001400 BEQ ENDCLE ; THEN SKIP THE REST
2952
2953 ;EVENTUALLY MORE CODE WILL BE PLACED HERE TO GUARANTEE THAT AN ABORT
2954 ;COMMAND IS ISSUED TO THE UUT TO STOP EXECUTION OF THE LOCAL PROGRAM.
2955
2956 041064 005064 000014 ENDCLE: CLR LUNFLG(R4) ;CLEAR OUT THE LUN FLAGS
2957
2958 ;NOTE: THIS LINE OF CODE MAY HAVE TO BE REMOVED TO HANDLE tC FOLLOWED
2959 ;BY A PROCEED COMMAND CORRECTLY.
2960 041070 CLRVEC TKVEC(R4) ;PUT "TRAP CATCHER" INTO VECTOR
      041070 016400 000004 MOV TKVEC(R4),R0
      041074 104436 TRAP C$CVEC
2961
2962 041076 EXIT CLN
      041076 104432 TRAP C$EXIT
      041100 000002 .WORD L10012-.
2963
2975
2976 .EVEN
2977
2978 041102 ENDCLN
      041102 L10012:
      041102 104412 TRAP C$CLEAN

```


DROP UNIT SECTION

```

2980          .SBTTL  DROP UNIT SECTION
2981
2982          ;**
2983          ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
2984          ; TO NO LONGER BE TESTED.
2985          ;--
2986
2987 041104          BGNDU
          041104          L$DU::
2988
2994
2995 041104 012764 000001 000014          MOV      #DRPFLG,LUNFLG(R4)          ;LETS PROGRAM KNOW IT'S DEAD
2996
2997 041112          EXIT      DU
          041112 000167          .WORD  J$JMP
          041114 000000          .WORD  L10013-2-.
2998
3010
3011          .EVEN
3012
3013 041116          ENDDU
          041116          L10013:
          041116 104453          TRAP   C$DU

```

```

3015          .SBTTL  ADD UNIT SECTION
3016
3017
3018          ;++
3019          ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
3020          ; TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
3021          ; TO THE TEST CYCLE.
3022          ;--
3023 041120          BGNUAU
3024          L$AU::
3030
3031 041120          EXIT      AU
3032          041120 000167    .WORD  J$JMP
3033          041122 000000    .WORD  L10014-2-.
3034
3044
3045          .EVEN
3046
3047 041124          ENDAU
3048          041124 104452    L10014:
3049          041126          TRAP   C$AU
3050          ENDMOD

```

```

3053
3054
3058      00 000
3059
3060
3061
3062
3063
3064
3068
3069
3082
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3101 041126
      041126
3102 041126 000240
3103 041130 012737 000001 002272
3104 041136 005737 002270
3105 041142 001404
3106 041144 012737 000010 002272
3107 041152 000240
3108 041154
      041154
      041154 104402
3109 041156 005037 002276
3110
3111 041162
      041162 012746 000340
      041166 012746 036054
      041172 012746 000004
      041176 012746 000003
      041202 104437
      041204 062706 000010
3112 041210 000240
3113 041212 005074 000000
3114 041216 000240
3115 041220
      041220 012727 000001
      041224 000000
      041226 013727 002116
      041232 000000
      041234 005367 177772
      041240 001375
      041242 005367 177756

      .TITLE HARDWARE TEST
      HELP=0      ; CONTROL LISTING OF HELP INFORMATION
                  ; HELP=0  NO LIST
                  ; HELP=1  LIST

      ;ONEFILE=   ; CONTROL USE OF SOURCE FILES
                  ; ONEFILE IS NOT DEFINED  ASSEMBLE EACH SOURCE FILE SEPARATELY
                  ; ONEFILE=ANYTHING  ASSEMBLE ALL SOURCE FILES TOGETHER

      .SBTTL  TEST 1: EXISTENCE VERIFICATION TEST

      ;*****
      ;*****
      ;
      ;TEST 1 - EXISTENCE VERIFICATION TEST
      ; THIS TEST VERIFIES THE EXISTENCE OF THE UUT BY
      ; ATTEMPTING TO READ FIRST THE IP AND THEN THE SA
      ; REGISTERS OF THE TK50.  VECTOR 4 IS SET UP WITH
      ; A TRAP HANDLING ROUTINE IN CASE OF A NON-EXISTENT
      ; MEMORY TIMEOUT.
      ;
      ;*****
      ;*****

      BGNTST
T1::
      NOP
      MOV      #1,ITRCNT      ;SET UP FOR ONE TEST ITERATION
      TST      PASCNT        ;IF PASS 0
      BEQ      T1.1          ; THEN START TEST
      MOV      #10,ITRCNT    ; ELSE DO MULTIPLE ITERATIONS
      NOP
      BGNSUB

T1.1:
      TRAP     C$BSUB
1$:  CLR      TRP4FG        ;CLEAR NXM TRAP FLAG

      SETVEC   #VEC4,#TRAP4,#PRI07 ;SET UP VECTOR 4 FOR NXM TRAP
      MOV      #PRI07,-(SP)
      MOV      #TRAP4,-(SP)
      MOV      #VEC4,-(SP)
      MOV      #3,-(SP)
      TRAP     C$SVEC
      ADD      #10,SP
      NOP
      CLR      @TKIP(R4)    ;WRITE THE IP REGISTER
      NOP
      DELAY    1           ;MAKE SURE TIMEOUT CAN OCCUR
      MOV      #1,(PC)+
      .WORD    0
      MOV      L$DLY,(PC)+
      .WORD    0
      DEC      -6(PC)
      BNE     .-4
      DEC     -22(PC)

```

```

041246 001367      BNE      .-20
3116
3117 041250 005737 002276      TST      TRP4FG      ;IF NO TRAP OCCURRED
3118 041254 001416      BEQ      5$          ; THEN CONTINUE TEST
3119 041256 000240      NOP
3120 041260 012737 032640 002310  MOV      #CTRL,FRUIS ;IDENTIFY FAILING FRU FOR PRINTOUT
3121 041266 012737 032640 002310  ERRDF    5,MSG5,PRIERR ;"NXM ON READ TKIP"
      041266 104455      TRAP     C$ERDF
      041270 000005      .WORD   5
      041272 030334      .WORD   MSG5
      041274 033146      .WORD   PRIERR
3122 041276      CKLOOP           ;LOOP ON ERROR?
      041276 104406      TRAP     C$CLP1
3123 041300      DODU            ;DROP UNIT
      041300 013700 002312  MOV      LOGUNT
      041304 104451      TRAP     C$DODU
3124 041306      ESCAPE         ;CAN'T CONTINUE
      041306 104410      TRAP     C$ESCAPE
      041310 000002      .WORD   L10016-.
3125
3126 041312      5$: ENDSUB
      041312      L10016:
      041312 104403      TRAP     C$ESUB
3127 041314 000240      NOP
3128 041316      CLRVEC        #VEC4      ;RESTORE VECTOR 4
      041316 012700 000004  MOV      #VEC4,RO
      041322 104436      TRAP     C$CVEC
3129 041324 032764 000001 000014  BIT      #DRPFLG,LUNFLG(R4) ;IF UNIT WAS NOT DROPPED
3130 041332 001402      BEQ      T1.2      ; THEN CONTINUE TESTING
3131 041334      ESCAPE         TST
      041334 104410      TRAP     C$ESCAPE      ; ELSE LEAVE TEST
      041336 000264      .WORD   L10015-.
3132
3133 041340      BGNSUB
      041340      T1.2:
      041340 104402      TRAP     C$BSUB
3134 041342 005037 002276      10$: CLR      TRP4FG      ;CLEAR NXM ERROR FLAG
3135
3136 041346      SETVEC        #VEC4,#TRAP4,#PRI07 ;SET VECTOR 4 FOR NXM TRAPS
      041346 012746 000340  MOV      #PRI07,-(SP)
      041352 012746 036054  MOV      #TRAP4,-(SP)
      041356 012746 000004  MOV      #VEC4,-(SP)
      041362 012746 000003  MOV      #3,-(SP)
      041366 104437      TRAP     C$SVEC
      041370 062706 000010  ADD      #10,SP
3137 041374 000240      NOP
3138 041376 005774 000002      TST      @TKSA(R4)      ;READ THE SA REGISTER
3139 041402 000240      NOP
3140 041404      DELAY        25.      ;WAIT TO ALLOW NXM TRAP
      041404 012727 000031  MOV      #25.,(PC)+
      041410 000000      .WORD   0
      041412 013727 002116  MOV      L$DLY,(PC)+
      041416 000000      .WORD   0
      041420 005367 177772  DEC      -6(PC)
      041424 001375      BNE      .-4
      041426 005367 177756  DEC      -22(PC)
      041432 001367      BNE      . 20

```

```

3141
3142 041434 005737 002276      TST      TRP4FG      ;IF NXM DID NOT OCCUR
3143 041440 001416      BEQ      15$        ; THEN CONTINUE TEST
3144 041442 000240      NOP
3145 041444 012737 032640 002310  MOV      #CTRL,FRUIS ;IDENTIFY FAILING FRU FOR PRINTOUT
3146 041452      ERRDF      7,MSG7,PRIERR ;"NXM ON FIRST READ OF SA"
      041452 104455      TRAP      C$ERDF
      041454 000007      .WORD     7
      041456 030406      .WORD     MSG7
      041460 033146      .WORD     PRIERR
3147 041462      CKLOOP
      041462 104406      TRAP      C$CLP1      ;LOOP ON ERROR?
3148 041464      DODU      LOGUNT      ;DROP UNIT IF NOT
      041464 013700 002312  MOV      LOGUNT,RO
      041470 104451      TRAP      C$DODU
3149 041472      ESCAPE     SUB          ;LEAVE TEST
      041472 104410      TRAP      C$ESCAPE
      041474 000062      .WORD     L10017-.
3150
3151 041476 017464 000002 000012 15$:  MOV      @TKSA(R4),TKSASV(R4) ;GET A COPY OF SA IN MEMORY
3152 041504 032764 004000 000012  BIT      #B.S1,TKSASV(R4) ;IF STEP 1 BIT IS SET
3153 041512 001021      BNE      16$        ; THEN TEST 1 IS COMPLETE
3154 041514 000240      NOP
3155 041516 012737 004000 002314  MOV      #B.S1,SAEXP ;LOAD "EXPECTED FOR PRINTOUT
3156 041524 012737 032640 002310  MOV      #CTRL,FRUIS ;IDENTIFY FAILING FRU FOR PRINTOUT
3157 041532      ERRDF      8.,MSG8,PRISA ;"SA REG IN ERROR ON FIRST READ"
      041532 104455      TRAP      C$ERDF
      041534 000010      .WORD     8
      041536 030427      .WORD     MSG8
      041540 032764      .WORD     PRISA
3158 041542      CKLOOP
      041542 104406      TRAP      C$CLP1      ;LOOP ON ERROR?
3159 041544      DODU      LOGUNT      ;DROP UNIT IF NOT
      041544 013700 002312  MOV      LOGUNT,RO
      041550 104451      TRAP      C$DODU
3160 041552      ESCAPE     SUB          ;LEAVE TEST
      041552 104410      TRAP      C$ESCAPE
      041554 000002      .WORD     L10017-.
3161 041556      ENDSUB
      041556      L10017:
      041556 104403      TRAP      C$ESUB
3162
3163 041560 005037 002314      20$:  CLR      SAEXP      ;CLEAR ERROR INDICATOR
3164 041564      CLRVEC     #VEC4      ;RESTORE VECTOR 4
      041564 012700 000004  MOV      #VEC4,RO
      041570 104436      TRAP      C$CVEC
3165 041572 032764 000001 000014  BIT      #DRPFLG,LUNFLG(R4) ;IF UNIT DROPPED
3166 041600 001006      BNE      25$        ; THEN LEAVE NOW
3167 041602 005337 002272  DEC      ITRCNT      ;IF ITERATIONS EQUAL 0
3168 041606 000240      NOP
3169 041610 001402      BEQ      25$        ; THEN LEAVE TEST
3170 041612 000137 041154  JMP      T1.1        ; ELSE GO BACK FOR MORE
3171
3172 041616      EXIT
      041616 104432      TRAP      C$EXIT
      041620 000002      .WORD     L10015-.
3173

```

3174
3175
3176
3177 041622
041622
041622 104401

.EVEN
ENDTST
L10015:
TRAP C\$ETST

```

3180 .SBTTL TEST 2: SA REGISTER WRAP TEST
3184
3185 ;*****
3186 ;*****
3187 ;
3188 ;TEST 2 - SA REGISTER WRAP TEST
3189 ; THIS TEST WILL INITIALIZE THE UUT BY WRITING TO ITS
3190 ; IP REGISTER. IT WILL FORCE THE UUT INTO DIAGNOSTIC
3191 ; WRAP MODE, AND WRITE FIRST A FLOATING 0 DATA PATTERN,
3192 ; FOLLOWED BY A FLOATING 1 DATA PATTERN TO THE SA REG
3193 ; EACH WRITE WILL BE FOLLOWED BY A READ AND COMPARE
3194 ; OPERATION.
3195 ;
3196 ;*****
3197 ;*****
3201 041624 BGNTST
    041624
3202 T2::
3203 041624 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;IF UUT NOT DROPPED
3204 041632 001423 BEQ G02 ; THEN DO TEST
3205 041634 RFLAGS FLAGS ;ELSE GO GET SUPRVISOR FLAGS
    041634 104421 TRAP C$RFLA
3206 041642 010037 002334 MOV RO,FLAGS
3207 041650 032737 001000 002334 BIT #PNT,FLAGS ;SEE IF WE'RE PRINTING TEST NUMBERS
3208 041652 001412 BEQ 1$ ;NO, DON'T PRINT BYPASSED
    041652 013746 002114 PRINTF #BYPASS,L$TEST ; ELSE PRINT THE TEST
    041656 012746 047502 MOV L$TEST,-(SP)
    041662 012746 000002 MOV #BYPASS,-(SP)
    041666 010600 MOV #2,-(Sf)
    041670 104417 MOV SP,RO
    041672 062706 000006 TRAP C$PNTF
3209 041676 104432 1$: EXIT TST ;BYPASSED MESSAGE AND GET OUT
    041700 000504 TRAP C$EXIT
    .WORD L10020-.
3210
3211 041702 012737 000001 002316 G02: MOV #1,INISTP ;STEP 1 FOR ERROR PRINTOUT
3212 041710 012737 000001 002272 MOV #1,ITRCNT ;SET UP FOR ONE TEST ITERATION
3213 041716 022737 000001 002270 CMP #1,PASCNT ;IF FIRST PASS
3214 041724 001403 BEQ 2$ ; THEN START TEST
3215 041726 012737 000002 002272 MOV #2,ITRCNT ; ELSE DO 2 ITERATIONS
3216
3217 041734 012737 140000 002314 2$: MOV #BIT15!B.WR,SAEXP ;SET UP STEP 1 FOR DIAG. WRAP MODE
3218 041742 013737 002314 002250 MOV SAEXP,STPTBL ;PUT IT IN STEP 1 OF TABLE
3219 041750 004737 036434 JSR PC,STEP1 ;GO DO IT
3220
3221 041754 005737 002320 TST STEPST ;IF STATUS OKAY
3222 041760 001415 BEQ 5$ ; THEN CONTINUE TEST
3223
3224 041762 012737 032640 002310 MOV #CTRL,FRUIS ;FAILING FRU FOR PRINTOUT
3225 041770 ERRDF 9.,MSG9,PRIINI ;"SA CONTENTS IN ERROR"
    041770 104455 TRAP C$ERDF
    041772 000011 .WORD 9
    041774 030465 .WORD MSG9
    041776 032740 .WORD PRIINI
3226 042000 CKLOOP ;LOOP ON ERROR?
    042000 104406 TRAP C$CLP1
    
```

```

3227 042002          DODU      LOGUNT          ;DROP UUT
      042002 013700 002312  MOV      LOGUNT,RO
      042006 104451  TRAP     C$DODU
3228 042010          ESCAPE    TST           ;LEAVE TST
      042010 104410  TRAP     C$ESCAPE
      042012 000372  .WORD   L10020-.
3229
3230 042014 012737 000100 002326 5$:  MOV      #100, OUTER ;SET UP FOR DELAY ROUTINE
3231 042022 012737 037200 002324 6$:  MOV      #16000., INNER ;SET UP INNER
3232 042030 017464 000002 000012  MOV      @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
3233 042036 023764 002314 000012  CMP      SAEXP,TKSASV(R4) ;IF SA IS WHAT WE EXPECT
3234 042044 001422  BEQ      10$ ; THEN MOVE ALONG
3235 042046 004737 036406  JSR      PC,PDELAY ; ELSE GIVE UUT SOME TIME
3236 042052 005737 002330  TST      TOUT ;IF NO TIMEOUT YET
3237 042056 001761  BEQ      6$ ; THEN GO TAKE ANOTHER LOOK
3238
3239 042060 012737 032640 002310  MOV      #CTRL,FRUIS ;FAILING FRU FOR PRINTOUT
3240 042066          ERRDF    10.,MSG9,PRIINI ;"SA CONTENTS IN ERROR"
      042066 104455  TRAP     C$ERDF
      042070 000012  .WORD   10
      042072 030465  .WORD   MSG9
      042074 032740  .WORD   PRIINI
3241 042076          CKLOOP
      042076 104406  TRAP     C$CLP1
3242 042100          DODU      LOGUNT
      042100 013700 002312  MOV      LOGUNT,RO
      042104 104451  TRAP     C$DODU
3243 042106          ESCAPE    TST
      042106 104410  TRAP     C$ESCAPE
      042110 000274  .WORD   L10020-.
3244
3245 042112 012737 177776 002322 10$: MOV      #177776,WRDATA ;INITIALIZE WRAP DATA
3246 042120 013774 002322 000002 11$: MOV      WRDATA,@TKSA(R4) ;SEND DATA TO UUT
3247 042126 013737 002322 002314  MOV      WRDATA,SAEXP ;SAVE A COPY FOR COMPARE
3248 042134 012737 000100 002326  MOV      #100, OUTER ;SET UP FOR DELAY ROUTINE
3249
3250 042142 013737 037200 002324 15$: MOV      16000., INNER ;INNER TOO
3251 042150 017464 000002 000012  MOV      @TKSA(R4),TKSASV(R4) ;READ SA
3252 042156 023764 002314 000012  CMP      SAEXP,TKSASV(R4) ;IF DATA MATCHES
3253 042164 001422  BEQ      20$ ; THEN CHANGE DATA
3254 042166 004737 036406  JSR      PC,PDELAY ; ELSE GIVE UUT SOME TIME
3255 042172 005737 002330  TST      TOUT ;IF NO TIMEOUT YET
3256 042176 001761  BEQ      15$ ; THEN GO TAKE ANOTHER LOOK
3257
3258 042200 012737 032640 002310  MOV      #CTRL,FRUIS ;FAILING FRU FOR PRINTOUT
3259 042206          ERRDF    11.,MSG10,PRIINI ;"SA WRONG IN DATA WRAP"
      042206 104455  TRAP     C$ERDF
      042210 000013  .WORD   11
      042212 030512  .WORD   MSG10
      042214 032740  .WORD   PRIINI
3260 042216          CKLOOP
      042216 104406  TRAP     C$CLP1
3261 042220          DODU      LOGUNT
      042220 013700 002312  MOV      LOGUNT,RO
      042224 104451  TRAP     C$DODU
3262 042226          ESCAPE    TST ;GET OUT IF NOT LOOPING
      042226 104410  TRAP     C$ESCAPE

```



```

042230 000154 .WORD L10020-.
3263
3264 042232 006137 002322 20$: ROL WRDATA ;SHIFT TEST PATTERN
3265 042236 103730 BCS 11$ ;WE'RE NOT DONE YET
3266
3267 042240 012737 000001 002322 MOV #1,WRDATA ;SET UP FOR FLOATING 1 PATTERN
3268 042246 013774 002322 000002 24$: MOV WRDATA,@TKSA(R4) ;SEND DATA TO UUT
3269 042254 013737 002322 002314 MOV WRDATA,SAEXP ;KEEP A COPY FOR COMPARE
3270 042262 012737 000100 002326 MOV #100,OUTER ;SET UP FOR DELAY ROUTINE
3271
3272 042270 012737 016000 002324 25$: MOV #16000,INNER ;DELAY ROUTINE TOO
3273 042276 017464 000002 000012 MOV @TKSA(R4),TKSASV(R4) ;READ THE SA
3274 042304 023764 002314 000012 CMP SAEXP,TKSASV(R4) ;IF IT MATCHES
3275 042312 001422 BEQ 30$ ; THEN SEE IF WE'RE DONE
3276 042314 004737 036406 JSR PC,PDELAY ; ELSE GIVE UUT SOME MORE TIME
3277 042320 005737 002330 TST TOUT ;IF NO TIMEOUT YET
3278 042324 001761 BEQ 25$ ; THEN TAKE ANOTHER LOOK
3279
3280 042326 012737 032640 002310 MOV #CTRL,FRUIS ;FAILING FRU FOR PRINTOUT
3281 042334 ERROF 12.,EMSG10,PRIINI ;"SA WRONG IN DATA WRAP"
042334 104455 TRAP C$ERDF
042336 000014 .WORD 12
042340 030512 .WORD EMSG10
042342 032740 .WORD PRIINI
3282 042344 CKLOOP
042344 104406 TRAP C$CLP1
3283 042346 DODU LOGUNT
042346 013700 002312 MOV LOGUNT,RO
042352 104451 TRAP C$DODU
3284 042354 ESCAPE TST ;LEAVE TEST IF NOT LOOPING
042354 104410 TRAP C$ESCAPE
042356 000026 .WORD L10020-.
3285
3286 042360 006137 002322 30$: ROL WRDATA ;SHIFT DATA PATTERN
3287 042364 103330 BCC 24$ ;WE'RE NOT DONE YET
3288 042366 005337 002272 DEC ITRCNT ;IF ITERATIONS = 0
3289 042372 001402 BEQ T2EXT ; THEN LEAVE TEST
3290 042374 000137 041734 JMP 2$ ; ELSE DO ANOTHER ONE
3291
3292 042400 T2EXT: EXIT TST ;GET OUTTA HERE
042400 104432 TRAP C$EXIT
042402 000002 .WORD L10020-.
3293
3294 042404 ENDTST
042404 L10020: TRAP C$ETST
042404 104401

```

```

3297          .SBTTL TEST 3: INITIALIZATION TEST
3301
3302
3303          ;*****
3304          ;*****
3305          ;
3306          ;TEST 3 - INITIALIZATION TEST
3307          ;
3308          ;   THIS TEST COMMENCES THE UQ-PORT INITIALIZATION SEQUENCE
3309          ;   WITH INTERRUPTS DISABLED. IT VERIFIES THAT ALL STEP
3310          ;   TRANSITIONS OCCUR WITHIN THE ALLOTTED TIME, AND THAT ALL
3311          ;   HOST SUPPLIED INFORMATION IS ECHOED BY THE UUT. THE
3312          ;   PROGRAM FURTHER VERIFIES THAT NO INTERRUPTS OCCUR AS A
3313          ;   RESULT OF THE STEP TRANSITIONS.
3314          ;
3315          ;*****
3316          ;*****
3319
3320 042406      BGNTST
          042406      T3::
3321
3322 042406 032764 000001 000014      BIT      #DRPFLG,LUNFLG(R4)      ;IF UUT NOT DROPPED
3323 042414 001423                    BEQ      G03                          ; THEN DO TEST
3324 042416                    RFLAGS   FLAGS                          ;ELSE GO GET SUPRVISOR FLAGS
          042416 104421                    TRAP   C$RFLA
          042420 010037 002334                    MOV    RO,FLAGS
3325 042424 032737 001000 002334      BIT      #PNT,FLAGS                          ;SEE IF WE'RE PRINTING TEST NUMBERS
3326 042432 001412                    BEQ      1$                          ;NO, DON'T PRINT BYPASSED
3327 042434                    PRINTF   #BYPASS,L$TEST                      ; ELSE PRINT THE TEST
          042434 013746 002114                    MOV    L$TEST,-(SP)
          042440 012746 047502                    MOV    #BYPASS,-(SP)
          042444 012746 000002                    MOV    #2,-(SP)
          042450 010600                    MOV    SP,RO
          042452 104417                    TRAP   C$PNTF
          042454 062706 000006                    ADD    #6,SP
3328 042460                    1$: EXIT    TST                          ;BYPASSED MESSAGE AND GET OUT
          042460 104432                    TRAP   C$EXIT
          042462 000374                    .WORD  L10021-.
3329
3330 042464 012737 032640 002310      G03:  MOV    #CTRL,FRUIS                      ;FAILING FRU IN CASE OF ERROR
3331 042472 012737 000001 002272      MOV    #1,ITRCNT                          ;SET UP FOR ONE TEST ITERATION
3332 042500 022737 000001 002270      CMP    #1,PASCNT                          ;IF FIRST PASS
3333 042506 001403                    BEQ    2$                          ; THEN START TEST
3334 042510 012737 000012 002272      MOV    #10.,ITRCNT                        ; ELSE DO 10 ITERATIONS
3335
3336 042516 012705 000000                    2$:  MOV    #0,R5                          ;SET UP R5 AS INDEX TO STEP TABLES
3337 042522 012737 000001 002316      MOV    #1,INISTP                          ;STEP 1 FOR ERROR PRINTOUT
3338 042530 016437 000004 002250      MOV    TKVEC(R4),STPTBL                    ;PUT VECTOR IN STEP 1
3339 042536 006237 002250                    ASR    STPTBL                          ;DIVIDE BY TWO
3340 042542 006237 002250                    ASR    STPTBL                          ;DIVIDE BY FOUR
3341 042546 013737 002250 002264      MOV    STPTBL,CMPTBL*4                    ;PUT VECTOR IN STEP 3 COMPARE
3342 042554 052737 104400 002250      BIS    #104400,STPTBL                    ;REST OF STEP ONE
3343 042562 012737 005700 002260      MOV    #B.S1!B.QB!B.DI!B.OO!B.MP,CMPTBL
3344
3345 042570 012737 060050 002252      MOV    #COMMAR,STPTBL*2                    ;STEP 1 COMPARE VALUE
3346 042576 012737 010211 002262      MOV    #010211,CMPTBL*2                    ;STEP 2 - COMM AREA ADDRESS
3347 042604 012737 000000 002254      MOV    #0,STPTBL*4                          ;STEP 3 - HIGH ADDRESS
3348 042612 112737 000040 002265      MOVB   #40,CMPTBL*5                          ;REST OF STEP 3 COMPARE

```

```

3349 042620 012737 000000 002256      MOV      #0,STPTBL+6      ;STEP 4
3350 042626 012737 040000 002266      MOV      #040000,CMPTBL+6 ;STEP 4 COMPARE
3351
3352 042634 004737 036434      JSR      PC,STEP1      ;GO DO IT
3353 042640 005737 002320      TST      STEPST      ;IF STATUS OKAY
3354 042644 001412      BEQ      5$           ; THEN CONTINUE TEST
3355
3356 042646      ERRDF    9.,EMSG9,PRIINI ;"SA CONTENTS IN ERROR"
      042646 104455      TRAP    C$ERDF
      042650 000011      .WORD  9
      042652 030465      .WORD  EMSG9
      042654 032740      .WORD  PRIINI
3357 042656      CKLOOP   ;LOOP ON ERROR?
      042656 104406      TRAP    C$CLP1
3358 042660      DODU    LOGUNT      ;DROP UUT
      042660 013700 002312      MOV     LOGUNT,R0
      042664 104451      TRAP    C$DODU
3359 042666      ESCAPE  TST          ;LEAVE TST
      042666 104410      TRAP    C$ESCAPE
      042670 000166      .WORD  L10021-.
3360
3361 042672 005237 002316      5$:     INC     INISTP      ;ADJUST STEP COUNTER
3362 042676 062705 000002      ADD     #2,R5          ;ADJUST TABLE INDEX
3363 042702 012737 000100 002326      6$:     MOV     #100,OUTER    ;SET UP FOR DELAY ROUTINE
3364 042710 016537 002260 002314      MOV     CMPTBL(R5),SAEXP ;SET UP FOR COMPARE
3365 042716 012737 037200 002324      7$:     MOV     #16000.,INNER ;SET UP INNER
3366 042724 017464 000002 000012      MOV     @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
3367 042732 022705 000006      CMP     #6,R5          ;ARE WE IN STEP 4?
3368 042736 001005      BNE     8$           ;BRANCH IF NOT
3369 042740 033764 002314 000012      BIT     SAEXP,TKSASV(R4) ;JUST LOOK FOR STEP 4 BIT
3370 042746 001024      BNE     10$          ;IT'S SET SO LET'S GO
3371 042750 000404      BR      9$           ;STAY IN LOOP OTHERWISE
3372 042752 023764 002314 000012      8$:     CMP     SAEXP,TKSASV(R4) ;IF SA IS WHAT WE EXPECT
3373 042760 001417      BEQ     10$          ; THEN MOVE ALONG
3374 042762 004737 036406      9$:     JSR     PC,PDELAY    ; ELSE GIVE UUT SOME TIME
3375 042766 005737 002330      TST     TOUT          ;IF NO TIMEOUT YET
3376 042772 001751      BEQ     7$           ; THEN GO TAKE ANOTHER LOOK
3377
3378 042774      ERRDF    13.,EMSG9,PRIINI ;"SA CONTENTS IN ERROR"
      042774 104455      TRAP    C$ERDF
      042776 000015      .WORD  13
      043000 030465      .WORD  EMSG9
      043002 032740      .WORD  PRIINI
3379 043004      CKLOOP   ;
      043004 104406      TRAP    C$CLP1
3380 043006      DODU    LOGUNT
      043006 013700 002312      MOV     LOGUNT,R0
      043012 104451      TRAP    C$DODU
3381 043014      ESCAPE  TST
      043014 104410      TRAP    C$ESCAPE
      043016 000040      .WORD  L10021-.
3382
3383 043020 016574 002250 000002 10$:     MOV     STPTBL(R5),@TKSA(R4) ;WRITE NEXT STEP TO UUT
3384 043026 022705 000006      CMP     #6,R5          ;IF NOT IN STEP 4
3385 043032 001317      BNE     5$           ;GO BACK TO MAIN LOOP
3386
3387 043034 032764 000001 000014      BIT     #DRPFLG,LUNFLG(R4) ;HAS UUT BEEN DROPPED

```

3388 043042 001003
3389 043044 005337 002272
3390 043050 001222
3391

BNE T3EXT
DEC ITRCNT
BNE 2↓

;LEAVE NOW IF SO
;IF MORE ITERATIONS LEFT
; THEN GO DO IT AGAIN

3392 043052
043052 104432
043054 000002

T3EXT: EXIT TST
TRAP C#EXIT
.WORD L10021-.

3393
3394 043056
043056
043056 104401

ENDTST
L10021:
TRAP C#ETST

```

3397 .SBTTL TEST 4: VECTOR AND INTERRUPT TEST
3401
3402 ;*****
3403 ;*****
3404
3405 ;TEST 4 - VECTOR AND INTERRUPT TEST
3406 ; TEST 3 IS REPEATED, BUT WITH INTERRUPTS ENABLED.
3407 ; THE PROGRAM VERIFIES THAT AN INTERRUPT OCCURS AT
3408 ; THE END OF STEPS 1 - 3.
3409 ;
3410 ;*****
3411 ;*****
3415
3416
3417 043060          BGNTST
      043060          T4::
341P
3419 043060 032764 000001 000014      BIT    #DRPFLG,LUNFLG(R4)      ;IF UUT NOT DROPPED
3420 043066 001423                      BEQ    G04                      ; THEN DO TEST
3421 043070          RFLAGS    FLAGS      ;ELSE GO GET SUPRVISOR FLAGS
      043070 104421          TRAP    C$RFLA
      043072 010037 002334          MOV    R0,FLAGS
3422 043076 032737 001000 002334      BIT    #PNT,FLAGS      ;SEE IF WE'RE PRINTING TEST NUMBERS
3423 043104 001412          BEQ    1$      ;NO, DON'T PRINT BYPASSED
3424 043106          PRINTF    #BYPASS,L$TEST ; ELSE PRINT THE TEST
      043106 013746 002114          MOV    L$TEST,-(SP)
      043112 012746 047502          MOV    #BYPASS,-(SP)
      043116 012746 000002          MOV    #2,-(SP)
      043122 010600          MOV    SP,R0
      043124 104417          TRAP    C$PNTF
      043126 062706 000006          ADD    #6,SP
3425 043132          1$:          EXIT    TST      ;BYPASSED MESSAGE AND GET OUT
      043132 104432          TRAP    C$EXIT
      043134 000502          .WORD    L10022-.
3426
3427 043136 042764 000004 000014      G04:   BIC    #BRFLAG,LUNFLG(R4)      ;DO TEST WITH PRIORITY SET TO 0
3428 043144 042764 000002 000014      BIC    #INTFLG,LUNFLG(R4)      ;CLEAR THE INTERUPT FLAG
3429 043152 012737 032640 002310      MOV    #CTRL,FRUIS      ;FAILING FRU IN CASE OF ERROR
3430 043160 012737 000001 002272      MOV    #1,ITRCNT      ;SET UP FOR ONE TEST ITERATION
3431 043166 022737 000001 002270      CMP    #1,PASCNT      ;IF FIRST PASS
3432 043174 001403          BEQ    2$      ; THEN START TEST
3433 043176 012737 000012 002272      MOV    #10.,ITRCNT      ; ELSE DO 10 ITERATIONS
3434
3435 043204 004737 036316          2$:   JSR    PC,VECTOR      ;SET UP VECTOR WITH INTERRUPT HANDLER
3436 043210 012705 000000          MOV    #0,R5      ;SET UP R5 AS INDEX TO STEP TABLES
3437 043214 012737 000001 002316      MOV    #1,INISTP      ;STEP 1 FOR ERROR PRINTOUT
3438 043222 016437 000004 002250      MOV    TKVEC(R4),STPTBL ;PUT VECTOR IN STEP 1
3439 043230 006237 002250          ASR    STPTBL      ;DIVIDE BY TWO
3440 043234 006237 002250          ASR    STPTBL      ;DIVIDE BY FOUR
3441 043240 013737 002250 002264      MOV    STPTBL,CMPTBL+4 ;PUT VECTOR IN STEP 3 COMPARE
3442 043246 052737 104600 002250      BIS    #104600,STPTBL ;REST OF STEP ONE
3443 043254 012737 005700 002260      MOV    #B.S1!B.QB!B.DI!B.OO!B.MP,CMPTBL
3444
3445 043262 012737 060050 002252          MOV    #COMMAR,STPTBL+2 ;STEP 1 COMPARE VALUE
3446 043270 012737 010211 002262          MOV    #010211,CMPTBL+2 ;STEP 2 - COMM AREA ADDRESS
3447 043276 012737 000000 002254          MOV    #0,STPTBL+4 ;STEP 2 COMPARE
3448 043304 052737 000200 002264          BIS    #B.IE,CMPTBL+4 ;STEP 3 - HIGH ADDRESS ;SET THE INTERRUPT ENABLE BIT
    
```

```

3449 043312 112737 000040 002265      MOVB    #40,CMPTBL+5      ;REST OF STEP 3 COMPARE
3450 043320 012737 000000 002256      MOV     #0,STPTBL+6      ;STEP 4
3451 043326 012737 040000 002266      MOV     #040000,CMPTBL+6 ;STEP 4 COMPARE
3452
3453 043334 004737 036434      JSR     PC,STEP1         ;GO DO IT
3454 043340 005737 002320      TST     STEPST          ;IF STATUS OKAY
3455 043344 001412      BEQ     5$              ; THEN CONTINUE TEST
3456
3457 043346      ERRDF   14.,EMSG9,PRIINI ;"SA CONTENTS IN ERROR"
      043346 104455      TRAP   C$ERDF
      043350 000016      .WORD  14
      043352 030465      .WORD  EMSG9
      043354 032740      .WORD  PRIINI
3458 043356      CKLOOP          ;LOOP ON ERROR?
      043356 104406      TRAP   C$CLP1
3459 043360      DODU           ;DROP UUT
      043360 013700 002312      MOV     LOGUNT,R0
      043364 104451      TRAP   C$DODU
3460 043366      ESCAPE        ;LEAVE TST
      043366 104410      TRAP   C$ESCAPE
      043370 000246      .WORD  L10022-.
3461
3462 043372 012737 000100 002326 5$:      MOV     #100,OUTER      ;SET UP FOR DELAY ROUTINE
3463 043400 016537 002260 002314      MOV     CMPTBL(R5),SAEXP ;SET UP FOR COMPARE
3464 043406 012737 037200 002324 7$:      MOV     #16000.,INNER   ;SET UP INNER
3465 043414 032764 000002 000014      BIT     #INTFLG,LUNFLG(R4) ;IF INTERRUPT OCCURRED
3466 043422 001017      BNE     10$           ; THEN SEE IF SA IS CORRECT
3467 043424 004737 036406 9$:      JSR     PC,PDELAY      ; ELSE GIVE UUT SOME TIME
3468 043430 005737 002330      TST     TOUT          ;IF NO TIMEOUT YET
3469 043434 001764      BEQ     7$           ; THEN GO TAKE ANOTHER LOOK
3470
3471 043436      ERRDF   15.,EMSG11,PRIERR ;"EXPECTED INTERRUPT DID NOT OCCUR"
      043436 104455      TRAP   C$ERDF
      043440 000017      .WORD  15
      043442 030540      .WORD  EMSG11
      043444 033146      .WORD  PRIERR
3472 043446      CKLOOP
      043446 104406      TRAP   C$CLP1
3473 043450      DODU           ;DROP UUT
      043450 013700 002312      MOV     LOGUNT,R0
      043454 104451      TRAP   C$DODU
3474 043456      ESCAPE        ;LEAVE TST
      043456 104410      TRAP   C$ESCAPE
      043460 000156      .WORD  L10022-.
3475
3476 043462 042764 000002 000014 10$:     BIC     #INTFLG,LUNFLG(R4) ;CLEAR THE INTERRUPT FLAG
3477 043470 005237 002316      INC     INISTP         ;ADJUST THE STEP COUNTER
3478 043474 062705 000002      ADD     #2,R5          ;ADJUST TABLE INDEX
3479 043500 016537 002260 002314      MOV     CMPTBL(R5),SAEXP ;GET THE COMPARISON VALUE
3480 043506 017464 000002 000012      MOV     @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
3481 043514 022705 000006      CMP     #6,R5          ;ARE WE IN STEP 4?
3482 043520 001005      BNE     15$           ;BRANCH IF NOT
3483 043522 033764 002314 000012      BIT     SAEXP,TKSASV(R4) ;JUST LOOK FOR STEP 4 BIT
3484 043530 001017      BNE     20$           ;IT'S SET SO LET'S GO
3485 043532 000404      BR     16$           ;ERROR
3486 043534 023764 002314 000012 15$:     CMP     SAEXP,TKSASV(R4) ;IF SA IS WHAT WE EXPECT
3487 043542 001412      BEQ     20$           ; THEN MOVE ALONG

```

```

3488
3489 043544      16$:  ERRDF  16.,EMSG9,PRIINI      ;"SA CONTENTS IN ERROR"
      043544  104455  TRAP   C$ERDF
      043546  000020  .WORD  16
      043550  030465  .WORD  EMSG9
      043552  032740  .WORD  PRIINI
3490 043554      CKLOOP
      043554  104406  TRAP   C$CLP1
3491 043556      DODU   LOGUNT
      043556  013700  002312 MOV   LOGUNT,R0
      043562  104451  TRAP   C$DODU
3492 043564      ESCAPE  TST
      043564  104410  TRAP   C$ESCAPE
      043566  000050  .WORD  L10022-.
3493
3494 043570  016574  002250  000002  20$:  MOV   STPTBL(R5),@TKSA(R4)  ;WRITE NEXT STEP TO UUT
3495 043576  022705  000006  CMP   #6,R5                ;IF NOT IN STEP 4
3496 043602  001273  BNE   5$                    ;GO BACK TO MAIN LOOP
3497
3498 043604  032764  000001  000014  BIT   #DRPFLG,LUNFLG(R4)  ;HAS UUT BEEN DROPPED
3499 043612  001005  BNE   T4EXT                 ;LEAVE NOW IF SO
3500 043614  005337  002272  DEC   ITRCNT               ;IF NO MORE ITERATIONS LEFT
3501 043620  001402  BEQ   T4EXT                 ; THEN EXIT
3502 043622  000137  043204  JMP   2$                    ; ELSE DO IT AGAIN
3503
3504 043626  004737  036266  T4EXT: JSR   PC,RSTVEC       ;CATCH ILLEGAL INTERRUPTS
3505 043632      EXIT   TST
      043632  104432  TRAP   C$EXTT
      043634  000002  .WORD  L10022-.
3506
3507 043636      ENDTST
      043636      L10022:
      043636  104401  TRAP   C$ETST

```

```

3510 .SBTTL TEST 5: BR LEVEL TEST
3514
3515 ;*****
3516 ;*****
3517
3518 ;TEST 5 - BR LEVEL TEST
3519 ; THIS TEST INSURES THAT THE TK50 CAN NOT INTERRUPT
3520 ; WHEN THE CPU PRIORITY IS SET TO 7. THE TEST GOES
3521 ; ONLY THROUGH THE FIRST STEP OF THE INIT SEQUENCE
3522 ; SINCE THE CONTROLLER WILL "HANG" WAITING FOR THE
3523 ; INTERRUPT ACKNOWLEDGE.
3524
3525 ;*****
3526 ;*****
3530
3531 043640 BGNTST
3532 043640 T5::
3533 043640 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;IF UUT NOT DROPPED
3534 043646 001423 BEQ G05 ; THEN DO TEST
3535 043650 RFLAGS FLAGS ;ELSE GO GET SUPRVISOR FLAGS
3536 043652 104421 TRAP C$RFLA
3537 043656 032737 001000 002334 MOV RO,FLAGS
3538 043664 001412 BIT #PNT,FLAGS ;SEE IF WE'RE PRINTING TEST NUMBERS
3539 043666 PRINTF #BYPASS,L$TEST ;NO, DON'T PRINT BYPASSED
3540 043666 013746 002114 MOV L$TEST,-(SP) ; ELSE PRINT THE TEST
3541 043672 012746 047502 MOV #BYPASS,-(SP)
3542 043676 012746 000002 MOV #2,-(SP)
3543 043702 010600 MOV SP,RO
3544 043704 104417 TRAP C$PNTF
3545 043706 062706 000006 ADD #6,SP
3546 043712 104432 1$: EXIT TST ;BYPASSED MESSAGE AND GET OUT
3547 043714 000364 TRAP C$EXIT
3548 .WORD L10023-.
3549
3550 G05: BIS #BRFLAG,LUNFLG(R4) ;DO TEST WITH HIGH PRIORITY
3551 043716 052764 000004 000014 MOV #CTRL,FRUIS ;FAILING FRU IN CASE OF ERROR
3552 043724 012737 032640 002310 MOV #1,ITRCNT ;SET UP FOR ONE TEST ITERATION
3553 043732 012737 000001 002272 CMP #1,PASCNT ;IF FIRST PASS
3554 043740 022737 000001 002270 BEQ 2$ ; THEN START TEST
3555 043746 001403 MOV #2,ITRCNT ; ELSE DO 10 ITERATIONS
3556 043750 012737 000002 002272
3557
3558 2$: MTPS #PRI07 ;CPU PRIORITY = 7
3559 043756 106427 000340 JSR PC,VECTOR ;SET UP VECTOR WITH INTERRUPT HANDLER
3560 043762 004737 036316 MOV #0,R5 ;SET UP R5 AS INDEX TO STEP TABLES
3561 043766 012705 000000 MOV #1,INISTP ;STEP 1 FOR ERROR PRINTOUT
3562 043772 012737 000001 002316 MOV TKVEC(R4),STPTBL ;PUT VECTOR IN STEP 1
3563 044000 016437 000004 002250 ASR STPTBL ;DIVIDE BY TWO
3564 044006 006237 002250 ASR STPTBL ;DIVIDE BY FOUR
3565 044012 006237 002250 BIS #104600,STPTBL ;REST OF STEP ONE
3566 044016 052737 104600 002250 MOV TKVEC(R4),CMPTBL ;STEP 1 COMPARE VALUE
3567 044024 016437 000004 002260
3568
3569 JSR PC,STEP1 ;GO DO IT
3570 044032 004737 036434 TST STEPST ;IF STATUS OKAY
3571 044036 005737 002320 BEQ 5$ ; THEN CONTINUE TEST
3572 044042 001412
3573

```



```

3562 044044      ERRDF      14.,EMSG9,PRIINI      ;"SA CONTENTS IN ERROR"
      044044      104455      TRAP      C$ERDF
      044046      000016      .WORD     14
      044050      030465      .WORD     EMSG9
      044052      032740      .WORD     PRIINI
3563 044054      CKLOOP
      044054      104406      TRAP      C$CLP1      ;LOOP ON ERROR?
3564 044056      DODU      LOGUNT      ;DROP UUT
      044056      013700      002312      MOV      LOGUNT,RO
      044062      104451      TRAP      C$DODU
3565 044064      ESCAPE     TST      ;LEAVE TST
      044064      104410      TRAP      C$ESCAPE
      044066      000212      .WORD     L10023-.
3566
3567 044070      012737      000100      002326      5$:      MOV      #100, OUTER      ;SET UP FOR DELAY ROUTINE
3568 044076      016537      002260      002314      MOV      CMPTBL(R5),SAEXP      ;SET UP FOR COMPARE
3569 044104      012737      037200      002324      7$:      MOV      #16000.,INNER      ;SET UP INNER
3570 044112      004737      036406      JSR      PC,PDELAY      ; ELSE GIVE UUT SOME TIME
3571 044116      005737      002330      9$:      TST      TOUT      ;IF NO TIMEOUT YET
3572 044122      001770      BEQ      7$      ; THEN GO TAKE ANOTHER LOOK
3573
3574 044124      017464      000002      000012      MOV      @TKSA(R4),TKSASV(R4)      ;GET SA CONTENTS
3575 044132      023764      002314      000012      CMP      SAEXP,TKSASV(R4)      ;IF CONTENTS OKAY
3576 044140      001412      BEQ      10$      ; THEN CHECK FOR INTERRUPT
3577
3578 044142      ERRDF      17.,EMSG9,PRIINI      ;"SA CONTENTS IN ERROR"
      044142      104455      TRAP      C$ERDF
      044144      000021      .WORD     17
      044146      030465      .WORD     EMSG9
      044150      032740      .WORD     PRIINI
3579 044152      CKLOOP
      044152      104406      TRAP      C$CLP1
3580 044154      DODU      LOGUNT
      044154      013700      002312      MOV      LOGUNT,RO
      044160      104451      TRAP      C$DODU
3581 044162      ESCAPE     TST
      044162      104410      TRAP      C$ESCAPE
      044164      000114      .WORD     L10023-.
3582
3583 044166      032764      000002      000014      10$:      BIT      #INTFLG,LUNFLG(R4)      ;IF NO INTERRUPT OCCURRED
3584 044174      001415      BEQ      20$      ; THEN CARRY ON WITH TEST
3585 044176      042764      000002      000014      BIC      #INTFLG,LUNFLG(R4)      ;CLEAR FLAG IN CASE WE'RE LOOPING
3586 044204      ERRDF      18.,EMSG12,PRIINI      ;"INTRPT WITH CPU PRIORITY =7"
      044204      104455      TRAP      C$ERDF
      044206      000022      .WORD     18
      044210      030601      .WORD     EMSG12
      044212      032740      .WORD     PRIINI
3587 044214      CKLOOP
      044214      104406      TRAP      C$CLP1
3588 044216      DODU      LOGUNT
      044216      013700      002312      MOV      LOGUNT,RO
      044222      104451      TRAP      C$DODU
3589 044224      ESCAPE     TST
      044224      104410      TRAP      C$ESCAPE
      044226      000052      .WORD     L10023-.
3590
3591 044230      106427      000000      20$:      MTPS     #PRI00      ;CPU PRIORITY = 0

```

```

3592 044234 000240      NOP
3593 044236 000240      NOP
3594 044240 042764 000002 000014  BIC      #INTFLG,LUNFLG(R4)      ;DELAY FOR PENDING INTERRUPT
3595                                     ;CLEAR THE FLAG NOW
3596 044246 032764 000001 000014  BIT      #DRPFLG,LUNFLG(R4)      ;HAS UUT BEEN DROPPED
3597 044254 001005      BNE      T5EXT                ;LEAVE NOW IF SO
3598 044256 005337 002272      DEC      ITRCNT                ;IF NO MORE ITERATIONS LEFT
3599 044262 001402      BEQ      T5EXT                ; THEN EXIT
3600 044264 000137 043756      JMP      2$                    ; ELSE DO IT AGAIN
3601
3602 044270 004737 036266      T5EXT: JSR      PC,RSTVEC        ;CATCH ILLEGAL INTERRUPTS
3603 044274      EXIT      TST
      044274 104432      TRAP     C$EXIT
      044276 000002      .WORD   L10023-.
3604
3605 044300      ENDTST
      044300      L10023:
      044300 104401      TRAP     C$ETST

```

3608
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659

.SBTTL TEST 6: PURGE AND POLL TEST

```
*****  
*****  
;TEST 6 - PURGE AND POLL TEST  
; THIS TEST WILL AGAIN RUN THROUGH THE INIT SEQUENCE, THIS  
; TIME SETTING THE "PURGE AND POLL" BIT IN STEP 3. THIS  
; SHOULD CAUSE THE PORT TO DMA VARIOUS DATA PATTERNS TO  
; AND FROM THE COMMUNICATIONS AREA AND FINALLY LEAVE IT  
; CLEARED BEFORE TRANSITIONING TO STEP 4. THE PROGRAM WILL  
; HAVE FILLED THIS AREA WITH A BACKGROUND PATTERN OF ALL  
; 1'S DATA PRIOR TO STARTING THE INIT. WHNE STEP 4 IS  
; REACHED, THE PROGRAM WILL VERIFY THAT THE COMM AREA IS  
; ALL 0'S, AND THAT THE 20 WORDS PRECEDING AND SUCCEEDING  
; THE COMM AREA ARE UNTOUCHED. RING DEPTH USED IN THIS  
; TEST IS THE MINIMUM.  
*****  
*****
```

T6:: BGNTST

```
BIT #DRPFLG,LUNFLG(R4) ;IF UUT NOT DROPPED  
BEQ G06 ; THEN DO TEST  
RFLAGS FLAGS ;ELSE GO GET SUPRVISOR FLAGS  
TRAP C$RFLA  
MOV RO,FLAGS  
BIT #PNT,FLAGS ;SEE IF WE'RE PRINTING TEST NUMBERS  
BEQ 1$ ;NO, DON'T PRINT BYPASSED  
PRINTF #BYPASS,L$TEST ; ELSE PRINT THE TEST  
MOV L$TEST,-(SP)  
MOV #BYPASS,-(SP)  
MOV #2,-(SP)  
MOV SP,RO  
TRAP C$PNTF  
ADD #6,SP  
EXIT TST ;BYPASSED MESSAGE AND GET OUT  
TRAP C$EXIT  
.WORD L10024-.  
G06: MOV #CTRL,FRUIS ;FAILING FRU IN CASE OF ERROR  
MOV #1,ITRCNT ;SET UP FOR ONE TEST ITERATION  
CMP #1,PASCNT ;IF FIRST PASS  
BEQ 2$ ; THEN START TEST  
MOV #10.,ITRCNT ; ELSE DO 10 ITERATIONS  
2$: MOV #0,R5 ;SET UP R5 AS INDEX TO STEP TABLES  
MOV #1,INISTP ;STEP 1 FOR ERROR PRINTOUT  
MOV TKVEC(R4),STPTBL ;PUT VECTOR IN STEP 1  
ASR STPTBL ;DIVIDE BY TWO  
ASR STPTBL ;DIVIDE BY FOUR  
MOV STPTBL,CMPTBL+4 ;PUT VECTOR IN STEP 3 COMPARE  
BIS #104400,STPTBL ;REST OF STEP ONE  
MOV #B.S1!B.QB!B.DI!B.OD!B.MP,CMPTBL  
;STEP 1 COMPARE VALUE
```

3660	044464	012737	060050	002252	MOV	#COMMAR,STPTBL+2	;STEP 2 - COMM AREA ADDRESS
3661	044472	012737	010211	002262	MOV	#010211,CMPTBL+2	;STEP 2 COMPARE
3662	044500	012737	100000	002254	MOV	#8.PP,STPTBL+4	;STEP 3 - HIGH ADDRESS AND PRGE/POLL
3663	044506	112737	000040	002265	MOVB	#40,CMPTBL+5	;REST OF STEP 3 COMPARE
3664	044514	012737	000000	002256	MOV	#0,STPTBL+6	;STEP 4
3665	044522	012737	040000	002266	MOV	#040000,CMPTBL+6	;STEP 4 COMPARE
3666							
3667	044530	012737	000012	002306	MOV	#10.,CMARLG	;LENGTH OF COMM AREA FOR THIS TEST
3668	044536	004737	036532		JSR	PC,BAKPAT	;FILL COMM AREA WITH ALL 1'S DATA
3669							
3670	044542	004737	036434		JSR	PC,STEP1	;GO DO IT
3671	044546	005737	002320		TST	STEPST	;IF STATUS OKAY
3672	044552	001412			BEQ	5\$; THEN CONTINUE TEST
3673							
3674	044554				ERRDF	19.,EMSG9,PRIINI	; "SA CONTENTS IN ERROR"
	044554	104455			TRAP	C\$ERDF	
	044556	000023			.WORD	19	
	044560	030465			.WORD	EMSG9	
	044562	032740			.WORD	PRIINI	
3675	044564				CKLOOP		;LOOP ON ERROR?
	044564	104406			TRAP	C\$CLP1	
3676	044566				DODU	LOGUNT	;DROP UUT
	044566	013700	002312		MOV	LOGUNT,R0	
	044572	104451			TRAP	C\$DODU	
3677	044574				ESCAPE	TST	;LEAVE TST
	044574	104410			TRAP	C\$ESCAPE	
	044576	000306			.WORD	L10024-.	
3678							
3679	044600	005237	002316	5\$:	INC	INISTP	;ADJUST STEP COUNTER
3680	044604	062705	000002		ADD	#2,R5	;ADJUST TABLE INDEX
3681	044610	012737	000100	002326	6\$:	MOV	#100,OUTER
3682	044616	016537	002260	002314		MOV	CMPTBL(R5),SAEXP
3683	044624	012737	037200	002324	7\$:	MOV	#16000.,INNER
3684	044632	017464	000002	000012		MOV	@TKSA(R4),TKSASV(R4)
3685	044640	022705	000006		CMP	#6,R5	;GET SA CONTENTS
3686	044644	001005			BNE	8\$;ARE WE IN STEP 4?
3687	044646	033764	002314	000012		BNE	8\$
3688	044654	001024			BIT	SAEXP,TKSASV(R4)	;BRANCH IF NOT
3689	044656	000404			BNE	10\$;JUST LOOK FOR STEP 4 BIT
3690	044660	023764	002314	000012	8\$:	BR	9\$
3691	044666	001417			CMP	SAEXP,TKSASV(R4)	;IT'S SET SO LET'S GO
3692	044670	004737	036406	9\$:	BEQ	10\$;STAY IN LOOP OTHERWISE
3693	044674	005737	002330		JSR	PC,PDELAY	;IF SA IS WHAT WE EXPECT
3694	044700	001751			TST	TOUT	; THEN MOVE ALONG
3695					BEQ	7\$; ELSE GIVE UUT SOME TIME
3696	044702						;IF NO TIMEOUT YET
	044702	104455			ERRDF	20.,EMSG9,PRIINI	; THEN GO TAKE ANOTHER LOOK
	044704	000024			TRAP	C\$ERDF	
	044706	030465			.WORD	20	
	044710	032740			.WORD	EMSG9	
3697	044712				.WORD	PRIINI	
	044712	104406			CKLOOP		
3698	044714				TRAP	C\$CLP1	
	044714	013700	002312		DODU	LOGUNT	
	044720	104451			MOV	LOGUNT,R0	
3699	044722				TRAP	C\$DODU	
	044722	104410			ESCAPE	TST	
					TRAP	C\$ESCAPE	

```

044724 000160 .WORD L10024-.
3700
3701 044726 016574 002250 000002 10$: MOV STPTBL(R5),@TKSA(R4) ;WRITE NEXT STEP TO UUT
3702 044734 022705 000004 CMP #4,R5 ;IF STEP 3
3703 044740 001404 BEQ 15$ ; THEN DO PURGE/POLL STUFF
3704 044742 022705 000006 CMP #6,R5 ;IF NOT IN STEP 4
3705 044746 001314 BNE 5$ ; THEN GO BACK TO MAIN LOOP
3706 044750 000440 BR 20$ ; ELSE GO CHECK RESULTS
3707
3708 044752 15$: DELAY 1 ;GIVE PORT SOME TIME
044752 012727 000001 MOV #1,(PC)+
044756 000000 .WORD 0
044760 013727 002116 MOV L$DLY,(PC)+
044764 000000 .WORD 0
044766 005367 177772 DEC -6(PC)
044772 001375 BNE -.4
044774 005367 177756 DEC -22(PC)
045000 001367 BNE -.20
3709 045002 017464 000002 000012 MOV @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
3710 045010 001412 BEQ 16$ ;BRANCH IF OKAY
3711
3712 045012 ERRDF 21.,EMSG13,PRIINI ;SA NOT 0 IN PURGE/POLL
045012 104455 TRAP C$ERDF
045014 000025 .WORD 21
045016 030650 .WORD EMSG13
045020 032740 .WORD PRIINI
3713 045022 CKLOOP
045022 104406 TRAP C$CLP1
3714 045024 DODU LOGUNT
045024 013700 002312 MOV LOGUNT,R0
045030 104451 TRAP C$DODU
3715 045032 ESCAPE TST
045032 104410 TRAP C$ESCAPE
045034 000050 .WORD L10024-.
3716
3717 045036 012774 000000 000002 16$: MOV #0,@TKSA(R4) ;WRITE 0'S TO SA
3718 045044 005774 000000 TST @TKIP(R4) ;AND READ IP
3719 045050 000653 BR 5$ ;GO WAIT FOR NEXT TRANSITION
3720
3721 045052 004737 036562 20$: JSR PC,CHKCOM ;GO CHECK COMM AREA
3722 045056 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;HAS UUT BEEN DROPPED
3723 045064 001005 BNE T6EXT ;LEAVE NOW IF SO
3724 045066 005337 002272 DEC ITRCNi ;IF NO MORE ITERATIONS LEFT
3725 045072 001402 BEQ T6EXT ; THEN LEAVE TEST
3726 045074 000137 044412 JMP 2$ ; ELSE DO IT AGAIN
3727
3728 045100 T6EXT: EXIT TST
045100 104432 TRAP C$EXIT
045102 000002 .WORD L10024-.
3729
3730 045104 ENDTST
045104 L10024: TRAP C$ETST
045104 104401

```

```

3733          .SBTTL TEST 7: MAXIMUM RING BUFFER TEST
3734
3735 045106          BGNTST
    045106          T7::
3736
3737 045106 032764 000001 000014          BIT      #DRPFLG,LUNFLG(R4)          ;IF UUT NOT DROPPED
3738 045114 001423          BEQ      G07                          ; THEN DO TEST
3739 045116          RFLAGS   FLAGS                          ;ELSE GO GET SUPRVISOR FLAGS
    045116 104421          TRAP   C$RFLA
    045120 010037 002334          MOV      RO,FLAGS
3740 045124 032737 001000 002334          BIT      #PNT,FLAGS          ;SEE IF WE'RE PRINTING TEST NUMBERS
3741 045132 001412          BEQ      1$                          ;NO, DON'T PRINT BYPASSED
3742 045134          PRINTF  #BYPASS,L$TEST          ; ELSE PRINT THE TEST
    045134 013746 002114          MOV      L$TEST,-(SP)
    045140 C12746 047502          MOV      #BYPASS,-(SP)
    045144 012746 000002          MOV      #2,-(SP)
    045150 010600          MOV      SP,RO
    045152 104417          TRAP   C$PNTF
    045154 062706 000006          ADD     #6,SP
3743 045160          1$:          EXIT   TST                          ;BYPASSED MESSAGE AND GET OUT
    045160 104432          TRAP   C$EXIT
    045162 000526          .WORD  L10025-.
3744
3745 045164 012737 032640 002310          G07:   MOV      #CTRL,FRUIS          ;FAILING FRU IN CASE OF ERROR
3746 045172 012737 000001 002272          MOV      #1,ITRCNT          ;SET UP FOR ONE TEST ITERATION
3747 045200 022737 000001 002270          CMP     #1,PASCNT          ;IF FIRST PASS
3748 045206 001403          BEQ     2$                          ; THEN START TEST
3749 045210 012737 000012 002272          MOV     #10.,ITRCNT        ; ELSE DO 10 ITERATIONS
3750
3751 045216 012705 000000          2$:   MOV     #0,R5                          ;SET UP R5 AS INDEX TO STEP TABLES
3752 045222 012737 000001 002316          MOV     #1,INISTP          ;STEP 1 FOR ERROR PRINTOUT
3753 045230 016437 000004 002250          MOV     TKVEC(R4),STPTBL    ;PUT VECTOR IN STEP 1
3754 045236 006237 002250          ASR    STPTBL              ;DIVIDE BY TWO
3755 045242 006237 002250          ASR    STPTBL              ;DIVIDE BY FOUR
3756 045246 013737 002250 002264          MOV     STPTBL,CMPTBL+4    ;PUT VECTOR IN STEP 3 COMPARE
3757 045254 052737 137400 002250          BIS    #137400,STPTBL     ;REST OF STEP ONE
3758 045262 012737 005700 002260          MOV     #B.S1!B.QB!B.DI!B.OD!B.MP,CMPTBL
3759
3760 045270 012737 060050 002252          MOV     #COMMAR,STPTBL+2   ;STEP 1 COMPARE VALUE
3761 045276 012737 010277 002262          MOV     #010277,CMPTBL+2  ;STEP 2 - COMM AREA ADDRESS
3762 045304 012737 100000 002254          MOV     #B.PP,STPTBL+4    ;STEP 2 COMPARE
3763 045312 112737 000040 002265          MOV     #40,CMPTBL+5      ;STEP 3 - HIGH ADDRESS AND PRGE/POLL
3764 045320 012737 000000 002256          MOV     #0,STPTBL+6       ;REST OF STEP 3 COMPARE
3765 045326 012737 040000 002266          MOV     #040000,CMPTBL+6  ;STEP 4
3766
3767 045334 012737 001002 002306          MOV     #514.,CMARLG      ;LENGTH OF COMM AREA FOR THIS TEST
3768 045342 004737 036532          JSR    PC,BAKPAT          ;FILL COMM AREA WITH ALL 1'S DATA
3769
3770 045346 004737 036434          JSR    PC,STEP1          ;GO DO IT
3771 045352 005737 002320          TST    STEPST            ;IF STATUS OKAY
3772 045356 001412          BEQ    5$                          ; THEN CONTINUE TEST
3773
3774 045360          ERRDF  22.,EMSG9,PRIINI          ;"SA CONTENTS IN ERROR"
    045360 104455          TRAP   C$ERDF
    045362 000026          .WORD  22
    045364 030465          .WORD  EMSG9
    045366 032740          .WORD  PRIINI

```

```

3775 045370           CKLOOP           ;LOOP ON ERROR?
      045370 104406  TRAP C$CLP1
3776 045372           DODU LOGUNT           ;DROP UUT
      045372 013700 002312  MOV LOGUNT,R0
      045376 104451  TRAP C$DODU
3777 045400           ESCAPE TST           ;LEAVE TST
      045400 104410  TRAP C$ESCAPE
      045402 000306  .WORD L10025-
3778
3779 045404 005237 002316 5$: INC INISTP           ;ADJUST STEP COUNTER
3780 045410 062705 000002  ADD #2,R5           ;ADJUST TABLE INDEX
3781 045414 012737 000100 002326 6$: MOV #100,OUTER      ;SET UP FOR DELAY ROUTINE
3782 045422 016537 002260 002314  MOV CMPTBL(R5),SAEXP ;SET UP FOR COMPARE
3783 045430 012737 037200 002324 7$: MOV #16000,INNER    ;SET UP INNER
3784 045436 017464 000002 000012  MOV @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
3785 045444 022705 000006  CMP #6,R5           ;ARE WE IN STEP 4?
3786 045450 001005  BNE 8$             ;BRANCH IF NOT
3787 045452 033764 002314 000012  BIT SAEXP,TKSASV(R4) ;JUST LOOK FOR STEP 4 BIT
3788 045460 001024  BNE 10$           ;IT'S SET SO LET'S GO
3789 045462 000404  BR 9$            ;STAY IN LOOP OTHERWISE
3790 045464 023764 002314 000012 8$: CMP SAEXP,TKSASV(R4) ;IF SA IS WHAT WE EXPECT
3791 045472 001417  BEQ 10$          ; THEN MOVE ALONG
3792 045474 004737 036406 9$: JSR PC,PDELAY     ; ELSE GIVE UUT SOME TIME
3793 045500 005737 002330  TST TOUT           ;IF NO TIMEOUT YET
3794 045504 001751  BEQ 7$            ; THEN GO TAKE ANOTHER LOOK
3795
3796 045506           ERRDF 23.,EMSG9,PRIINI ;"SA CONTENTS IN ERROR'
      045506 104455  TRAP C$ERDF
      045510 000027  .WORD 23
      045512 030465  .WORD EMSG9
      045514 032740  .WORD PRIINI
3797 045516           CKLOOP
      045516 104406  TRAP C$CLP1
3798 045520           DODU LOGUNT
      045520 013700 002312  MOV LOGUNT,R0
      045524 104451  TRAP C$DODU
3799 045526           ESCAPE TST
      045526 104410  TRAP C$ESCAPE
      045530 000160  .WORD L10025-
3800
3801 045532 016574 002250 000002 10$: MOV STPTBL(R5),@TKSA(R4) ;WRITE NEXT STEP TO UUT
3802 045540 022705 000004  CMP #4,R5           ;IF STEP 3
3803 045544 001404  BEQ 15$          ; THEN DO PURGE/POLL STUFF
3804 045546 022705 000006  CMP #6,R5           ;IF NOT IN STEP 4
3805 045552 001314  BNE 5$            ; THEN GO BACK TO MAIN LOOP
3806 045554 000440  BR 20$           ; ELSE GO CHECK RESULTS
3807
3808 045556           DELAY 1           ;GIVE PORT SOME TIME
      045556 012727 000001  MOV #1,(PC)+
      045562 000000  .WORD 0
      045564 013727 002116  MOV L$DLY,(PC)+
      045570 000000  .WORD 0
      045572 005367 177772  DEC -6(PC)
      045576 001375  BNE -4
      045600 005367 177756  DEC -22(PC)
      045604 001367  BNE -20
3809 045606 017464 000002 000012  MOV @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS

```

```

3810 045614 001412          BEQ      16$          ;BRANCH IF OKAY
3811
3812 045616          ERRDF   24.,EMSG13,PRIINI ;SA NOT 0 IN PURGE/POLL
    045616 104455      TRAP    C$ERDF
    045620 000030      .WORD   24
    045622 030650      .WORD   EMSG13
    045624 032740      .WORD   PRIINI
3813 045626          CKLOOP
    045626 104406      TRAP    C$CLP1
3814 045630          DODU    LOGUNT
    045630 013700 002312  MOV    LOGUNT,RO
    045634 104451      TRAP    C$DODU
3815 045636          ESCAPE  TST
    045636 104410      TRAP    C$ESCAPE
    045640 000050      .WORD   L10025-.
3816
3817 045642 012774 000000 000002 16$:  MOV    #0,@TKSA(R4)      ;WRITE 0'S TO SA
3818 045650 005774 000000      TST    @TKIP(R4)      ;AND READ IP
3819 045654 000653          BR      5$          ;GO WAIT FOR NEXT TRANSITION
3820
3821 045656 004737 036562          JSR    PC,CHKCOM      ;GO CHECK COMM AREA
3822 045662 032764 000001 000014 20$:  BIT    #DRPFLG,LUNFLG(R4) ;HAS UUT BEEN DROPPED
3823 045670 001005          BNE    T7EXT         ;LEAVE NOW IF SO
3824 045672 005337 002272          DEC    ITRCNT        ;IF NO MORE ITERATIONS LEFT
3825 045676 001402          BEQ    T7EXT         ; THEN LEAVE TEST
3826 045700 000137 045216          JMP    2$          ; ELSE DO IT AGAIN
3827
3828 045704          T7EXT:  EXIT    TST
    045704 104432      TRAP    C$EXIT
    045706 000002      .WORD   L10025-.
3829
3830 045710          ENDTST
    045710          L10025:
    045710 104401      TRAP    C$ETST

```



```

3833          .SBTTL TEST 8: EXTENDED ADDRESS TEST
3834
3835 045712          BGNTST
      045712          T8::
3836
3837 045712 032764 000001 000014          BIT      #DRPFLG,LUNFLG(R4)          ;IF UUT NOT DROPPED
3838 045720 001423          BEQ      G08                          ; THEN DO TEST
3839 045722          RFLAGS   FLAGS                          ;ELSE GO GET SUPRVISOR FLAGS
      045722 104421          TRAP   C#RFLA
      045724 010037 002334          MOV     RO,FLAGS
3840 045730 032737 001000 002334          BIT     #PNT,FLAGS          ;SEE IF WE'RE PRINTING TEST NUMBERS
3841 045736 001412          BEQ     1$                          ;NO, DON'T PRINT BYPASSED
3842 045740          PRINTF  #BYPASS,L$TEST          ; ELSE PRINT THE TEST
      045740 013746 002114          MOV     L$TEST,-(SP)
      045744 012746 047502          MOV     #BYPASS,-(SP)
      045750 012746 000002          MOV     #2,-(SP)
      045754 010600          MOV     SP,RO
      045756 104417          TRAP   C#PNTF
      045760 062706 000006          ADD     #6,SP
3843 045764          1$:      EXIT     TST                          ;BYPASSED MESSAGE AND GET OUT
      045764 104432          TRAP   C$EXIT
      045766 000632          .WORD  L10026-.
3844
3845 045770 005737 002274          G08:   TST     KTFLAG          ;IF MEMORY MANAGEMENT AVAILABLE
3846 045774 001002          BNE     1$                          ; THEN DO TEST
3847 045776          EXIT     TST                          ; ELSE GET OUT
      045776 104432          TRAP   C$EXIT
      046000 000620          .WORD  L10026-.
3848 046002 012737 032640 002310          1$:   MOV     #CTRL,FRUIS          ;FAILING FRU IN CASE OF ERROR
3849 046010 012737 000001 002272          MOV     #1,IIRCNT          ;SFT UP FOR ONE TEST ITERATION
3850 046016 022737 000001 002270          CMP     #1,PASCNT          ;IF FIRST PASS
3851 046024 001403          BEQ     2$                          ; THEN START TEST
3852 046026 012737 000012 002272          MOV     #10.,IIRCNT        ; ELSE DO 10 ITERATIONS
3853
3854 046034 004737 036714          2$:   JSR     PC,INTMMU          ;INITIALIZE MMU REGISTERS
3855 046040 012705 000000          3$:   MOV     #0,R5                ;SET UP R5 AS INDEX TO STEP TABLES
3856 046044 012737 000001 002316          MOV     #1,INISTP          ;STEP 1 FOR ERROR PRINTOUT
3857 046052 016437 000004 002250          MOV     TKVEC(R4),STPTBL    ;PUT VECTOR IN STEP 1
3858 046060 006237 002250          ASR     STPTBL              ;DIVIDE BY TWO
3859 046064 006237 002250          ASR     STPTBL              ;DIVIDE BY FOUR
3860 046070 013737 002250 002264          MOV     STPTBL,CMPTBL*4     ;PUT VECTOR IN STEP 3 COMPARE
3861 046076 052737 111000 002250          BIS     #111000,STPTBL     ;REST OF STEP ONE
3862 046104 012737 005700 002260          MOV     #B.S1!B.QB!B.DI!B.OD!B.MP,CMPTBL
3863          ;STEP 1 COMPARE VALUE
3864 046112 012737 060050 002252          MOV     #COMMAR,STPTBL*2   ;STEP 2 - COMM AREA ADDRESS
3865 046120 042737 160000 002252          BIC     #BIT15!BIT14!BIT13,STPTBL*2
3866          ;CLEAR THE ACTIVE PAGE FIELD
3867 046126 012737 010222 002262          MOV     #010222,CMPTBL*2   ;STEP 2 COMPARE
3868 046134 013737 172346 002332          MOV     KPAR3,TEMP          ;GET RELOCATION VALUE
3869 046142 113737 002333 002254          MOV     TEMP*1,STPTBL*4    ;JUST THE HGH BYTE
3870 046150 006237 002254          ASR     STPTBL*4           ;MAKE IT THE EXTENDED
3871 046154 006237 002254          ASR     STPTBL*4           ; ADDRESS OF THE COMM AREA
3872 046160 052737 100000 002254          BIS     #B.PP,STPTBL*4     ;NOW SET PURGE/POLL BIT
3873 046166 112737 000040 002265          MOV     #40,CMPTBL*5       ;REST OF STEP 3 COMPARE
3874 046174 012737 000000 002256          MOV     #0,STPTBL*6        ;STEP 4
3875 046202 012737 040000 002266          MOV     #040000,CMPTBL*6   ;STEP 4 COMPARE
3876
    
```

```

3877 046210 012737 000022 002306      MOV    #18.,CMARLG      ;LENGTH OF COMM AREA FOR THIS TEST
3878 046216 004737 036532      JSR    PC,BAKPAT      ;FILL COMM AREA WITH ALL 1'S DATA
3879
3880 046222 004737 036434      JSR    PC,STEP1      ;GO DO IT
3881 046226 005737 002320      TST    STEPST        ;IF STATUS OKAY
3882 046232 001412      BEQ    5$            ; THEN CONTINUE TEST
3883
3884 046234      ERRDF  25 ,EMSG9,PRIINI ;"SA CONTENTS IN ERROR"
      046234 104455      TRAP  C$ERDF
      046236 000031      .WORD 25
      046240 030465      .WORD EMSG9
      046242 032740      .WORD PRIINI
3885 046244      CKLOOP
      046244 104406      TRAP  C$CLP1        ;LOOP ON ERROR?
3886 046246      DODU   LOGUNT
      046246 013700 002312      MOV    LOGUNT,RO     ;DROP UUT
      046252 104451      TRAP  C$DODU
3887 046254      ESCAPE TST            ;LEAVE TST
      046254 104410      TRAP  C$ESCAPE
      046256 000342      .WORD L10026-.
3888
3889 046260 005237 002316      5$:   INC    INISTP      ;ADJUST STEP COUNTER
3890 046264 062705 000002      ADD    #2,R5        ;ADJUST TABLE INDEX
3891 046270 012737 000100 002326 6$:   MOV    #100,OUTER   ;SET UP FOR DELAY ROUTINE
3892 046276 016537 002260 002314      MOV    CMPTBL(R5),SAEXP ;SET UP FOR COMPARE
3893 046304 012737 037200 002324 7$:   MOV    #16000.,INNER ;SET UP INNER
3894 046312 017464 000002 000012      MOV    @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
3895 046320 022705 000006      CMP    #6,R5        ;ARE WE IN STEP 4?
3896 046324 001005      BNE    8$            ;BRANCH IF NOT
3897 046326 033764 002314 000012      BIT    SAEXP,TKSASV(R4) ;JUST LOOK FOR STEP 4 BIT
3898 046334 001024      BNE    10$           ;IT'S SET SO LET'S GO
3899 046336 000404      BR     9$            ;STAY IN LOOP OTHERWISE
3900 046340 023764 002314 000012 8$:   CMP    SAEXP,TKSASV(R4) ;IF SA IS WHAT WE EXPECT
3901 046346 001417      BEQ    10$           ; THEN MOVE ALONG
3902 046350 004737 036406      9$:   JSR    PC,PDELAY    ; ELSE GIVE UUT SOME TIME
3903 046354 005737 002330      TST    TOUT         ;IF NO TIMEOUT YET
3904 046360 001751      BEQ    7$            ; THEN GO TAKE ANOTHER LOOK
3905
3906 046362      ERRDF  26 ,EMSG9,PRIINI ;"SA CONTENTS IN ERROR"
      046362 104455      TRAP  C$ERDF
      046364 000032      .WORD 26
      046366 030465      .WORD EMSG9
      046370 032740      .WORD PRIINI
3907 046372      CKLOOP
      046372 104406      TRAP  C$CLP1
3908 046374      DODU   LOGUNT
      046374 013700 002312      MOV    LOGUNT,RO
      046400 104451      TRAP  C$DODU
3909 046402      ESCAPE TST
      046402 104410      TRAP  C$ESCAPE
      046404 000214      .WORD L10026-.
3910
3911 046406 016574 002250 000002 10$:  MOV    STPTBL(R5),@TKSA(R4) ;WRITE NEXT STEP TO UUT
3912 046414 022705 000004      CMP    #4,R5        ;IF STEP 3
3913 046420 001404      BEQ    15$           ; THEN DO PURGE/POLL STUFF
3914 046422 022705 000006      CMP    #6,R5        ;IF NOT IN STEP 4
3915 046426 001314      BNE    5$            ; THEN GO BACK TO MAIN LOOP

```

```

3916 046430 000440 BR 20$ ; ELSE GO CHECK RESULTS
3917
3918 046432 15$: DELAY 1 ;GIVE PORT SOME TIME
046432 012727 000001 MOV #1,(PC)+
046436 000000 .WORD 0
046440 013727 002116 MOV L$DLY,(PC)+
046444 000000 .WORD 0
046446 005367 177772 DEC -6(PC)
046452 001375 BNE -.4
046454 005367 177756 DEC -22(PC)
046460 001367 BNE -.20
3919 046462 017464 000002 000012 MOV @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
3920 046470 001412 BEQ 16$ ;BRANCH IF OKAY
3921
3922 046472 ERRDF 27.,EMSG13,PRIINI ;SA NOT 0 IN PURGE/POLL
046472 104455 TRAP C$ERDF
046474 000033 .WORD 27
046476 030650 .WORD EMSG13
046500 032740 .WORD PRIINI
3923 046502 CKLOOP
046502 104406 TRAP C$CLP1
3924 046504 DODU LOGUNT
046504 013700 002312 MOV LOGUNT,R0
046510 104451 TRAP C$DODU
3925 046512 ESCAPE TST
046512 104410 TRAP C$ESCAPE
046514 000104 .WORD L10026-.
3926
3927 046516 012774 000000 000002 16$: MOV #0,@TKSA(R4) ;WRITE 0'S TO SA
3928 046524 005774 000000 TST @TKIP(R4) ;AND READ IP
3929 046530 000653 BR 5$ ;GO WAIT FOR NEXT TRANSITION
3930
3931 046532 004737 036562 20$: JSR PC,CHKCOM ;GO CHECK COMM AREA
3932 046536 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;HAS UUT BEEN DROPPED
3933 046544 001021 BNE T8EXT ;LEAVE NOW IF SO
3934
3935 046546 062737 002000 172346 ADD #2000,KPAR3 ;POINT TO NEXT 32KWORDS
3936 046554 103406 BCS 25$ ;DON'T ALLOW OVERFLOW IF 4 MBYTES
3937 046556 023737 002120 172346 CMP L$HIME,KPAR3 ;IF THERE'S NO MORE MEMORY AVAILABLE
3938 046564 103402 BLO 25$ ; THEN CHECK FOR MORE ITERATIONS
3939 046566 000137 046040 JMP 3$ ; ELSE DO IT AGAIN
3940
3941 046572 005037 177572 25$: CLR MMUSRO ;SHUT DOWN MEMORY MANAGEMENT
3942 046576 005337 002272 DEC ITRCNT ;IF NO MORE ITERATIONS LEFT
3943 046602 001402 BEQ T8EXT ; THEN LEAVE TEST
3944 046604 000137 046034 JMP 2$ ; ELSE DO IT AGAIN
3945
3946 046610 005037 177572 T8EXT: CLR MMUSRO ;MAKE SURE IT'S OFF
3947 046614 EXIT TST
046614 104432 TRAP C$EXIT
046616 000002 .WORD L10026-.
3948
3949 046620 ENDTST
046620 L10026: TRAP C$ETST
046620 104401

```

```

3952          .SBTTL TEST 9:GET DUST STATUS
3953
3954 046622    BGNTST
      046622    T9::
3955
3956 046622    032764 000001 000014    BIT      #DRPFLG,LUNFLG(R4)    ;IS THE DRIVE AVAILABLE
3957 046630    001423                    BEQ      G09                ;GO DO TEST IF AVAILABLE
3958 046632    104421                    RFLAGS  FLAGS                ;ELSE GO GET SUPRVISOR FLAGS
      046634    010037 002334            TRAP    C$RFLA
3959 046640    032737 001000 002334    MOV      RO,FLAGS
3960 046646    001412                    BIT      #PNT,FLAGS        ;SEE IF WE'RE PRINTING TEST NUMBERS
3961 046650                    BEQ      1$                ;NO, DON'T PRINT BYPASSED
      046650    013746 002114            PRINTF  #BYPASS,L$TEST      ; ELSE PRINT THE TEST
      046654    012746 047502            MOV      L$TEST,-(SP)
      046660    012746 000002            MOV      #BYPASS,-(SP)
      046664    010600                    MOV      #2,-(SP)
      046666    104417                    MOV      SP,RO
      046670    062706 000006            TRAP    C$PNTF
3962 046674                    ADD      #6,SP
      046674    104432                    EXIT     TST                ;BYPASSED MESSAGE AND GET OUT
      046676    000132                    TRAP    C$EXIT
      .WORD  L10027-.
3963
3964 046700    042764 000010 000014    G09:    BIC      #DUPFLG,LUNFLG(R4)    ;CLEAR DUP FLAG
3965 046706    012737 032640 002310    MOV      #CTRL,FRUIS      ;SET UP FRU POINTER
3966 046714    004737 037014                    JSR      PC,PRINT          ;GO DO A PORT INITIALIZE
3967 046720    032764 000001 000014    BIT      #DRPFLG,LUNFLG(R4)    ;IS THE DRIVE AVAILABLE
3968 046726    001036                    BNE     T9EXT             ;GET OUT IF NOT AVAILABLE
3969 046730    012705 002342            MOV      #SCTRLC,R5      ;SET UP TO DO THE SCC COMMAND
3970 046734    012737 000001 002272    MOV      #1,I TRCNT      ;DO IT ONCE
3971 046742    012737 032663 023042    MOV      #SCCCMD,CURCMD   ;SET UP COMMAND ASCII
3972 046750    004737 037246                    JSR      PC,CLSDRV        ;GO ISSUE THE COMMAND
3973 046754    032764 000001 000014    BIT      #DRPFLG,LUNFLG(R4)    ;IS THE DRIVE AVAILABLE
3974 046762    001020                    BNE     T9EXT             ;GET OUT IF NOT AVAILABLE
3975 046764    052764 000010 000014    BIS      #DUPFLG,LUNFLG(R4)    ;SET DUP FLAG FOR FOLLOWING COMMAND
3976 046772    012705 002456            MOV      #GDUST,R5       ;SET UP TO DO GET DUST STATUS COMMAND
3977 046776    012737 000001 002272    MOV      #1,I TRCNT
3978 047004    012737 032714 023042    MOV      #GDSCMD,CURCMD   ;SET UP COMMAND ASCII
3979 047012    004737 037246                    JSR      PC,CLSDRV        ;GO ISSUE THE COMMAND
3980 047016    042764 000010 000014    BIC      #DUPFLG,LUNFLG(R4)    ;CLEAR DUP FLAG
3981 047024                    T9EXT: EXIT     TST
      047024    104432                    TRAP    C$EXIT
      047026    000002                    .WORD  L10027-.
3982 047030                    ENDTST
      047030                    L10027: TRAP    C$ETST
      047030    104401

```

```

3984          .SBTTL TEST 10:EXECUTE LOCAL PROGRAM (Level II microdiagnostics)
3985
3986 047032    BGNTST
          047032    T10::
3987
3988 047032    032764 000001 000014    BIT      #DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
3989 047040    001423                    BEQ      GO10                    ;GO DO TEST IF AVAILABLE
3990 047042    104421                    RFLAGS  FLAGS                    ;ELSE GO GET SUPRVISOR FLAGS
          047044    010037 002334                    TRAP    C$RFLA
3991 047050    032737 001000 002334    MOV      RO,FLAGS
3992 047056    001412                    BIT      #PNT,FLAGS              ;SEE IF WE'RE PRINTING TEST NUMBERS
3993 047060    013746 002114                    BEQ      1$                      ;NO, DON'T PRINT BYPASSED
          047064    012746 047502                    PRINTF  #BYPASS,L$TEST           ; ELSE PRINT THE TEST
          047070    012746 000002                    MOV      L$TEST,-(SP)
          047074    010600                    MOV      #BYPASS,-(SP)
          047076    104417                    MOV      #2,-(SP)
          047100    062706 000006                    MOV      SP,RO
          047104    104432                    TRAP    C$PNTF
          047106    000372                    ADD      #6,SP
3994 047104    000372                    1$: EXIT    TST                      ;BYPASSED MESSAGE AND GET OUT
          047106    000372                    TRAP    C$EXIT
          .WORD  L10030-.
3995
3996 047110    042764 000010 000014    GO10: BIC      #DUPFLG,LUNFLG(R4)      ;CLEAR DUP FLAG
3997 047116    042764 000020 000014    BIC      #ABTFLG,LUNFLG(R4)      ;CLEAR ABORT FLAG
3998 047124    042764 000040 000014    BIC      #CNTRLC,LUNFLG(R4)      ;CLEAR CONTROL C FLAG
3999 047132    012737 032640 002310    MOV      #CTRL,FRUIS             ;SET UP FRU POINTER
4000 047140    004737 037014                    JSR     PC,PRINT                 ;GO DO A PORT INITIALIZE
4001 047144    032764 000001 000014    BIT      #DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
4002 047152    001150                    BNE     TSTXEX                   ;GET OUT IF NOT AVAILABLE
4003 047154    012705 002342                    MOV      #SCTRLC,R5              ;SET UP TO DO THE SCC COMMAND
4004 047160    012737 000001 002272    MOV      #1,I TRCNT              ;DO IT ONCE
4005 047166    012737 032663 023042    MOV      #SCCCMD,CURCMD           ;SET UP COMMAND ASCII
4006 047174    004737 037246                    JSR     PC,CLSDRV                ;GO ISSUE THE COMMAND
4007 047200    032764 000001 000014    BIT      #DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
4008 047206    001132                    BNE     TSTXEX                   ;GET OUT IF NOT AVAILABLE
4009 047210    012705 002406                    MOV      #ONLINE,R5              ;SET UP TO DO THE ONLINE COMMAND
4010 047214    012737 000001 002272    MOV      #1,I TRCNT              ;DO IT ONCE
4011 047222    012737 032670 023042    MOV      #ONLCMD,CURCMD           ;SET UP COMMAND ASCII
4012 047230    016465 000006 000004    MOV      MSCPUN(R4),P.UNIT(R5)    ;PUT THE UNIT NUMBER IN THE PACKET
4013 047236    004737 037246                    JSR     PC,CLSDRV                ;GO ISSUE THE COMMAND
4014 047242    032764 000001 000014    BIT      #DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
4015 047250    001111                    BNE     TSTXEX                   ;GET OUT IF NOT AVAILABLE
4016 047252    052764 000010 000014    BIS      #DUPFLG,LUNFLG(R4)      ;ALL FOLLOWING COMMANDS ARE DUP
4017 047260    012705 002476                    MOV      #XLOCPR,R5              ;SET UP TO DO ELP COMMAND
4018 047264    012737 000001 002272    MOV      #1,I TRCNT              ;DO IT ONCE
4019 047272    012737 032721 023042    MOV      #ELPCMD,CURCMD           ;SET UP COMMAND ASCII
4020 047300    004737 037246                    JSR     PC,CLSDRV                ;GO ISSUE THE COMMAND
4021 047304    032764 000001 000014    BIT      #DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
4022 047312    001070                    BNE     TSTXEX                   ;GET OUT IF NOT AVAILABLE
4023 047314    032737 000200 177560    1$: BIT      #BIT7,RCSR            ;CHECK FOR INPUT FROM KEYBOARD
4024 047322    001413                    BEQ      2$                      ;IF NONE, CONTINUE
4025 047324    013705 177562                    MOV      RBUF,R5                 ;GET DATA INPUT FROM KEYBOARD
4026 047330    042705 000200                    BIC      #BIT7,R5                 ;STRIP PARITY
4027 047334    022705 000003                    CMP      #3,R5                    ;CHECK FOR "CONTROL C" INPUT
4028 047340    001004                    BNE     2$                      ;IF NOT, CONTINUE
4029 047342    052764 000040 000014    BIS      #CNTRLC,LUNFLG(R4)      ;SET "CONTROL C" FLAG

```

```

4030 047350 000432          BR      20$          ;GO ABORT MICRODIAGNOSTICS
4031 047352 012705 002524 2$:    MOV     #RCVDAT,R5      ;SET UP TO DO RECEIVE DATA COMMAND
4032 047356 012737 000001 002272  MOV     #1,ITRCNT      ;DO IT ONCE
4033 047364 012737 032726 023042  MOV     #RCVCMD,CURCMD ;SET UP COMMAND ASCII
4034 047372 004737 037246          JSR     PC,CLSDRV      ;GO ISSUE THE COMMAND
4035 047376 032764 000001 000014  BIT     #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
4036 047404 001033          BNE     TSTXEX        ;GET OUT IF NOT AVAILABLE
4037 047406 032764 000020 000014  BIT     #ABTFLG,LUNFLG(R4) ;ABORT LOCAL PROGRAM ?
4038 047414 001010          BNE     20$          ;YES, ISSUE ABORT COMMAND
4039 047416 005002          CLR     R2            ;NO, SET UP DELAY LOW COUNT
4040 047420 012703 000020          MOV     #20,R3        ;SET UP DELAY HIGH COUNT
4041 047424 005202          10$:   INC     R2            ;DELAY
4042 047426 001376          BNE     10$          ;
4043 047430 005303          DEC     R3            ;
4044 047432 001374          BNE     10$          ;
4045 047434 000727          BR      1$            ;POLL DIAGL2 PROGRESS
4046 047436 012705 002554 20$:   MOV     #ABORT,R5      ;SET UP TO DO ABORT COMMAND
4047 047442 012737 000001 002272  MOV     #1,ITRCNT      ;DO IT ONCE
4048 047450 012737 032733 023042  MOV     #ABTCMD,CURCMD ;SET UP COMMAND ASCII
4049 047456 004737 037246          JSR     PC,CLSDRV      ;GO ISSUE THE COMMAND
4050 047462 032764 000040 000014  BIT     #CNTRLC,LUNFLG(R4) ;CHECK IF ABORTING DUE TO "CONTROL C"
4051 047470 001401          BEQ     TSTXEX        ;IF NOT, EXIT
4052 047472          BREAK
4053 047474          TRAP   C$BRK
4054 047500          TSTXEX: EXIT
4055 047502          TRAP   C$EXIT
4056 047504          .WORD L10030-.
4057 047506          .ENDTST
4058 047508          L10030: TRAP   C$ETST
4059 047510          BYPASS: .ASCIZ /%N%A TEST %Z3%A BYPASSED%N/
4060 047512          .EVEN
4061 047514          ENDMOD
4062 047516          .TITLE PARAMETER CODING
4063 047518          .SBTTL HARDWARE PARAMETER CODING SECTION
4064 047520          BGNMOD
4065 047522          ;++
4066 047524          ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
4067 047526          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
4068 047528          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
4069 047530          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
4070 047532          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
4071 047534          ; WITH THE OPERATOR.
4072 047536          ;--
4073 047538          BGNHRD
4074 047540          .WORD L10031-L$HARD/2
4075 047542          L$HARD:
4076 047544          GPRMA TKIPAD,0,0,160002,177564,YES
4077 047546          .WORD T$CODE
4078 047548          .WORD TKIPAD
4079 047550          .WORD T$LOLIM
    
```

```

4119 047546 177564 .WORD T$HILIM
      047550 GPRMD TKVECT,2,0,777,60,776,YES
      047550 001032 .WORD T$CODE
      047552 047613 .WORD TKVECT
      047554 000777 .WORD 777
      047556 000060 .WORD T$LLOLIM
      047560 000776 .WORD T$HILIM
4120 047562 GPRMD TKUNT,4,0,777,0,251,YES
      047562 002032 .WORD T$CODE
      047564 047625 .WORD TKUNT
      047566 000777 .WORD 777
      047570 000000 .WORD T$LLOLIM
      047572 000251 .WORD T$HILIM
4121
4122 047574 EXIT HRD
      047574 026004 .WORD T$CODE
4123
4124 047576 124 113 111 TKIPAD: .ASCIZ ?TKIP ADDRESS?
4125 047613 124 113 040 TKVECT: .ASCIZ ?TK VECTOR?
4126 047625 124 057 115 TKUNT: .ASCIZ ?T/MSCP UNIT NUMBER?
4127 .EVEN
4128
4129
4130 047650 ENDHRD
      .EVEN
      047650 L10031:
4131
4138

```

```

4141      .SBTTL  SOFTWARE PARAMETER CODING SECTION
4142
4143      ;**
4144      ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
4145      ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
4146      ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
4147      ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
4148      ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
4149      ; WITH THE OPERATOR.
4150      ;--
4151
4152 047650      BGNSFT
          047650 000000      .WORD L10032-L$SOFT/2
          047652      L$SOFT::

4153
4160
4161      .EVEN
4162
4163 047652      ENDSFT
          047652      .EVEN
          L10032:

4164
4165
4175      ;*****
4176      ;*****
4177
4178      ;COMMUNICATIONS AREA
4179      ; THIS IS THE COMMUNICATIONS AREA THAT IS USED
4180      ; THROUGHOUT THE PROGRAM IN TESTING THE PERMUTATIONS
4181      ; OF THE UQ-PORT INIT SEQUENCE.  IT IS ESSENTIAL THAT
4182      ; THIS AREA RESIDE IN AN 8KBYTE AREA OF MEMORY FREE
4183      ; OF DIAGNOSTIC CODE SO THAT IT MAY BE SUCCESSFULLY
4184      ; RELOCATED THROUGHOUT UPPER MEMORY VIA MEMORY MAN-
4185      ; AGEMENT.
4186      ;
4187      ;*****
4188      ;*****
4189
4190      060000      .=60000      ;START OF THE THIRD 8KBYTE BLOCK
4191      ;OF VIRTUAL MEMORY SPACE.  ACCESSIBLE
4192      ;VIA PAR/PDR 2.
4193 060000      RDBUF::
4194 060000      COMMBF::
4195 060000      .BLKW 20.      ;BUFFER SPACE PRECEDING COMM AREA
4196 060050      COMMAR::
4197 060050      .BLKW 514.      ;MAXIMUM COMM AREA LENGTH
4198 062054      LASTBF::
4199 062054      .BLKW 20.      ;BUFFER SPACE SUCCEEDING COMM AREA
4200
4201 062124      .BLKW 7638.
4205
4206 120000      LASTAD
          120000 000000      .EVEN
          120002 000000      .WORD 0
          120004      .WORD 0
4207 120004      L$LAST::
          ENDMOD

```


4208

000001

.END

PARAMETER CODING
Symbol table

ABORT 002554
 ABTCMD 032733 G
 ABTFLG= 000020 G
 ADR = 000020 G
 ASSEMB= 000010
 BAKPAT 036532 G
 BIT0 = 000001 G
 BIT00 = 000001 G
 BIT01 = 000002 G
 BIT02 = 000004 G
 BIT03 = 000010 G
 BIT04 = 000020 G
 BIT05 = 000040 G
 BIT06 = 000100 G
 BIT07 = 000200 G
 BIT08 = 000400 G
 BIT09 = 001000 G
 BIT1 = 000002 G
 BIT10 = 002000 G
 BIT11 = 004000 G
 BIT12 = 010000 G
 BIT13 = 020000 G
 BIT14 = 040000 G
 BIT15 = 100000 G
 BIT2 = 000004 G
 BIT3 = 000010 G
 BIT4 = 000020 G
 BIT5 = 000040 G
 BIT6 = 000100 G
 BIT7 = 000200 G
 BIT8 = 000400 G
 BIT9 = 001000 G
 BOE = 000400 G
 BPNERR 023342
 BRFLAG= 000004 G
 BYPASS 047502 G
 B.DI = 000400 G
 B.ER = 100000 G
 B.GO = 000001 G
 B.IE = 000200 G
 B.LF = 000002 G
 B.MP = 000100 G
 B.NV = 002000 G
 B.OD = 000200 G
 B.PI = 000001 G
 B.PP = 100000 G
 B.QB = 001000 G
 B.S1 = 004000 G
 B.S2 = 010000 G
 B.S3 = 020000 G
 B.S4 = 040000 G
 B.WR = 040000 G
 CDRECV 037746 G
 CHKCOM 036562 G
 CKCMEX 036712
 CLSDRV 037246 G
 CMARLG 002306 G

CMDCNT 003034 G
 CMDREF 003032 G
 CMDRNG 003014 G
 CMDSAV 023036 G
 CMMERR 002302 G
 CMPTBL 002260 G
 CMTBLG 002304 G
 CNTER = 000000 G
 CNTERR 023322
 CNTFLG 003026 G
 CNTHI 003024 G
 CNTRLC= 000040 G
 COMMAR 060050 G
 COMMBF 060000 G
 CONID = 177777 G
 CRD = 177776 G
 CTRL 032640 G
 CURCMD 023042 G
 C\$AU = 000052
 C\$AUTO= 000061
 C\$BRK = 000022
 C\$BSEG= 000004
 C\$BSUB= 000002
 C\$CLCK= 000062
 C\$CLEA= 000012
 C\$CLOS= 000035
 C\$CLP1= 000006
 C\$CPBF= 000074
 C\$CPME= 000075
 C\$CVEC= 000036
 C\$DCLN= 000044
 C\$DODU= 000051
 C\$DRPT= 000024
 C\$DU = 000053
 C\$EDIT= 000000
 C\$ERDF= 000055
 C\$ERHR= 000056
 C\$ERRQ= 000060
 C\$ERSF= 000054
 C\$ERSO= 000057
 C\$ESCA= 000010
 C\$ESEG= 000005
 C\$ESUB= 000003
 C\$ETST= 000001
 C\$EXIT= 000032
 C\$FREQ= 000101
 C\$FRME= 000100
 C\$GETB= 000026
 C\$GETW= 000027
 C\$GMAN= 000043
 C\$GPHR= 000042
 C\$GPRI= 000040
 C\$INIT= 000011
 C\$INLP= 000020
 C\$MANI= 000050
 C\$MAP = 000102
 C\$MEM = 000031

C\$MMU = 000103
 C\$MSG = 000023
 C\$OPNR= 000034
 C\$OPNW= 000104
 C\$PNTB= 000014
 C\$PNTF= 000017
 C\$PNTS= 000016
 C\$PNTX= 000015
 C\$PUTB= 000072
 C\$PUTW= 000073
 C\$QIO = 000377
 C\$RDBU= 000007
 C\$REFG= 000047
 C\$REL = 000077
 C\$RESE= 000033
 C\$REVI= 000004
 C\$RFLA= 000021
 C\$RPT = 000025
 C\$SEFG= 000046
 C\$SPRI= 000041
 C\$SVEC= 000037
 C\$TOME= 000076
 DCERR 023312
 DFPTBL 002202 G
 DIAGMC= 000000
 DRPFLG= 000001 G
 DRVE 032655 G
 DRIVER = 000011 G
 DSCEND 003024 G
 DSCRNG 003000 G
 DUPFLG= 000010 G
 DUSTFL= 000017 G
 EF.CON= 000036 G
 EF.NEW= 000035 G
 EF.PWR= 000034 G
 EF.RES= 000037 G
 EF.STA= 000040 G
 ELPCMD 032721 G
 MSG10 030512 G
 MSG11 030540 G
 MSG12 030601 G
 MSG13 030650 G
 MSG14 030677 G
 MSG15 030726 G
 MSG5 030334 G
 MSG6 030355 G
 MSG7 030406 G
 MSG8 030427 G
 MSG9 030465 G
 END 041016
 ENDCLE 041064
 ERR = 100000 G
 ERRBLK 023060 G
 ERRMSG 023056 G
 ERRNBR 023054 G
 ERRTP 023052 G
 EVL = 000004 G

EXIT 040564
 EXTINT 036070
 EXTVEC 036404
 E\$END = 002100
 E\$LOAD= 000035
 FLAG = 040000 G
 FLAGS 002334 G
 FRUIS 002310 G
 F\$AU = 000015
 F\$AUTO= 000020
 F\$BGN = 000040
 F\$CLEA= 000007
 F\$DU = 000016
 F\$END = 000041
 F\$HARD= 000004
 F\$HW = 000013
 F\$INIT= 000006
 F\$JMP = 000050
 F\$MOD = 000000
 F\$MSG = 000011
 F\$PROT= 000021
 F\$PWR = 000017
 F\$RPT = 000012
 F\$SEG = 000003
 F\$SOFT= 000005
 F\$SRV = 000010
 F\$SUB = 000002
 F\$SW = 000014
 F\$TEST= 000001
 GDSCMD 032714 G
 GDUST 002456
 GO = 000001 G
 GOABO 040554
 GO10 047110
 GO2 041702
 GO3 042464
 GO4 043136
 GO5 043716
 GO6 044360
 GO7 045164
 GO8 045770
 GO9 046700
 G\$CNT0= 000200
 G\$DELM= 000372
 G\$DISP= 000003
 G\$EXCP= 000400
 G\$HILI= 000002
 G\$LOLI= 000001
 G\$NO = 000000
 G\$OFFS= 000400
 G\$OFFSI= 000376
 G\$PRMA= 000001
 G\$PRMD= 000002
 G\$PRML= 000000
 G\$RADA= 000140
 G\$RADB= 000000
 G\$RADD= 000040

G\$RADL= 000120
 G\$RADO= 000020
 G\$XFER= 000004
 G\$YES = 000010
 HBFADD 023362
 HELP = 000000
 HIADDR= 000002 G
 HOE = 100000 G
 HSTIMO= 000000 G
 IBE = 010000 G
 IDU = 000040 G
 IER = 020000 G
 ILLINT 036072 G
 ILOOP 037044
 IMM = 000200 G
 IMSG 041022
 INISTP 002316 G
 INNER 002324 G
 INTFLG= 000002 G
 INTMMU 036714 G
 INTRCV 036062 G
 INTTBL 037236
 INVSTA 023332
 ISR = 000100 G
 ITRCNT 002272 G
 IXE = 004000 G
 I\$AU = 000041
 I\$AUTO= 000041
 I\$CLN = 000041
 I\$DU = 000041
 I\$HRD = 000041
 I\$INIT= 000041
 I\$MOD = 000041
 I\$MSG = 000041
 I\$PROT= 000040
 I\$PTAB= 000041
 I\$PWR = 000041
 I\$RPT = 000041
 I\$SEG = 000041
 I\$SETU= 000041
 I\$SFT = 000041
 I\$SRV = 000041
 I\$SUB = 000041
 I\$TST = 000041
 J\$JMP = 000167
 KPAR0 = 172340 G
 KPAR1 = 172342 G
 KPAR2 = 172344 G
 KPAR3 = 172346 G
 KPAR4 = 172350 G
 KPAR5 = 172352 G
 KPAR6 = 172354 G
 KPAR7 = 172356 G
 KPDR0 = 172300 G
 KPDR1 = 172302 G
 KPDR2 = 172304 G
 KPDR3 = 172306 G

PARAMETER CODING
Symbol table

MACRO Y05.02 Tuesday 16-Apr-85 08:44 Page 67-3

SEQ 114

KPDR4 = 172310 G	L\$LUN 002074 G	L2ELDF= 000156 G	L21 024417 G	MSCPER 040474
KPDR5 = 172312 G	L\$MREV 002050 G	L2ELDS= 000166 G	L210 025073 G	MSCPUN= 000006 G
KPDR6 = 172314 G	L\$NAME 002000 G	L2ELEC= 000140 G	L211 025137 G	MSCPVR= 000000 G
KPDR7 = 172316 G	L\$PRIO 002042 G	L2ELEV= 000136 G	L212 025203 G	MSGLEN= 177774 G
KTEXT 036252	L\$PROT 023044 G	L2ELFL= 000135 G	L213 025247 G	NEXT 040714
KTFLAG 002274 G	L\$PRT 002112 G	L2ELFM= 000134 G	L214 025313 G	NOKT 036246
KTTEST 036102 G	L\$REPP 002062 G	L2ELLB= 000162 G	L215 025357 G	NUPASS 040700
LASTBF 062054 G	L\$REV 002010 G	L2ELOF= 000172 G	L216 025423 G	ONEFIL= 000001
LINE1 023420 G	L\$RPT 035002 G	L2ELPB= 000160 G	L219 025467 G	ONLCMD 032670 G
LINE2 023454 G	L\$SOFT 047652 G	L2ELP1= 000146 G	L22 024473 G	ONLINE 002406
LINE3 023534 G	L\$SPC 002056 G	L2ELP2= 000150 G	L220 025533 G	OP.ABT= 000006 G
LINE4 023564 G	L\$SPCP 002020 G	L2ELRL= 000144 G	L221 025601 G	OP.ELP= 000003 G
LINE5 023626 G	L\$SPTP 002024 G	L2ELRT= 000145 G	L222 025646 G	OP.END= 000200 G
LINE6 023703 G	L\$STA 002030 G	L2ELRW= 000170 G	L223 025710 G	OP.GDS= 000001 G
LINE7 023746 G	L\$SW 002212 G	L2ELST= 000154 G	L224 025752 G	OP.ONL= 000011 G
LOE = 040000 G	L\$TEST 002114 G	L2ELTN= 000157 G	L225 026011 G	OP.REC= 000005 G
LOGUNT 002312 G	L\$TIML 002014 G	L2ELT0= 000163 G	L226 026053 G	OP.SCC= 000004 G
LOOP 037034	L\$UNIT 002012 G	L2ELT1= 000164 G	L227 026115 G	OUTER 002326 G
LOT = 000010 G	L10000 002210	L2ELT2= 000165 G	L228 026154 G	OWN = 100000 G
LUNBLK 002212 G	L10001 002212	L2ER1 030763 G	L229 026213 G	O\$APTS= 000000
LUNFLG= 000014 G	L10003 033542	L2ER10 031211 G	L23 024531 G	O\$AU = 000000
L\$ACP 002110 G	L10004 035000	L2ER11 031232 G	L230 026255 G	O\$BGNR= 000001
L\$APT 002036 G	L10005 036052	L2ER12 031246 G	L231 026317 G	O\$BGNS= 000000
L\$AU 041120 G	L10006 036060	L2ER13 031266 G	L232 026361 G	O\$DU = 000001
L\$AUT 002070 G	L10007 036070	L2ER14 031306 G	L233 026423 G	O\$ERRT= 000001
L\$AUTO= ***** GX	L10010 036100	L2ER15 031334 G	L234 026462 G	O\$GNSW= 000000
L\$CCP 002106 G	L10011 041052	L2ER16 031364 G	L235 026524 G	O\$POIN= 000001
L\$CLEA 041054 G	L10012 041102	L2ER17 031412 G	L236 026566 G	O\$SETU= 000000
L\$CO 002032 G	L10013 041116	L2ER18 031430 G	L237 026630 G	PAROFF 002300 G
L\$DEPO 002011 G	L10014 041124	L2ER19 031451 G	L238 026672 G	PASCNT 002270 G
L\$DESC 002150 G	L10015 041622	L2ER2 031003 G	L239 026731 G	PCKSIZ 003030 G
L\$DESP 002076 G	L10016 041312	L2ER20 031467 G	L24 024571 G	PDELAY 036406 G
L\$DEVP 002060 G	L10017 041556	L2ER21 031551 G	L240 026770 G	PDLYEX 036432
L\$DISP 002124 G	L10020 042404	L2ER22 031631 G	L241 027027 G	PDRECV 040054 G
L\$DLY 002116 G	L10021 043056	L2ER23 031720 G	L242 027071 G	PNT = 001000 G
L\$DTP 002040 G	L10022 043636	L2ER24 032010 G	L244 027130 G	PRI = 002000 G
L\$DTYP 002034 G	L10023 044300	L2ER25 032076 G	L245 027224 G	PRIDAT 033074 G
L\$DU 041104 G	L10024 045104	L2ER26 032167 G	L246 027320 G	PRIERR 033146 G
L\$DUT 002072 G	L10025 045710	L2ER27 032261 G	L247 027414 G	PRIEX 033536
L\$DVTY 023412 G	L10026 046620	L2ER3 031023 G	L248 027510 G	PRIINI 032740 G
L\$EF 002052 G	L10027 047030	L2ER4 031040 G	L249 027604 G	PRIIP 033122 G
L\$ENVI 002044 G	L10030 047500	L2ER5 031057 G	L25 024631 G	PRIPAD 033022 G
L\$ERRT 023052 G	L10031 047650	L2ER6 031107 G	L250 027700 G	PRISA 032764 G
L\$ETP 002102 G	L10032 047652	L2ER7 031127 G	L251 027774 G	PRIVAD 033046 G
L\$EXP1 002046 G	L2BRD1= 000024 G	L2ER8 031145 G	L252 030070 G	PRI00 = 000000 G
L\$EXP4 002064 G	L2BRD2= 000030 G	L2ER9 031160 G	L253 030164 G	PRI01 = 000040 G
L\$EXP5 002066 G	L2BWR1= 000014 G	L2ETBL 023062 G	L254 030260 G	PRI02 = 000100 G
L\$HARD 047540 G	L2BWR2= 000020 G	L2MSG 023312	L26 024671 G	PRI03 = 000140 G
L\$HIME 002120 G	L2CMD = 000010 G	L2REP1= 000050 G	L27 024731 G	PRI04 = 000200 G
L\$HPCP 002016 G	L2CRC1= 000044 G	L2REP2= 000052 G	L28 024771 G	PRI05 = 000240 G
L\$HPTP 002022 G	L2CRC2= 000046 G	L2RSP = 000011 G	L29 025026 G	PRI06 = 000300 G
L\$HW 002202 G	L2DATA 040200 G	L2STA = 000002 G	MMON = 000001 G	PRI07 = 000340 G
L\$ICP 002104 G	L2DRV = 000004 G	L2SWR1= 000034 G	MMUSRO= 177572 G	PRTDRV 037604 G
L\$INIT 040574 G	L2DUMP 033544 G	L2SWR2= 000036 G	MMUSR1= 177574 G	PRTINT 037014 G
L\$LADP 002026 G	L2ECC1= 000040 G	L2TRK = 000007 G	MMUSR2= 177576 G	P.BCNT= 000014 G
L\$LAST 120004 G	L2ECC2= 000042 G	L2TST = 000006 G	MMUSR3= 172516 G	P.BUFF= 000020 G
L\$LOAD 002100 G	L2ELDE= 000155 G	L2UNT = 000012 G	MM220N= 000020 G	P.CRF = 000000 G

PARAMETER CODING
Symbol table

MACRO Y05.02 Tuesday 16-Apr-85 08:44 Page 67-4

SEQ 115

P.EXT1= 000014 G	START 040646	T\$ERRN= 000033	T\$\$RPT= 010005	WRBUF 003036 G
P.EXT2= 000015 G	STEPST 002320 G	T\$EXCP= 000000	T\$\$SOF= 010032	WRCMD 032702 G
P.EXT3= 000016 G	STEP1 036434 G	T\$FLAG= 000041	T\$\$SRV= 010010	WRCMDE 033256 G
P.FLGS= 000011 G	STPTBL 002250 G	T\$GMAN= 000000	T\$\$SUB= 010017	WRDATA 002322 G
P.IND1= 000020 G	STP1ER 036524	T\$HILI= 000251	T\$\$SW = 010001	WRER1 032347 G
P.IND2= 000022 G	STP1EX 036530	T\$LAST= 000001	T\$\$TES= 010030	WRER2 032401 G
P.MOD = 000012 G	SVCGBL= 000000	T\$LOLI= 000000	T1 041126 G	WRER3 032422 G
P.OPCD= 000010 G	SVCINS= 000000	T\$LSYM= 010000	T1.1 041154	WRER4 032450 G
P.STS = 000012 G	SVCSUB= 000000	T\$LTNO= 000012	T1.2 041340	WRER5 032474 G
P.UNIT= 000004 G	SVCTAG= 000000	T\$NEST= 177777	T10 047032 G	WRER6 032525 G
RBUF = 177562 G	SVCTST= 000000	T\$NSO = 000000	T2 041624 G	WRER7 032546 G
RCSR = 177560 G	S\$LSYM= 010000	T\$NS1 = 000005	T2EXT 042400	WRER8 032614 G
RCVCMO 032726 G	S1 = 004000 G	T\$NS2 = 000002	T3 042400 G	WREX 033512 G
RCVDAT 002524	TEMP 002332 G	T\$PTNU= 000000	T3EXT 043000	WRINTO 033176 G
RDBUF 060000 G	TF.BLK= 000010 G	T\$SAVL= 177777	T4 043000 G	WRPRTE 033222 G
RDCMD 032707 G	TIMOUT= 000024 G	T\$SEGL= 177777	T4EXT 043626	WRSEQE 033410 G
RESPBF 002570 G	TKIP = 000000 G	T\$SUBN= 000000	T5 043640 G	WRTOE 033466 G
REWCMO 032675 G	TKIPAD 047576	T\$TAGL= 177777	T5EXT 044270	WR1 023772 G
RNGSTP= 000004 G	TKIPSV= 000010 G	T\$TAGN= 010033	T6 044302 G	WR2 024035 G
RSPADD 023352	TKSA = 000002 G	T\$TEMP= 000000	T6EXT 045100	WR3 024071 G
RSPBUF 002574 G	TKSASV= 000012 G	T\$TEST= 000012	T7 045106 G	WR4 024122 G
RSPEND 003004 G	TKUNT 047625	T\$TSTM= 177777	T7EXT 045704	WR5 024147 G
RSPRNG 003004 G	TKVEC = 000004 G	T\$TSTS= 000001	T8 045712 G	WR6 024211 G
RSPSAV 023040 G	TKVECT 047613	T\$\$AU = 010014	T8EXT 046610	WR7 024214 G
RSPSTP= 000104 G	TOUT 002330 G	T\$\$CLE= 010012	T9 046622 G	WR8 024265 G
RSPTO 023402	TRAP4 036054 G	T\$\$DU = 010013	T9EXT 047024	WR9 024342 G
RSTVEC 036266 G	TRP4FG 002276 G	T\$\$HAR= 010031	T9FLAG= ***** GX	XLOCPR 002476
SAEXP 002314 G	TSTXEX 047474	T\$\$HW = 010000	UAM = 000200 G	X\$ALWA= 000000
SCCCMD 032663 G	TXFER = 000005 G	T\$\$INI= 010011	UNERLG 023372	X\$FALS= 000040
SCTRLC 002342	T\$ARGC= 000002	T\$\$MSG= 010004	VECTOR 036316 G	X\$OFFS= 000400
SFPTBL 002212 G	T\$CODE= 026004	T\$\$PRO= 010002	VEC4 = 000004 G	X\$TRUE= 000020

. ABS. 120004 000 (RW,I,GBL,ABS,OVR)
 000000 001 (RW,I,LCL,REL,CON)
 Errors detected: 0

*** Assembler statistics

Work file reads: 337
 Work file writes: 334
 Size of work file: 34208 Words (134 Pages)
 Size of core pool: 19714 Words (75 Pages)
 Operating system: RSX-11M/PLUS (Under VAX/VMS)

Elapsed time: 00:08:41.85
 CZTKAA.BIN,CZTKAA/-SP-SVC40R.MLB/ML,CZTKAA