

TK50 TK70

TK FRONT END FUNC
CZTKAF0

AH-T772F-MC
1 OF 1 JUN 1992
COPYRIGHT © 1984, 92

digital™
MADE IN USA

The main body of the document is a large grid of 150 small, illegible data tables or charts, arranged in 10 columns and 15 rows. Each cell in the grid contains a small, structured piece of information, likely a data point or a small table. The text is too small to be read, but the layout suggests a comprehensive data set or a series of related metrics.

B1

d nD w

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

.REM @

IDENTIFICATION

PRODUCT CODE:	AC - T771F - MC
PRODUCT NAME:	CZTKAFO TKxx FRONT END FUNCTIONAL
PRODUCT DATE:	June 1991
MAINTAINER:	TAPE AND OPTICAL DIAGNOSTIC ENGINEERING
AUTHOR:	Brian LeBlanc

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) 1985,1991 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

C1

PARAMETER CODING

MACRO V05.05 Wednesday 27-Feb-91 11:55 Page 2

SEQ 0002

46
47
48
49
50
51
52
53
54
55
56

REVISION HISTORY

APRIL 1985	NEW RELEASE
JUNE 1985	REVISION B ADDED CODE SO THAT PROGRAM CAN RUN ON PDP - 11 UNIBUS FAMILY CPUS.
FEBRUARY 1991	REVISION F ADDED CODE TO CORRECT PROBLEM WITH 4 MBYTE MEMORY.

58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114

1 GENERAL INFORMATION

1.1 Product Description

The TKxx Functional Diagnostic is intended to provide confidence in the basic functionality of the TKxx subsystem. As such, this should be the first host level diagnostic run on the TKxx 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 TKxx'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 maintenance

1.3 Performance Goals

This program will test up to four TKxx'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 TKxx; 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.

115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168

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 family CPU
3. 28 KW memory
4. Console Terminal
5. Load Device
6. 1 to 4 TKxx tape drives with controllers
7. 1 to 4 TKxx scratch cartridges (optional)
8. LCP-5 IFD software (optional)

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
209
210
211
212
213
214
215
216
217
218
219

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 TKxx'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.

221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277

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
```

```

278
279      **** LEVEL 2 MICRODIAGNOSTIC DUMP ****
280
281      Program Status      *****
282      Drive Status        ***
283      Test Number         ***
284      Track Number        ***
285      TMSCP Command       ***
286      TMSCP Response Flag ***
287      TMSCP Unit Flags    *****
288
289      Error Log (Tape Transfer Error)
290      Error Format          ***
291      Error Flags          ***
292      Event Code           *****
293      Error Retry Level    ***
294      Number of Retries    ***
295      Position (Low Order) *****
296      Position (High Order) *****
297      Controller Status    ***
298      Drive Error Code     ***
299      Drive Flags          ***
300      Track number         ***
301      Physical Block Number *****
302      Logical Block Number ***
303      Tape Count 0         ***
304      Tape Count 1         ***
305      Tape Count 2         ***
306      Drive State          *****
307      Read/Write State     *****
308      Operation Flags      *****
309
310      Blocks Written Channel 1 *****
311      Blocks Written Channel 2 *****
312      Blocks Read Channel 1 *****
313      Blocks Read Channel 2 *****
314      Soft Write Channel 1 *****
315      Soft Write Channel 2 *****
316      Ecc Corrected Channel 1 *****
317      Ecc Corrected Channel 2 *****
318      Read Repositions Channel 1 *****
319      Read Repositions Channel 2 *****
320
321
322
323
324
325      3.2.1 Program Status -
326
327      The program status word is only valid when read with a message
328      number of 1. Upon an error, it will contain the status field of the
329      TMSCP command which produced the error. Note that there are several
330      diagnostic error codes which will be returned with a "ST_DIA"
331      diagnostic status.
332
333      If the test completed without a fatal or hard error, this word
334      will be zero (Normal Successful Completion).

```


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
381
382
383
384
385
386
387
388
389
390
391

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).

392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448

In this case, all information which follows may be ignored (i.e. it's undefined). This is also only valid for message 1.

3.2.9 Error Flags -

This byte will contain the flags from the error log. It is only valid when the error log format code is not equal to 255(10).

3.2.10 Event Code -

This word will contain the event code of the error log. For more detail on this and other fields of the error log, please refer to "TKxx Magnetic Tape Sub-system Functions", 11 January 1985, Ric Perron.

3.2.11 Error Retry Level -

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

3.2.12 Number Of Retries -

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

3.2.13 Position -

This double word will contain the tape position for a Tape Transfer error. Only valid when Error Log Format Code is equal to Tape Transfer error.

3.2.14 Controller Status -

This byte will contain the controller status code and is only valid when Error Log Format Code is equal to Tape Transfer error.

3.2.15 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.

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
495
496
497
498
499
500
501
502
503
504
505

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.

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
552
553
554
555
556
557
558
559
560
561
562

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.

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
609
610
611
612
613
614
615
616
617
618
619

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.

N1

PARAMETER CODING

MACRO V05.05 Wednesday 27-Feb-91 11:55 Page 5-7

SEQ 0013

620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651

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.

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
699
700
701
702
703
704
705
706
707
708
709

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

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
756
757
758
759
760
761
762
763
764
765
766

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 TKxx 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.

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
813
814
815
816
817
818
819
820
821
822
823

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

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
870
871
872
873
874
875
876
877
878
879
880

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

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

883
 884
 885
 886 5.11.6 Test 6 - Overwrite Test -

887
 888 One hundred blocks of data will be written on a single track,
 889 using an "MW" data pattern. (A binary representation of the MW
 890 pattern is 1110111, although the actual number of ones appearing in a
 891 group is subject to further worst case testing by the Drive
 892 Engineering group.) The tape will then be rewound and rewritten. (The
 893 data pattern used for rewrite consists of 50 blocks of worst case MFM
 894 followed by 50 blocks of random data.) The purpose of the test is to
 895 guarantee that the drive is capable of overwriting previously recorded
 896 data.

897
 898 5.11.7 Test 7 - Track Access Test -

899
 900 Fifty blocks of data will be written on a single track. The
 901 tape will then be rewound, the head stepped to the next track and the
 902 same number of blocks will be written. This process will be repeated
 903 until all the tracks for that head have been written. The test will
 904 then reread the blocks from all data. This test is intended to verify
 905 tape tracking and the ability to successfully record and retrieve data
 906 on adjacent tracks.

907
 908
 909
 910 5.11.8 Test 8 - Positioning Test -

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

920
 921
 922
 923 5.12 Retry Algorithms

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

929
 930
 931
 942 .TITLE PROGRAM HEADER AND TABLES
 943 .SBTTL PROGRAM HEADER

969
 971 000000 .ENABL ABS,AMA
 972 .DSABL GBL
 973 000052 . = 52

```

974 000052 010000
975          002000
976
977
978
979 002000
980
981
982
983
984
985
986 002000
987
995
996 002000
002000
002000 103
002001 132
002002 124
002003 113
002004 101
002005 000
002006 000
002007 000
002010
002010 106
002011
002011 060
002012
002012 000000
002014
002014 000170
002016
002016 050064
002020
002020 000000
002022
002022 002202
002024
002024 000000
002026
002026 062130
002030
002030 000000
002032
002032 000000
002034
002034 000000
002036
002036 000000
002040
002040 002124
002042
002042 000000
002044
002044 000000
002046

```

```

.WORD 10000
.= 2000
.NLIST BEX

BGNMOD

; **
; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
; --

POINTER BGNDU,ERRTB!,BGNRPT

HEADER CZTKA,F,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 /F/
L$DEPO:: ;0
.ASCII /O/
L$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
L$APT:: ;APT EXPANSION
.WORD 0
L$DTP:: ;PTR. TO DISPATCH TABLE
.WORD L$DISPATCH
L$PRIO:: ;DIAGNOSTIC RUN PRIORITY
.WORD PRI00
L$ENVI:: ;FLAGS DESCRIBE HOW IT WAS SETUP
.WORD 0
L$EXP1:: ;EXPANSION WORD

```

PROGRAM HEADER

002046	000000			
002050		L\$MREV::	.WORD 0	;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	023414		.WORD L\$DVTYP	
002062		L\$REPP::		;PTR. TO REPORT CODE
002062	035042		.WORD L\$RPT	
002064		L\$EXP4::		
002064	000000		.WORD 0	
002066		L\$EXP5::		
002066	000000		.WORD 0	
002070		L\$AUT::		;PTP. TO ADD UNIT CODE
002070	000000		.WORD 0	
002072		L\$DUT::		;PTR. TO DROP UNIT CODE
002072	041302		.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 ERRRTBL
002102	023054		.WORD L\$ERRRTBL	
002104		L\$ICP::		;PTR. TO INIT CODE
002104	040772		.WORD L\$INIT	
002106		L\$CCP::		;PTR. TO CLEAN-UP CODE
002106	041252		.WORD L\$CLEAN	
002110		L\$ACP::		;PTR. TO AUTO CODE
002110	041324		.WORD L\$AUTO	
002112		L\$PRT::		;PTR. TO PROTECT TABLE
002112	023046		.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	

997

DISPATCH TABLE

1004
1005
1006
1007
1008
1009
1010
1011

002122
002122 000012
002124
002124 041326
002126 042024
002130 042606
002132 043274
002134 044070
002136 044536
002140 045356
002142 046176
002144 047132
002146 047356

.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.

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

1012
1019

1020 002150
002150
002150

103 132 124

DESCRIPT <CZTKAFO TKxx FUNCTIONAL >
L\$DESC::
.ASCIZ /CZTKAFO TKxx FUNCTIONAL/
.EVEN

1021

J2

DEFAULT HARDWARE P-TABLE

```

1023
1024
1025
1026
1027
1028
1029
1030
1031 002200
      002200 000003
      002202
      002202
1032
1038 002202 174500
1039 002204 000260
1040 002206 000000
1041 002210
      002210

```

.SBTTL DEFAULT HARDWARE P-TABLE

```

;***
; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
; THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES.
;---

```

```

      BGNHW  DFPTBL
      .WORD  L10000-L$HW/2
L$HW::
DFPTBL::
      .WORD  174500      ;TKIP BASE ADDRESS
      .WORD  260        ;VECTOR
      .WORD  0          ;T/MSCP UNIT NUMBER
      ENDPW
L10000:

```

K2

SOFTWARE P-TABLE

```
1044 .SBTTL SOFTWARE P-TABLE
1045
1046 ;**
1047 ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
1048 ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
1049 ;--
1050
1051 002210          BGNSW  SFPTBL
          002210 000000 .WORD L10001-L$SW/2
          002212
          002212 L$SW::
          SFPTBL::
1052
1059
1060 002212          ENDSW
          002212 L10001:
1061
1062 002212          ENDMOD
```


L2

1065
1076
1077
1105
1106
1107 002212
1108
1109
1110
1111
1112
1113
1114 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

;

M2

GLOBAL EQUATES SECTION

```
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
```

N2

GLOBAL EQUATES SECTION

```

1119 ;:*****
1120 ;:*****
1121 ;
1122 ;LUN_BLOCK OFFSETS
1123 ;   THESE LITERALS ARE USED AS WORD OFFSETS INTO THE LUNBLK, WHICH
1124 ;   IS POINTED TO THROUGHOUT THE PROGRAM BY R4.
1125 ;
1126 ;:*****
1127 ;:*****
1128
1129 000000 TKIP == 0 ;TKIP REGISTER ADDRESS
1130 000002 TKSA == 2 ;TKSA REGISTER ADDRESS
1131 000004 TKVEC == 4 ;TK INTERRUPT VECTOR
1132 000006 MSCPUN == 6 ;T/MSCP UNIT NUMBER
1133 000010 TKIPSV == 10 ;SAVE LOCATION FOR IP CONTENTS
1134 000012 TKSASV == 12 ;SAVE LOCATION FOR SA CONTENTS
1135 000014 LUNFLG == 14 ;BIT-SPECIFIC MEANINGS AS DEFINED BELOW
1136
1137
1138 ;:*****
1139 ;:*****
1140 ;
1141 ;LUNFLG
1142 ;   THIS WORD IN LUNBLK IS USED TO CONVEY VARIOUS INFORMATION
1143 ;   IN A BIT-SPECIFIC MANNER. BITS USED BY THE PROGRAM ARE
1144 ;   DEFINED AS FOLLOWS.
1145 ;
1146 ;:*****
1147 ;:*****
1148
1149 000001 DRPFLG == BIT0 ;=0 WUT AVAILABLE FOR TEST
1150 ;=1 WUT HAS BEEN DROPPED
1151 000002 INTFLG == BIT1 ;=1 EXPECTED INTERRUPT OCCURRED
1152
1153 000004 BRFLAG == BIT2 ;=1 INTERRUPT PRIORITY TEST
1154
1155 000010 DUPFLG == BIT3 ;=1 DUP COMMAND
1156
1157 000020 ABTFLG == BIT4 ;=1 ABORT LOCAL PROGRAM
1158
1159 000040 CNTRLC == BIT5 ;=1 RETURN TO DRS
1160
1161 000100 T9FLAG == BIT6 ;=1 IN TEST 9
1162

```

GLOBAL EQUATES SECTION

```

1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202

```

```

;*****
;*****
; UQ-PORT EQUATES
; THIS SECTION DEFINES THOSE LITERALS USED
; BY THE DIAGNOSTIC IN THE UQ-PORT PROTOCOL.
; IN GENERAL THEY HAVE BEEN FORMED BY USING
; THE TWO LETTER MNEMONIC DEFINED IN UQSSP,
; PRECEDED BY "B." INDICATING THEY ARE BITS.
;*****
;*****
;READ-ONLY BITS
B.S1 == BIT11 ;STEP 1
B.S2 == BIT12 ;STEP 2
B.S3 == BIT13 ;STEP 3
B.S4 == BIT14 ;STEP 4
B.ER == BIT15 ;ERROR INDICATION
B.NV == BIT10 ;=0 VECTOR IS HOST SETTABLE
B.QB == BIT9 ;=1 SUPPORTS 22 BIT HOST BUS
B.DI == BIT8 ;=1 SUPPORTS ENHANCED DIAGNOSTICS
B.OD == BIT7 ;=1 SUPPORTS ODD BUFFER ADDRESSES
B.MP == BIT6 ;=1 SUPPORTS ADDRESS MAPPING
;WRITE-ONLY BITS
B.PP == BIT15 ;PERFORM PURGE AND POLL TESTS
B.WR == BIT14 ;ENTER DIAGNOSTIC WRAP MODE
B.LF == BIT1 ;LAST FAIL REQUEST
B.PI == BIT0 ;ENABLE ADAPTER PURGE INTERRUPTS
B.GO == BIT0 ;GO BIT - START RUNNING
;READ/WRITE BITS
B.IE == BIT7 ;STEP X-TION INTERRUPT ENABLE

```

C3

GLOBAL EQUATES SECTION

```

1204      ;*****
1205      ;*****
1206      ;
1207      ;GENERAL PURPOSE EQUATES
1208      ;
1209      ;*****
1210      ;*****

```

```

1211
1212      000004      VEC4      ==      4      ;VECTOR FOUR - NXM TIMEOUTS, ETC.
1213      177560      RCSR      ==      177560 ;TERMINAL RECEIVE CONTROL/STATUS REGISTER ADDRESS
1214      177562      RBUF      ==      177562 ;TERMINAL RECEIVE BUFFER ADDRESS
1215

```

GLOBAL EQUATES SECTION

```

1217      ;:*****
1218      ;:*****
1219      ;
1220      ;MEMORY MANAGEMENT EQUATES
1221      ;
1222      ;:*****
1223      ;:*****
1224
1225      177572      MMUSRO ==      177572      ;STATUS REG 0
1226      177574      MMUSR1 ==      177574
1227      177576      MMUSR2 ==      177576
1228      172516      MMUSR3 ==      172516      ;SHOULD ONLY BE PRESENT ON 22 BIT CPU'S
1229
1230      172340      KPAR0  ==      172340      ;KERNEL MODE PAGE ADDRESS REG 0
1231      172342      KPAR1  ==      172342
1232      172344      KPAR2  ==      172344
1233      172346      KPAR3  ==      172346
1234      172350      KPAR4  ==      172350
1235      172352      KPAR5  ==      172352
1236      172354      KPAR6  ==      172354
1237      172356      KPAR7  ==      172356      ;ALWAYS FOR I/O PAGE
1238
1239      172300      KPDR0  ==      172300      ;KERNEL MODE PAGE DESCRIPTOR REG 0
1240      172302      KPDR1  ==      172302
1241      172304      KPDR2  ==      172304
1242      172306      KPDR3  ==      172306
1243      172310      KPDR4  ==      172310
1244      172312      KPDR5  ==      172312
1245      172314      KPDR6  ==      172314
1246      172316      KPDR7  ==      172316
1247
1248      000001      MMON   ==      BIT0      ;ENABLE MMU - MMUSRO
1249      000020      MM20N  ==      BIT4      ;ENABLE 22 BIT MMU - MMUSR3
1250

```

GLOBAL EQUATES SECTION

```

1252      ;:*****
1253      ;:*****
1254      ;
1255      ;COMMAND PACKET OPCODES
1256      ;
1257      ;:*****
1258      ;:*****
1259
1260      000004      OP.SCC ==      04      ;SET CONTROLLER CHARACTERISTICS OPCODE
1261      000011      OP.ONL ==      11      ;ONLINE OPCODE
1262      000001      OP.GDS ==      01      ;GET DUST STATUS OPCODE
1263      000003      OP.ELP ==      03      ;EXECUTE LOCAL PROGRAM OPCODE
1264      000005      OP.REC ==      05      ;RECEIVE DATA OPCODE
1265      000006      OP.ABT ==      06      ;ABORT PROGRAM OPCODE
1266      000200      OP.END ==      200     ;END MESSAGE FLAG OPCODE
1267
1268
1269      ;:*****
1270      ;:*****
1271      ;
1272      ;GENERIC COMMAND PACKET OFFSETS
1273      ;
1274      ;:*****
1275      ;:*****
1276
1277      000000      P.CRF ==      0      ;COMMAND REFERENCE NUMBER
1278      000004      P.UNIT ==      4      ;UNIT NUMBER
1279      000010      P.OPCD ==      10     ;OPCODE
1280      000011      P.FLGS ==      11     ;END MESSAGE FLAGS
1281      000012      P.MOD ==      12     ;MODIFIERS / STATUS
1282      000012      P.STS ==      12     ;MODIFIERS / STATUS
1283      000014      P.BCNT ==      14     ;BYTE COUNT
1284      000020      P.BUFF ==      20     ;BUFFER DESCRIPTOR
1285
1286
1287      ;:*****
1288      ;:*****
1289      ;
1290      ;GET DUST STATUS END PACKET OFFSETS
1291      ;
1292      ;:*****
1293      ;:*****
1294
1295      000014      P.EXT1 ==      14     ;1ST BYTE OF PROGRAM EXTENSION
1296      000015      P.EXT2 ==      15     ;2ND BYTE OF PROGRAM EXTENSION
1297      000016      P.EXT3 ==      16     ;3RD BYTE OF PROGRAM EXTENSION
1298      000017      DUSTFL ==      17     ;FLAGS
1299      000020      P.IND1 ==      20     ;1ST WORD OF PROGRESS INDICATOR
1300      000022      P.IND2 ==      22     ;2ND WORD OF PROGRESS INDICATOR
1301      000024      TIMOUT ==      24     ;TIMEOUT VALUE
1302

```

GLOBAL EQUATES SECTION

```

1304      ;*****
1305      ;*****
1306      ;
1307      ;TKSA BIT DEFINITIONS
1308      ;
1309      ;*****
1310      ;*****
1311
1312      100000      ERR      ==      100000      ;ERROR
1313      004000      S1      ==      004000      ;STEP 1
1314      000001      G0      ==      000001      ;GC
1315
1316
1317      ;*****
1318      ;*****
1319      ;
1320      ;U/Q PORT LITERALS
1321      ;
1322      ;*****
1323      ;*****
1324
1325      100000      OWN      ==      100000      ;DESCRIPTOR OWNERSHIP BIT
1326      040000      FLAG    ==      040000      ;DESCRIPTOR INTERRUPT FLAG BIT
1327      000200      IMM     ==      000200      ;IMMEDIATE COMMAND FLAG
1328      000010      TF.BLK  ==      10          ;TAPE FORMAT
1329      000000      HSTIMO  ==      0          ;HOST TIMEOUT VALUE
1330      000000      MSCPVR  ==      0          ;MSCP VERSION NUMBER
1331      000004      RNGSTP  ==      4          ;DESCRIPTOR RING STEP
1332      000104      RSPSTP  ==      68         ;RESPONCE BUFFER STEP
1333
1334

```


G3

GLOBAL EQUATES SECTION

```

1336      ;:*****
1337      ;:*****
1338      ;
1339      ;TMSCP DRIVER BUFFER OFFSETS
1340      ;
1341      ;:*****
1342      ;:*****
1343
1344      000002      HIADDR ==      2.      ;DESCRIPTOR ADDRESS OFFSET
1345      177777      CONID  ==      -1.     ;COMMAND/RESPONSE CONNECTION TYPE I.D.
1346      177776      CRD    ==      2.      ;COMMAND/RESPONSE CREDIT LIMIT OFFSET
1347      177774      MSGLEN ==      4.      ;COMMAND/RESPONSE MESSAGE LENGTH
1348      000005      TXFER  ==      5.      ;ERROR FORMAT FOR "TAPE TRANSFER" ERROR LOG
1349      000011      DRIVER ==      9.      ;ERROR FORMAT FOR "DRIVE ERROR" ERROR LOG
1350      000000      CNTER  ==      0.      ;ERROR FORMAT FOR "CONTROLLER ERROR" ERROR LOG
1351

```

GLOBAL EQUATES SECTION

```

1353      ;:*****
1354      ;:*****
1355      ;
1356      ;OFFSET VALUES INTO LEVEL 2 MICRODIAGNOSTIC RECEIVE DATA BUFFER
1357      ;
1358      ;:*****
1359      ;:*****
1360
1361      000002      L2STA  ==      2      ;OFFSET OF PROGRAM STATUS
1362      000004      L2DRV  ==      4      ;OFFSET OF DRIVE STATUS
1363      000006      L2TST  ==      6      ;OFFSET OF TEST NUMBER
1364      000007      L2TRK  ==      7      ;OFFSET OF TRACK NUMBER
1365      000010      L2CMD  ==     10      ;OFFSET OF COMMAND IN ERROR
1366      000011      L2RSP  ==     11      ;OFFSET OF RESPONSE FLAG
1367      000012      L2UNT  ==     12      ;OFFSET OF UNIT FLAGS
1368      000014      L2BWR1 ==     14      ;OFFSET OF BLOCKS WRITTEN CH. 1
1369      000020      L2BWR2 ==     20      ;OFFSET OF BLOCKS WRITTEN CH. 2
1370      000024      L2BRD1 ==     24      ;OFFSET OF BLOCKS READ CH. 1
1371      000030      L2BRD2 ==     30      ;OFFSET OF BLOCKS READ CH. 2
1372      000034      L2SWR1 ==     34      ;OFFSET OF SOFT WRITE ERRORS CH. 1
1373      000036      L2SWR2 ==     36      ;OFFSET OF SOFT WRITE ERRORS CH. 2
1374      000040      L2ECC1 ==     40      ;OFFSET OF ECC CORRECTED ERRORS CH. 1
1375      000042      L2ECC2 ==     42      ;OFFSET OF ECC CORRECTED ERRORS CH. 2
1376      000044      L2CRC1 ==     44      ;OFFSET OF CRC ERRORS ON ECC BLOCK CH. 1
1377      000046      L2CRC2 ==     46      ;OFFSET OF CRC ERRORS ON ECC BLOCK CH. 2
1378      000050      L2REP1 ==     50      ;OFFSET OF READ REPOSITIONS CH. 1
1379      000052      L2REP2 ==     52      ;OFFSET OF READ REPOSITIONS CH. 2
1380      000134      L2ELFM ==    134      ;OFFSET OF ERROR LOG ERROR FORMAT
1381      000135      L2ELFL ==    135      ;OFFSET OF ERROR LOG FLAGS
1382      000136      L2ELEV ==    136      ;OFFSET OF ERROR LOG EVENT CODE
1383      000144      L2ELRL ==    144      ;OFFSET OF ERROR LOG RETRY LEVEL
1384      000145      L2ELRT ==    145      ;OFFSET OF ERROR LOG NUMBER OF RETRIES
1385      000146      L2ELP1 ==    146      ;OFFSET OF ERROR LOG POSITION (low order)
1386      000150      L2ELP2 ==    150      ;OFFSET OF ERROR LOG POSITION (high order)
1387      000154      L2ELST ==    154      ;OFFSET OF CONTROLLER STATUS
1388      000155      L2ELDE ==    155      ;OFFSET OF ERROR LOG DRIVE ERROR CODE
1389      000156      L2ELDF ==    156      ;OFFSET OF ERROR LOG DRIVE FLAGS
1390      000157      L2ELTN ==    157      ;OFFSET OF ERROR LOG TRACK NUMBER
1391      000160      L2ELPB ==    160      ;OFFSET OF ERROR LOG PHYSICAL BLOCK NUMBER
1392      000162      L2ELLB ==    162      ;OFFSET OF ERROR LOG LOGICAL BLOCK NUMBER
1393      000163      L2ELT0 ==    163      ;OFFSET OF ERROR LOG LSB OF TAPE LENGTH COUNT
1394      000164      L2ELT1 ==    164      ;OFFSET OF ERROR LOG BYTE 1 OF TAPE LENGTH COUNT
1395      000165      L2ELT2 ==    165      ;OFFSET OF ERROR LOG MSB OF TAPE LENGTH COUNT
1396      000166      L2ELDS ==    166      ;OFFSET OF ERROR LOG DRIVE STATE
1397      000170      L2ELRW ==    170      ;OFFSET OF ERROR LOG READ/WRITE STATE
1398      000172      L2ELOF ==    172      ;OFFSET OF ERROR LOG OPERATION FLAGS
1399      000140      L2ELEC ==    140      ;OFFSET OF ERROR LOG CONTROLLER ERROR CODE

```

GLOBAL DATA SECTION

```

1401      .SBTTL GLOBAL DATA SECTION
1402
1403
1404      ;*****
1405      ;*****
1406      ;
1407      ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
1408      ; IN MORE THAN ONE TEST.
1409      ;
1410      ;*****
1411      ;*****
1412
1413
1414
1415      ;*****
1416      ;*****
1417      ;
1418      ;LUNBLK
1419      ; THIS BLOCK OF MEMORY IS USED TO STORE VARIABLE INFORMATION
1420      ; PERTAINING TO THE CURRENT LOGICAL UNIT UNDER TEST. LUNBLK
1421      ; IS POINTED TO THROUGHOUT THE PROGRAM BY R4 AND INDIVIDUAL
1422      ; LOCATIONS ARE ACCESSED VIA LITERALS DEFINED ABOVE.
1423      ;
1424      ;*****
1425      ;*****
1426
1427 002212 000000 000000 000000 LUNBLK::      .WORD  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1428
1429
1430      ;*****
1431      ;*****
1432      ;
1433      ;UQ-PORT NECESSITIES
1434      ; THESE TABLES ARE SET UP BY VARIOUS
1435      ; TESTS WITH VALUES TO BE WRITTEN TO
1436      ; THE PORT, AND COMPARISON VALUES TO
1437      ; CHECK THE PORT AFTER EACH STEP TRAN-
1438      ; SITION OCCURS, RESPECTIVELY.
1439      ;
1440      ;*****
1441      ;*****
1442
1443 002250      STPTBL::      .BLKW  4      ;VALUES WRITTEN TO THE PORT
1444
1445 002260      CMPTBL::      .BLKW  4      ;COMPARISON VALUES
1446

```

GLOBAL DATA SECTION

```

1448 ;*****
1449 ;*****
1450 ;
1451 ;PROGRAM CONTROL VARIABLES
1452 ; THESE GLOBAL VARIABLES ARE GENERALLY USED TO CONTROL THE
1453 ; OVERALL EXECUTION OF THE DIAGNOSTIC.
1454 ;
1455 ;*****
1456 ;*****
1457
1458 002270 000000 PASCNT:: .WORD 0 ;CUMULATIVE PROGRAM PASS COUNTER
1459 002272 000000 ITRCNT:: .WORD 0 ;LOADED BY EACH TEST TO SPECIFY # OF ITERATIONS
1460 002274 000000 KTFLAG:: .WORD 0 ;=0 MEMORY MANAGEMENT NOT AVAILABLE
1461 ;=1 MEMORY MANAGEMENT IS AVAILABLE
1462 002276 000000 TRP4FG:: .WORD 0 ;=1 TRAP TO VECTOR OCCURRED
1463 002300 000000 PAROFF:: .WORD 0 ;USED IN TEST 7 TO STEP THROUGH UPPER MEMORY
1464 002302 000000 CMMERR:: .WORD 0 ;=0 NO ERROR IN COMMUNICATION AREA
1465 ;=1 ERROR WITHIN COMMUNICATION AREA
1466 ;=-1 ERROR BEYOND BOUNDS OF COMM AREA
1467 002304 000000 CMTBLG:: .WORD 0 ;# OF CONTIGUOUS WORDS IN ERROR IN COMM AREA
1468 002306 000000 CMARLG:: .WORD 0 ;LENGTH OF COMM AREA FOR TEST N
1469 002310 000000 FRUIS:: .WORD 0 ;POINTER TO FAULTY FRU ASCII FOR PRINTOUT
1470 002312 000000 LOGUNT:: .WORD 0 ;LOGICAL UNIT # OF CURRENT UUT
1471 002314 000000 SAEXP:: .WORD 0 ;LOADED WITH EXPECTED SA FOR ERROR CHECKING
1472 002316 000000 INISTP:: .WORD 0 ;CURRENT STEP OF INIT SEQUENCE
1473 002320 000000 STEPST:: .WORD 0 ;SUCCESS/FAIL STATUS FROM STEP SUBROUTINES
1474 002322 000000 WRDATA:: .WORD 0 ;LOADED WITH DATA FRO WRAP MODE TEST
1475 002324 000000 INNER:: .WORD 0 ;COUNTER FOR PDELAY ROUTINE
1476 002326 000000 OUTER:: .WORD 0 ;OTHER COUNTER FOR PDELAY
1477 002330 000000 TOUT:: .WORD 0 ;TIMEOUT INDICATOR FOR PDELAY
1478 002332 000000 TEMP:: .WORD 0 ;TEMPORARY STORAGE LOCATION
1479 002334 000000 TEMP1:: .WORD 0 ;TEMPORARY STORAGE LOCATION
1480 002336 000000 FLAGS:: .WORD 0 ;WORK LOCATION FOR SUPERVISOR FLAGS

```

GLOBAL DATA SECTION

```

1482 ;:*****
1483 ;:*****
1484 ;
1485 ;TMSCP/DUP COMMAND PACKETS
1486 ;
1487 ;:*****
1488 ;:*****
1489 ;
1490 ;:*****
1491 ;
1492 ;TMSCP SET CONTROLLER CHARACTERISTICS COMMAND PACKET
1493 ;
1494 ;:*****
1495 ;
1496 002340 000040 000000
1497 002344 000000 000000
1498 002350 000000 000000
1499 002354 000004 000000
1500 002360 000000 000000
1501 002364 000000 000000
1502 002370 000000 000000
1503 002374 000000 000000
1504 002400 000000 000000
1505 ;
1506 ;
1507 ;STARS
1508 ;
1509 ;TMSCP ONLINE COMMAND PACKET
1510 ;
1511 ;:*****
1512 ;
1513 002404 000044 000000
1514 002410 000000 000000
1515 002414 000000 000000
1516 002420 000011 000000
1517 002424 000000 000000
1518 002430 000000 000000
1519 002434 000000 000000
1520 002440 000000 000000
1521 002444 000000 000000
1522 002450 000000 000000
1523 ;
1524 ;
1525 ;STARS
1526 ;
1527 ;DUP GET DUST STATUS COMMAND PACKET
1528 ;
1529 ;:*****
1530 ;
1531 002454 000020 000000
1532 002460 000000 000000
1533 002464 000000 000000
1534 002470 000001 000000
1535 ;
1536 ;
1537 ;STARS

```

```

;:*****
;:*****
;TMSCP/DUP COMMAND PACKETS
;:*****
;:*****
;:*****
;TMSCP SET CONTROLLER CHARACTERISTICS COMMAND PACKET
;:*****
SCTRLC: .WORD 32.,0
         .WORD 0,0
         .WORD 0,0
         .WORD OP.SCC,0
         .WORD 0,0
         .WORD 0,0
         .WORD 0,0
         .WORD 0,0
         .WORD 0,0
;STARS
;TMSCP ONLINE COMMAND PACKET
;:*****
ONLINE: .WORD 36.,0
        .WORD 0,0
        .WORD 0,0
        .WORD OP.ONL,0
        .WORD 0,0
        .WORD 0,0
        .WORD 0,0
        .WORD 0,0
        .WORD 0,0
        .WORD 0,0
;STARS
;DUP GET DUST STATUS COMMAND PACKET
;:*****
GDUST: .WORD 16.,0
       .WORD 0,0
       .WORD 0,0
       .WORD OP.GDS,0
;STARS

```

L3

GLOBAL DATA SECTION

```

1539 ;DUP EXECUTE LOCAL PROGRAM COMMAND PACKET
1540 ;
1541 ;:*****
1542 ;
1543 002474 000022 000000 XLOCPR: .WORD 18.,0
1544 002500 000000 000000 .WORD 0,0
1545 002504 000000 000000 .WORD 0,0
1546 002510 000003 000000 .WORD OP.ELP,0
1547 002514 104 111 101 .ASCII /DIAGL2/
1548 ;
1549 ;
1550 ;STARS
1551 ;
1552 ;DUP RECEIVE DATA COMMAND PACKET
1553 ;
1554 ;:*****
1555 ;
1556 002522 000024 000000 RCVDAT: .WORD 20.,0
1557 002526 000000 000000 .WORD 0,0
1558 002532 000000 000000 .WORD 0,0
1559 002536 000005 000000 .WORD OP.REC,0
1560 002542 000226 000000 .WORD 150.,0
1561 002546 060000 000000 .WORD RDBUF,0
1562 ;
1563 ;
1564 ;STARS
1565 ;
1566 ;DUP ABORT COMMAND PACKET
1567 ;
1568 ;:*****
1569 ;
1570 002552 000014 000000 ABORT: .WORD 12.,0
1571 002556 000000 000000 .WORD 0,0
1572 002562 000000 000000 .WORD 0,0
1573 002566 000006 000000 .WORD OP.ABT,0
1574 ;

```

GLOBAL DATA SECTION

```

1576      ;:*****
1577      ;:*****
1578      ;
1579      ;CLASS DRIVER BUFFERS
1580      ;
1581      ;:*****
1582      ;:*****
1583      ;
1584 002572 RESPBF::      .BLKW  2.      ;TOP 4 LOCATIONS OF RESPONSE BUFFER
1585 002576 RSPBUF::      .BLKW  66.      ;DRIVER RESPONSE BUFFER
1586      ;
1587      ;
1588      ;:*****
1589      ;:*****
1590      ;
1591      ;U/Q PORT DESCRIPTOR RINGS
1592      ;
1593      ;:*****
1594      ;:*****
1595      ;
1596 003002 DSCRNG::      .BLKW  2.      ;DESCRIPTOR RING
1597 003006 RSPEND::      ;END OF RESPONSE BUFFER
1598 003006 RSPRNG::      .BLKW  4.      ;RESPONSE DESCRIPTOR RING
1599 003016 CMDRNG::      .BLKW  4.      ;COMMAND DESCRIPTOR RING
1600 003026 DSCEND::      ;END OF DESCRIPTOR RING
1601      ;
1602      ;
1603      ;:*****
1604      ;:*****
1605      ;
1606      ;CLASS AND PORT DRIVER VARIABLES
1607      ;
1608      ;:*****
1609      ;:*****
1610      ;
1611 003026 000000 CNTHI::      .WORD  0      ;VALUE OF THE HIGH TIMEOUT
1612 003030 000000 CNTFLG::      .WORD  0      ;CONTROLLER FLAGS
1613 003032 000000 PCKSIZ::      .WORD  0      ;PACKET SIZE IN BYTES
1614 003034 000000 CMDREF::      .WORD  0      ;COMMAND REFERENCE NUMBER
1615 003036 000000 CMDCNT::      .WORD  0      ;COMMAND COUNT
1616 003040 WRBUF::      .BLKW  4096.      ;WRITE BUFFER
1617 023040 000000 CMDSAV::      .WORD  0      ;COMMAND DESCRIPTOR SAVE
1618 023042 000000 RSPSAV::      .WORD  0      ;RESPONSE DESCRIPTOR SAVE
1619 023044 000000 CURCMD::      .WORD  0      ;POINTER TO CURRENT COMMAND ASCII
1620

```

N3

GLOBAL DATA SECTION

```

1622      ;*****
1623      ;*****
1624      ;
1625      ;PROTECTION TABLE
1626      ;
1627      ;*****
1628      ;*****
1632
1633 023046 BGNPROT
      023046 L$PROT::
1634 023046 000000      .WORD 0
1635 023050 177777      .WORD -1
1636 023052 177777      .WORD -1
1637
1638 023054 ENDPROT
1639

```


B4

```

1641 023054 STARS
1642 023054 STARS
1643
1644 ;LEVEL 2 MICRODIAGNOSTIC ERROR TABLE
1645
1646 023054 STARS
1647 023054 STARS
1648
1649 023054

```

```

ERRTBL
L$ERRTBL::
ERRTYP:: .WORD 0
ERRNBR:: .WORD 0
ERRMSG:: .WORD 0
ERRBLK:: .WORD 0

```

```

1650
1651 023064 L2ETBL:: ;START OF TMSCP COMMAND ERROR MESSAGES
1652
1653 023064 000001 .WORD 1. ;INVALID COMMAND
1654 023066 000145 .WORD 101.
1655 023070 030771 .WORD L2ER1
1656 023072 033604 .WORD L2DUMP
1657
1658 023074 000001 .WORD 1. ;COMMAND ABORTED
1659 023076 000146 .WORD 102.
1660 023100 031011 .WORD L2ER2
1661 023102 033604 .WORD L2DUMP
1662
1663 023104 000001 .WORD 1. ;UNIT-OFFLINE
1664 023106 000147 .WORD 103.
1665 023110 031031 .WORD L2ER3
1666 023112 033604 .WORD L2DUMP
1667
1668 023114 000001 .WORD 1. ;UNIT-AVAILABLE
1669 023116 000150 .WORD 104.
1670 023120 031046 .WORD L2ER4
1671 023122 033604 .WORD L2DUMP
1672
1673 023124 000001 .WORD 1. ;INVALID STATUS
1674 023126 000151 .WORD 105.
1675 023130 031065 .WORD L2ER5
1676 023132 033604 .WORD L2DUMP
1677
1678 023134 000001 .WORD 1. ;WRITE PROTECTED
1679 023136 000152 .WORD 106.
1680 023140 031115 .WORD L2ER6
1681 023142 033604 .WORD L2DUMP
1682
1683 023144 000001 .WORD 1. ;COMPARE ERROR
1684 023146 000153 .WORD 107.
1685 023150 031135 .WORD L2ER7
1686 023152 033604 .WORD L2DUMP
1687
1688 023154 000001 .WORD 1. ;DATA ERROR

```

C4

1689	023156	000154	.WORD	108.	
1690	023160	031153	.WORD	L2ER8	
1691	023162	033604	.WORD	L2DUMP	
1692					
1693	023164	000001	.WORD	1.	;HOST BUFFER ACCESS ERROR
1694	023166	000155	.WORD	109.	
1695	023170	031166	.WORD	L2ER9	
1696	023172	033604	.WORD	L2DUMP	
1697					
1698	023174	000001	.WORD	1.	;CONTROLLER ERROR
1699	023176	000156	.WORD	110.	
1700	023200	031217	.WORD	L2ER10	
1701	023202	033604	.WORD	L2DUMP	
1702					
1703	023204	000001	.WORD	1.	;DRIVE ERROR
1704	023206	000157	.WORD	111.	
1705	023210	031240	.WORD	L2ER11	
1706	023212	033604	.WORD	L2DUMP	
1707					
1708	023214	000001	.WORD	1.	;FORMATTER ERROR
1709	023216	000160	.WORD	112.	
1710	023220	031254	.WORD	L2ER12	
1711	023222	033604	.WORD	L2DUMP	
1712					
1713	023224	000001	.WORD	1.	;BOT ENCOUNTERED
1714	023226	000161	.WORD	113.	
1715	023230	031274	.WORD	L2ER13	
1716	023232	033604	.WORD	L2DUMP	
1717					
1718	023234	000001	.WORD	1.	;TAPE MARK ENCOUNTERED
1719	023236	000162	.WORD	114.	
1720	023240	031314	.WORD	L2ER14	
1721	023242	033604	.WORD	L2DUMP	
1722					
1723	023244	000001	.WORD	1.	;INVALID STATUS
1724	023246	000163	.WORD	115.	
1725	023250	031342	.WORD	L2ER15	
1726	023252	033604	.WORD	L2DUMP	
1727					
1728	023254	000001	.WORD	1.	;RECORD DATA TRUNCATED
1729	023256	000164	.WORD	116.	
1730	023260	031372	.WORD	L2ER16	
1731	023262	033604	.WORD	L2DUMP	
1732					
1733	023264	000001	.WORD	1.	;POSITION LOST
1734	023266	000165	.WORD	117.	
1735	023270	031420	.WORD	L2ER17	
1736	023272	033604	.WORD	L2DUMP	
1737					
1738	023274	000001	.WORD	1.	;SERIOUS EXCEPTION
1739	023276	000166	.WORD	118.	
1740	023300	031436	.WORD	L2ER18	
1741	023302	033604	.WORD	L2DUMP	
1742					
1743	023304	000001	.WORD	1.	;LEOT DETECTED
1744	023306	000167	.WORD	119.	
1745	023310	031457	.WORD	L2ER19	

D4

GLOBAL DATA SECTION

1746	023312	033604	.WORD	L2DUMP	
1747					
1748	023314		L2MSG:		;START OF MICRODIAGNOSTIC DFTECTED ERROR MESSAGES
1749					
1750	023314	000001	DCERR:	.WORD 1.	DATA COMPARE ERROR
1751	023316	00017C		.WORD 120.	
1752	023320	031475		.WORD L2ER20	
1753	023322	033604		.WORD L2DUMP	
1754					
1755	023324	000001	CNTERR:	.WORD 1.	;CONTROLLER ERROR
1756	023326	000171		.WORD 121.	
1757	023330	031557		.WORD L2ER21	
1758	023332	033604		.WORD L2DUMP	
1759					
1760	023334	000001	INVSTA:	.WORD 1.	;INVALID STATUS
1761	023336	000172		.WORD 122.	
1762	023340	031637		.WORD L2ER22	
1763	023342	033604		.WORD L2DUMP	
1764					
1765	023344	000001	BPERR:	.WORD 1.	;BAD PATTERN NUMBER
1766	023346	000173		.WORD 123.	
1767	023350	031726		.WORD L2ER23	
1768	023352	033604		.WORD L2DUMP	
1769					
1770	023354	000001	RSPADD:	.WORD 1.	;RESPONSE ADDRESS ERROR
1771	023356	000174		.WORD 124.	
1772	023360	032016		.WORD L2ER24	
1773	023362	033604		.WORD L2DUMP	
1774					
1775	023364	000001	HBFADD:	.WORD 1.	;HOST BUFFER ADDRESS ERROR
1776	023366	000175		.WORD 125.	
1777	023370	032104		.WORD L2ER25	
1778	023372	033604		.WORD L2DUMP	
1779					
1780	023374	000001	UNERLG:	.WORD 1.	;UNKNOWN ERROR LOG RECEIVED
1781	023376	000176		.WORD 126.	
1782	023400	032175		.WORD L2ER26	
1783	023402	033604		.WORD L2DUMP	
1784					
1785	023404	000001	RSPTO:	.WORD 1.	;RESPONSE TIMEOUT ERROR
1786	023406	000177		.WORD 127.	
1787	023410	032267		.WORD L2ER27	
1788	023412	033604		.WORD L2DUMP	

E4

GLOBAL TEXT SECTION

```

1790 .SBTTL GLOBAL TEXT SECTION
1794 ;*****
1795 ;*****
1796 ;
1797 ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
1798 ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
1799 ; MORE THAN ONE TEST.
1800 ;
1801 ;*****
1802 ;*****
1803 ;
1804 ;*****
1805 ;*****
1806 ;
1807 ; NAMES OF DEVICES SUPPORTED BY PROGRAM
1808 ;
1809 ;*****
1813
1814 023414          DEVTYP <TK70/TK50>
      023414          L$DVTYP:
      023414 124    113    067      .ASCIZ *TK70/TK50*
1815          .EVEN

```

```

1820
1821          ;:*****
1822          ;
1823          ;FORMAT STATEMENTS
1824          ;
1825          ;:*****
1826
1827 023426    045    101    111  LINE1:: .ASCIZ  ?%AINIT SEQUENCE STEP #: %D1?
1828 023462    045    116    045  LINE2:: .ASCIZ  ?%ASA REG: %06%A EXPCTD: %06%A ACTUAL SA: %06?
1829 023542    045    116    045  LINE3:: .ASCIZ  ?%AIP REG ADDRESS: %06?
1830 023572    045    116    045  LINE4:: .ASCIZ  ?%A****FAILING FRU: %T%A****%N?
1831 023634    045    101    122  LINE5:: .ASCIZ  ?%ARELOCATION CONSTANT: %06%A VIRT. ADD: %06?
1832 023711    045    116    045  LINE6:: .ASCIZ  ?%AEXPECTED: %06%A RECEIVED: %06?
1833 023754    045    101    120  LINE7:: .ASCIZ  ?%APHYSICAL ADD: %06?
1834
1835 024000    045    116    045  WR1:: .ASCIZ  ?%ASA REG: %06%A SA CONTENTS: %06?
1836 024043    045    116    045  WR2:: .ASCIZ  ?%AFAILING COMMAND : %T%N?
1837 024077    045    116    045  WR3:: .ASCIZ  ?%A RESPONSE PACKET%N?
1838 024130    045    116    045  WR4:: .ASCIZ  ?%A %06%A %06?
1839 024155    045    116    045  WR5:: .ASCIZ  ?%ACMD EXP: %06%A CMD REC: %06?
1840 024217    045    116    000  WR6:: .ASCIZ  ?%N?
1841 024222    045    116    045  WR7:: .ASCIZ  ?%AFAILING COMMAND : %T%N%AUNIT UNKNOWN?
1842 024273    045    116    045  WR8:: .ASCIZ  ?%AFAILING COMMAND : %T%N%AND MEDIA MOUNTED?
1843 024350    045    116    045  WR9:: .ASCIZ  ?%AFAILING COMMAND : %T%N%AUNIT INOPERATIVE?
1844
1845
1846 024425    045    116    062  L21:: .ASCIZ  ?%N2%A**** LEVEL 2 MICRODIAGNOSTIC DUMP ****?
1847 024501    045    116    062  L22:: .ASCIZ  ?%N2%APROGRAM STATUS: %06?
1848 024537    045    116    045  L23:: .ASCIZ  ?%NADRIE STATUS: %03?
1849 024577    045    116    045  L24:: .ASCIZ  ?%NATEST NUMBER: %03?
1850 024637    045    116    045  L25:: .ASCIZ  ?%NATRACK NUMBER: %03?
1851 024677    045    116    045  L26:: .ASCIZ  ?%NATMSCP COMMAND: %03?
1852 024737    045    116    045  L27:: .ASCIZ  ?%NATMSCP RESPONSE FLAG: %03?
1853 024777    045    116    045  L28:: .ASCIZ  ?%NATMSCP UNIT FLAG: %06?
1854 025034    045    116    062  L29:: .ASCIZ  ?%N2%ABLOCKS WRITTEN CHANNEL 1: %D6?
1855 025101    045    116    045  L210:: .ASCIZ  ?%NABLOCKS WRITTEN CHANNEL 2: %D6?
1856 025145    045    116    045  L211:: .ASCIZ  ?%NABLOCKS READ CHANNEL 1: %D6?
1857 025211    045    116    045  L212:: .ASCIZ  ?%NABLOCKS READ CHANNEL 2: %D6?
1858 025255    045    116    045  L213:: .ASCIZ  ?%NASOFT WRITE CHANNEL 1: %D6?
1859 025321    045    116    045  L214:: .ASCIZ  ?%NASOFT WRITE CHANNEL 2: %D6?
1860 025365    045    116    045  L215:: .ASCIZ  ?%NAECC CORRECTED CHANNEL 1: %D6?
1861 025431    045    116    045  L216:: .ASCIZ  ?%NAECC CORRECTED CHANNEL 2: %D6?
1862 025475    045    116    045  L219:: .ASCIZ  ?%NAREAD REPOSITIONS CHANNEL 1: %D6?
1863 025541    045    116    045  L220:: .ASCIZ  ?%NAREAD REPOSITIONS CHANNEL 2: %D6%N?
1864 025607    045    116    062  L221:: .ASCIZ  ?%N2%AERROR LOG (TAPE TRANSFER ERROR)?
1865 025654    045    116    045  L222:: .ASCIZ  ?%NAERROR FORMAT: %03?
1866 025716    045    116    045  L223:: .ASCIZ  ?%NAERROR FLAGS: %03?
1867 025760    045    116    045  L224:: .ASCIZ  ?%NAEVENT CODE: %06?
1868 026017    045    116    045  L225:: .ASCIZ  ?%NAERROR RETRY LEVEL: %03?
1869 026061    045    116    045  L226:: .ASCIZ  ?%NANUMBER OF RETRIES: %D3?
1870 026123    045    116    045  L227:: .ASCIZ  ?%NAPOSITION (LOW ORDER): %06?
1871 026162    045    116    045  L228:: .ASCIZ  ?%NAPOSITION (HIGH ORDER): %06?
1872 026221    045    116    045  L229:: .ASCIZ  ?%NACONTROLLER STATUS: %03?
1873 026263    045    116    045  L230:: .ASCIZ  ?%NADRIE ERROR CODE: %03?
1874 026325    045    116    045  L231:: .ASCIZ  ?%NADRIE FLAGS: %03?
1875 026367    045    116    045  L232:: .ASCIZ  ?%NATRACK NUMBER: %D3?
1876 026431    045    116    045  L233:: .ASCIZ  ?%NAPHYSICAL BLOCK NUMBER: %D6?

```

1877	026470	045	116	045	L234::	.ASCIZ	?N#ALOGICAL BLOCK NUMBER:	#03?		
1878	026532	045	116	045	L235::	.ASCIZ	?N#ATAPE COUNT 0 :	#03?		
1879	026574	045	116	045	L236::	.ASCIZ	?N#ATAPE COUNT 1 :	#03?		
1880	026636	045	116	045	L237::	.ASCIZ	?N#ATAPE COUNT 2 :	#03?		
1881	026700	045	116	045	L238::	.ASCIZ	?N#ADRIIVE STATE:	#06?		
1882	026737	045	116	045	L239::	.ASCIZ	?N#AREAD/WRITE STATE:	#06?		
1883	026776	045	116	045	L240::	.ASCIZ	?N#AOPERATION FLAGS:	#06?		
1884	027035	045	116	062	L241::	.ASCIZ	?N2#AERROR LOG (CONTROLLER ERROR)?			
1885	027077	045	116	045	L242::	.ASCIZ	?N#ACONTROLLER ERROR CODE:	#06?		
1886										
1887	027136	045	116	062	L244::	.ASCIZ	?N2#ABYTE READ: #03#A	BYTE WRITTEN: #03#A	BYTE ADDRESS: #06?	
1888	027232	045	116	045	L245::	.ASCIZ	?N#A	#03#A	#03#A	#06?
1889	027326	045	116	045	L246::	.ASCIZ	?N#A	#03#A	#03#A	#06?
1890	027422	045	116	045	L247::	.ASCIZ	?N#A	#03#A	#03#A	#06?
1891	027516	045	116	045	L248::	.ASCIZ	?N#A	#03#A	#03#A	#06?
1892	027612	045	116	045	L249::	.ASCIZ	?N#A	#03#A	#03#A	#06?
1893	027706	045	116	045	L250::	.ASCIZ	?N#A	#03#A	#03#A	#06?
1894	030002	045	116	045	L251::	.ASCIZ	?N#A	#03#A	#03#A	#06?
1895	030076	045	116	045	L252::	.ASCIZ	?N#A	#03#A	#03#A	#06?
1896	030172	045	116	045	L253::	.ASCIZ	?N#A	#03#A	#03#A	#06?
1897	030266	045	116	062	L254::	.ASCIZ	?N2#ATOTAL NUMBER OF MISCOMPARED BYTES:	#06?		
1898						.EVEN				
1899										
1900										
1901										
1902										
1903										
1904										
1905										
1906										

```

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

```

1907	030342	116	130	115	EMSG5::	.ASCIZ	?NXM ON READ TKIP?			
1908	030363	124	113	111	EMSG6::	.ASCIZ	?TKIP NOT 0 ON FIRST READ?			
1909	030414	116	130	115	EMSG7::	.ASCIZ	?NXM ON READ TKSA?			
1910	030435	123	101	040	EMSG8::	.ASCIZ	?SA REG IN ERROR ON FIRST READ?			
1911	030473	123	101	040	EMSG9::	.ASCIZ	?SA CONTENTS IN ERROR?			
1912	030520	123	101	040	EMSG10::	.ASCIZ	?SA WRONG IN DATA WRAP?			
1913	030546	105	130	120	EMSG11::	.ASCIZ	?EXPECTED INTERRUPT DID NOT OCCUR?			
1914	030607	111	116	124	EMSG12::	.ASCIZ	?INTRRPT OCCURRED WITH CPU PRIORITY = 7?			
1915	030656	123	101	040	EMSG13::	.ASCIZ	?SA NOT 0 IN PURGE/POLL?			
1916	030705	120	125	122	EMSG14::	.ASCIZ	?PURGE/POLL TEST FAILED?			
1917	030734	105	130	124	EMSG15::	.ASCIZ	?EXTENDED ADDRESS TEST FAILED?			
1918										
1919	030771	111	116	126	L2ER1::	.ASCIZ	?INVALID COMMAND?			
1920	031011	103	117	115	L2ER2::	.ASCIZ	?COMMAND ABORTED?			
1921	031031	125	116	111	L2ER3::	.ASCIZ	?UNIT-OFFLINE?			
1922	031046	125	116	111	L2ER4::	.ASCIZ	?UNIT-AVAILABLE?			
1923	031065	111	116	126	L2ER5::	.ASCIZ	?INVALID STATUS RECEIVED?			
1924	031115	127	122	111	L2ER6::	.ASCIZ	?WRITE PROTECTED?			
1925	031135	103	117	115	L2ER7::	.ASCIZ	?COMPARE ERROR?			
1926	031153	104	101	124	L2ER8::	.ASCIZ	?DATA ERROR?			
1927	031166	110	117	123	L2ER9::	.ASCIZ	?HOST BUFFER ACCESS ERROR?			
1928	031217	103	117	116	L2ER10::	.ASCIZ	?CONTROLLER ERROR?			
1929	031240	104	122	111	L2ER11::	.ASCIZ	?DRIVE ERROR?			
1930	031254	106	117	122	L2ER12::	.ASCIZ	?FORMATTER ERROR?			
1931	031274	102	117	124	L2ER13::	.ASCIZ	?BOT ENCOUNTERED?			
1932	031314	124	101	120	L2ER14::	.ASCIZ	?TAPE MARK ENCOUNTERED?			
1933	031342	111	116	126	L2ER15::	.ASCIZ	?INVALID STATUS RECEIVED?			

GLOBAL TEXT SECTION

```

1934 031372      122      105      103 L2ER16::ASCIZ ?RECORD DATA TRUNCATED?
1935 031420      120      117      123 L2ER17::ASCIZ ?POSITION LOST?
1936 031436      123      105      122 L2ER18::ASCIZ ?SERIOUS EXEPTION?
1937 031457      114      105      117 L2ER19::ASCIZ ?LEOT DETECTED?
1938 031475      115      111      103 L2ER20::ASCIZ ?MICRODIAGNOSTIC PROGRAM ERROR: DATA COMPARE ERROR?
1939 031557      115      111      103 L2ER21::ASCIZ ?MICRODIAGNOSTIC PROGRAM ERROR: CONTROLLER ERROR?
1940 031637      115      111      103 L2ER22::ASCIZ ?MICRODIAGNOSTIC PROGRAM ERROR: INVALID STATUS RECEIVED?
1941 031726      115      111      103 L2ER23::ASCIZ ?MICRODIAGNOSTIC PROGRAM ERROR: BAD PATTERN NUMBER ERROR?
1942 032016      115      111      103 L2ER24::ASCIZ ?MICRODIAGNOSTIC PROGRAM ERROR: RESPONSE ADDRESS ERROR?
1943 032104      115      111      103 L2ER25::ASCIZ ?MICRODIAGNOSTIC PROGRAM ERROR: HOST BUFFER ADDRESS ERROR?
1944 032175      115      111      103 L2ER26::ASCIZ ?MICRODIAGNOSTIC PROGRAM ERROR: UNKNOWN ERROR LOG RECEIVED?
1945 032267      115      111      103 L2ER27::ASCIZ ?MICRODIAGNOSTIC PROGRAM ERROR: RESPONSE TIMEOUT ERROR?
1946
1947 032355      124      111      115 WRER1::ASCIZ ?TIME OUT DURING PORT INIT?
1948 032407      120      117      122 WRER2::ASCIZ ?PORT INIT FAILED?
1949 032430      124      115      123 WRER3::ASCIZ ?TMSCP COMMAND FAILURE?
1950 032456      120      117      122 WRER4::ASCIZ ?PORT DETECTED ERROR?
1951 032502      122      105      123 WRER5::ASCIZ ?RESPONSE OUT OF SEQUENCE?
1952 032533      103      117      115 WRER6::ASCIZ ?COMMAND TIME OUT?
1953 032554      125      116      105 WRER7::ASCIZ ?UNEXPECTED STATUS FOR ONLINE RECEIVED?
1954 032622      104      125      120 WRER8::ASCIZ ?DUP COMMAND FAILURE?
1955
1956 032646      045      116      045 WRER9::ASCIZ ?NONMEDIA WRONG FORMAT?
1957 .EVEN
1958
1959 ;*****
1960 ;
1961 ;MISCELLANEOUS ERROR MESSAGES
1962 ;
1963 ;*****
1964
1965 032700      103      116      124 CTRL::ASCIZ ?CNTRLR/CABLE?
1966 032715      104      122      111 DRVE::ASCIZ ?DRIVE?
1967 032723      123      103      103 SCCCMD::ASCIZ ?SCC ?
1968 032730      117      116      114 ONLCMD::ASCIZ ?ONL ?
1969 9 032735      122      105      127 REWCMD::ASCIZ ?REW ?
1970 032742      127      122      040 WRCMD::ASCIZ ?WR ?
1971 032747      122      104      040 RDCMD::ASCIZ ?RD ?
1972 032754      107      104      123 GDSCMD::ASCIZ ?GDS ?
1973 032761      105      114      120 ELPCMD::ASCIZ ?ELP ?
1974 032766      122      103      126 RCVCMD::ASCIZ ?RCV ?
1975 032773      101      102      124 ABTCMD::ASCIZ ?ABT ?
1976 .EVEN
1977

```

1979
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1997
1998 033000
1999
2000 033000
2001 033000
033000 013746 002316
033004 012746 023426
033010 012746 000002
033014 010600
033016 104415
033020 062706 000006

2002
2003 033024
2004 033024
033024 016446 000012
033030 013746 002314
033034 016446 000002
033040 012746 023462
033044 012746 000004
033050 010600
033052 104415
033054 062706 000012
2005 033060 000452
2006
2007 033062
2008 033062
033062 010246
033064 012746 023754
033070 012746 000002
033074 010600
033076 104415
033100 062706 000006
2009 033104 000413
2010
2011 033106
2012 033106
033106 010246
033110 013746 172346
033114 012746 023634
033120 012746 000003
033124 010600
033126 104415
033130 062706 000010
2013
2014 033134

```
.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::
```


J4

2015	033134			PRINTX	#LINE6,R1,(R2)
	033134	011246		MOV	(R2),-(SP)
	033136	010146		MOV	R1, -(SP)
	033140	012746	023711	MOV	#LINE6, -(SP)
	033144	012746	000003	MOV	#3, -(SP)
	033150	010600		MOV	SP,R0
	033152	104415		TRAP	C\$PNTX
	033154	062706	000010	ADD	#10,SP
2016	033160	000412		BR	PRIERR

2017					
2018	033162			PRIIP::	
2019	033162			PRINTX	#LINE3,TKIP(R4)
	033162	016446	000000	MOV	TKIP(R4),-(SP)
	033166	012746	023542	MOV	#LINE3, -(SP)
	033172	012746	000002	MOV	#2, -(SP)
	033176	010600		MOV	SP,R0
	033200	104415		TRAP	C\$PNTX
	033202	062706	000006	ADD	#6,SP

2020					
2021	033206			PRIERR::	
2022	033206			PRINTB	#LINE4,FRUIS
	033206	013746	002310	MOV	FRUIS, -(SP)
	033212	012746	023572	MOV	#LINE4, -(SP)
	033216	012746	000002	MOV	#2, -(SP)
	033222	010600		MOV	SP,R0
	033224	104414		TRAP	C\$PNTB
	033226	062706	000006	ADD	#6,SP
2023	033232	000137	033576	JMP	PRIEX

2024					
2025	033236			WRINTO::	
2026	033236			PRINTX	#LINE1,INISTP
	033236	013746	002316	MOV	INISTP, -(SP)
	033242	012746	023426	MOV	#LINE1, -(SP)
	033246	012746	000002	MOV	#2, -(SP)
	033252	010600		MOV	SP,R0
	033254	104415		TRAP	C\$PNTX
	033256	062706	000006	ADD	#6,SP

2027					
2028	033262			WRPRTE::	
2029	033262			PRINTX	#WR1,TKSA(R4),TKSASV(R4)
	033262	016446	000012	MOV	TKSASV(R4), -(SP)
	033266	016446	000002	MOV	TKSA(R4), -(SP)
	033272	012746	024000	MOV	#WR1, -(SP)
	033276	012746	000003	MOV	#3, -(SP)
	033302	010600		MOV	SP,R0
	033304	104415		TRAP	C\$PNTX
	033306	062706	000010	ADD	#10,SP
2030	033312	000137	033552	JMP	WREX

2031					
2032	033316			WRCMDE::	
2033	033316			PRINTX	R2,CURCMD
	033316	013746	023044	MOV	CURCMD, -(SP)
	033322	010246		MOV	R2, -(SP)
	033324	012746	000002	MOV	#2, -(SP)
	033330	010600		MOV	SP,R0
	033332	104415		TRAP	C\$PNTX
	033334	062706	000006	ADD	#6,SP

2034	033340			PRINTX	#WR3
	033340	012746	024077	MOV	#WR3,-(SP)
	033344	012746	000001	MOV	#1,-(SP)
	033350	010600		MOV	SP,R0
	033352	104415		TRAP	C\$PNTX
	033354	062706	000004	ADD	#4,SP
2035	033360	010301		MOV	R3,R1
2036	033362			PRINTX	#WR4,2(R1),(R1)
	033362	011146		MOV	(R1),-(SP)
	033364	016146	000002	MOV	2(R1),-(SP)
	033370	012746	024130	MOV	#WR4,-(SP)
	033374	012746	000003	MOV	#3,-(SP)
	033400	010600		MOV	SP,R0
	033402	104415		TRAP	C\$PNTX
	033404	062706	000010	ADD	#10,SP
2037	033410	062701	000004	ADD	#4,R1
2038	033414	162763	000004 177774	SUB	#4,MSGLEN(R3)
2039	033422	001357		BNE	W1
2040	033424			PRINTX	#WR6
	033424	012746	024217	MOV	#WR6,-(SP)
	033430	012746	000001	MOV	#1,-(SP)
	033434	010600		MOV	SP,R0
	033436	104415		TRAP	C\$PNTX
	033440	062706	000004	ADD	#4,SP
2041	033444	000137	033552	JMP	WREX
2042					
2043	033450			WRSEGE::	PRINTX #WR2,CURCMD
2044	033450			MOV	CURCMD,-(SP)
	033450	013746	023044	MOV	#WR2,-(SP)
	033454	012746	024043	MOV	#2,-(SP)
	033460	012746	000002	MOV	SP,R0
	033464	010600		TRAP	C\$PNTX
	033466	104415		ADD	#6,SP
2045	033470	062706	000006	PRINTX	#WR5,P.CRF(R3),P.CRF(R5)
	033474			MOV	P.CRF(R5),-(SP)
	033474	016546	000000	MOV	P.CRF(R3),-(SP)
	033500	016346	000000	MOV	#WR5,-(SP)
	033504	012746	024155	MOV	#3,-(SP)
	033510	012746	000003	MOV	SP,R0
	033514	010600		TRAP	C\$PNTX
	033516	104415		ADD	#10,SP
2046	033520	062706	000010	BR	WREX
2047	033524	000412			
2048	033526			WRTOE::	PRINTX #WR2,CURCMD
2049	033526			MOV	CURCMD,-(SP)
	033526	013746	023044	MOV	#WR2,(SP)
	033532	012746	024043	MOV	#2,-(SP)
	033536	012746	000002	MOV	SP,R0
	033542	010600		TRAP	C\$PNTX
	033544	104415		ADD	#6,SP
	033546	062706	000006		
2050				WREX::	PRINTB #LINE4,FRUIS
2051	033552			MOV	FRUIS,(SP)
2052	033552			MOV	#LINE4,(SP)
	033552	013746	002310		
	033556	012746	023572		

L4

	033562	012746	000002		MOV	#2,-(SP)
	033566	010600			MOV	SP,R0
	033570	104414			TRAP	C\$PNTB
	033572	062706	000006		ADD	#6,SP
2053						
2054	033576			PRIEX:	EXIT	MSG
	033576	000167			.WORD	J\$JMP
	033600	000000			.WORD	L10003-2-
2055						
2056	033602				ENDMSG	
	033602			L10003:		
	033602	104423			TRAP	C\$MSG
2057						
2058	033604				BGNMSG	L2DUMP
	033604			L2DUMP::		
2059						
2060	033604				PRINTB	#LINE4,FRUIS
	033604	013746	002310		MOV	FRUIS,-(SP)
	033610	012746	023572		MOV	#LINE4,-(SP)
	033614	012746	000002		MOV	#2,-(SP)
	033620	010600			MOV	SP,R0
	033622	104414			TRAP	C\$PNTB
	033624	062706	000006		ADD	#6,SP
2061	033630				PRINTB	#L21
	033630	012746	024425		MOV	#L21,-(SP)
	033634	012746	000001		MOV	#1,-(SP)
	033640	010600			MOV	SP,R0
	033642	104414			TRAP	C\$PNTB
	033644	062706	000004		ADD	#4,SP
2062	033650				PRINTB	#L22,L2STA(R1)
	033650	016146	000002		MOV	L2STA(R1),-(SP)
	033654	012746	024501		MOV	#L22,-(SP)
	033660	012746	000002		MOV	#2,-(SP)
	033664	010600			MOV	SP,R0
	033666	104414			TRAP	C\$PNTB
	033670	062706	000006		ADD	#6,SP
2063	033674				PRINTB	#L23,L2DRV(R1)
	033674	016146	000004		MOV	L2DRV(R1),-(SP)
	033700	012746	024537		MOV	#L23,-(SP)
	033704	012746	000002		MOV	#2,-(SP)
	033710	010600			MOV	SP,R0
	033712	104414			TRAP	C\$PNTB
	033714	062706	000006		ADD	#6,SP
2064	033720				PRINTB	#L24,<B,L2TST(R1)>
	033720	005046			CLR	-(SP)
	033722	156116	000006		BISB	L2TST(R1),(SP)
	033726	012746	024577		MOV	#L24,-(SP)
	033732	012746	000002		MOV	#2,-(SP)
	033736	010600			MOV	SP,R0
	033740	104414			TRAP	C\$PNTB
	033742	062706	000006		ADD	#6,SP
2065	033746				PRINTB	#L25,<B,L2TRK(R1)>
	033746	005046			CLR	-(SP)
	033750	156116	000007		BISB	L2TRK(R1),(SP)
	033754	012746	024637		MOV	#L25,-(SP)
	033760	012746	000002		MOV	#2,-(SP)
	033764	010600			MOV	SP,R0

	033766	104414			TRAP	C\$PNTB
	033770	062706	000006		ADD	#6,SP
2066	033774				PRINTB	#L26,<B,L2CMD(R1)>
	033774	005046			CLR	-(SP)
	033776	156116	000010		BISB	L2CMD(R1),(SP)
	034002	012746	024677		MOV	#L26,-(SP)
	034006	012746	000002		MOV	#2,-(SP)
	034012	010600			MOV	SP,R0
	034014	104414			TRAP	C\$PNTB
	034016	062706	000006		ADD	#6,SP
2067	034022				PRINTB	#L27,<B,L2RSP(R1)>
	034022	005046			CLR	-(SP)
	034024	156116	000011		BISB	L2RSP(R1),(SP)
	034030	012746	024737		MOV	#L27,-(SP)
	034034	012746	000002		MOV	#2,-(SP)
	034040	010600			MOV	SP,R0
	034042	104414			TRAP	C\$PNTB
	034044	062706	000006		ADD	#6,SP
2068	034050				PRINTB	#L28,L2UNT(R1)
	034050	016146	000012		MOV	L2UNT(R1),-(SP)
	034054	012746	024777		MOV	#L28,-(SP)
	034060	012746	000002		MOV	#2,-(SP)
	034064	010600			MOV	SP,R0
	034066	104414			TRAP	C\$PNTB
	034070	062706	000006		ADD	#6,SP
2069	034074	126127	000134	000005	CMPB	L2ELFM(R1),#TXFER
2070	034102	001402			BEQ	1\$
2071	034104	000137	034760		JMP	10\$
2072	034110			1\$:	PRINTB	#L221
	034110	012746	025607		MOV	#L221,-(SP)
	034114	012746	000001		MOV	#1,-(SP)
	034120	010600			MOV	SP,R0
	034122	104414			TRAP	C\$PNTB
	034124	062706	000004		ADD	#4,SP
2073	034130				PRINTB	#L222,<B,L2ELFM(R1)>
	034130	005046			CLR	-(SP)
	034132	156116	000134		BISB	L2ELFM(R1),(SP)
	034136	012746	025654		MOV	#L222,-(SP)
	034142	012746	000002		MOV	#2,-(SP)
	034146	010600			MOV	SP,R0
	034150	104414			TRAP	C\$PNTB
	034152	062706	000006		ADD	#6,SP
2074	034156				PRINTB	#L223,<B,L2ELFL(R1)>
	034156	005046			CLR	-(SP)
	034160	156116	000135		BISB	L2ELFL(R1),(SP)
	034164	012746	025716		MOV	#L223,-(SP)
	034170	012746	000002		MOV	#2,-(SP)
	034174	010600			MOV	SP,R0
	034176	104414			TRAP	C\$PNTB
	034200	062706	000006		ADD	#6,SP
2075	034204				PRINTB	#L224,L2ELEV(R1)
	034204	016146	000136		MOV	L2ELEV(R1),-(SP)
	034210	012746	025760		MOV	#L224,-(SP)
	034214	012746	000002		MOV	#2,-(SP)
	034220	010600			MOV	SP,R0
	034222	104414			TRAP	C\$PNTB
	034224	062706	000006		ADD	#6,SP

```

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

```

N4

2076	034230			PRINTB	#L225,<B,L2ELRL(R1)>
	034230	005046		CLR	-(SP)
	034232	156116	000144	BISB	L2ELRL(R1),(SP)
	034236	012746	026017	MOV	#225,-(SP)
	034242	012746	000002	MOV	#2,-(SP)
	034246	010600		MOV	SP,R0
	034250	104414		TRAP	C\$PNTB
	034252	062706	000006	ADD	#6,SP
2077	034256			PRINTB	#L226,<B,L2ELRT(R1)>
	034256	005046		CLR	-(SP)
	034260	156116	000145	BISB	L2ELRT(R1),(SP)
	034264	012746	026061	MOV	#L226,-(SP)
	034270	012746	000002	MOV	#2,-(SP)
	034274	010600		MOV	SP,R0
	034276	104414		TRAP	C\$PNTB
	034300	062706	000006	ADD	#6,SP
2078	034304			PRINTB	#L227,L2ELP1(R1)
	034304	016146	000146	MOV	L2ELP1(R1),-(SP)
	034310	012746	026123	MOV	#L227,-(SP)
	034314	012746	000002	MOV	#2,-(SP)
	034320	010600		MOV	SP,R0
	034322	104414		TRAP	C\$PNTB
	034324	062706	000006	ADD	#6,SP
2079	034330			PRINTB	#L228,L2ELP2(R1)
	034330	016146	000150	MOV	L2ELP2(R1),-(SP)
	034334	012746	026162	MOV	#L228,-(SP)
	034340	012746	000002	MOV	#2,-(SP)
	034344	010600		MOV	SP,R0
	034346	104414		TRAP	C\$PNTB
	034350	062706	000006	ADD	#6,SP
2080	034354			PRINTB	#L229,<B,L2ELST(R1)>
	034354	005046		CLR	-(SP)
	034356	156116	000154	BISB	L2ELST(R1),(SP)
	034362	012746	026221	MOV	#L229,(SP)
	034366	012746	000002	MOV	#2,-(SP)
	034372	010600		MOV	SP,R0
	034374	104414		TRAP	C\$PNTB
	034376	062706	000006	ADD	#6,SP
2081	034402			PRINTB	#L230,<B,L2ELDE(R1)>
	034402	005046		CLR	-(SP)
	034404	156116	000155	BISB	L2ELDE(R1),(SP)
	034410	012746	026263	MOV	#L230,(SP)
	034414	012746	000002	MOV	#2,-(SP)
	034420	010600		MOV	SP,R0
	034422	104414		TRAP	C\$PNTB
	034424	062706	000006	ADD	#6,SP
2082	034430			PRINTB	#L231,<B,L2ELDF(R1)>
	034430	005046		CLR	-(SP)
	034432	156116	000156	BISB	L2ELDF(R1),(SP)
	034436	012746	026325	MOV	#L231,-(SP)
	034442	012746	000002	MOV	#2,-(SP)
	034446	010600		MOV	SP,R0
	034450	104414		TRAP	C\$PNTB
	034452	062706	000006	ADD	#6,SP
2083	034456			PRINTB	#L232,<B,L2ELTN(R1)>
	034456	005046		CLR	-(SP)
	034460	156116	000157	BISB	L2ELTN(R1),(SP)

	034464	012746	026367	MOV	#L232,-(SP)
	034470	012746	000002	MOV	#2,-(SP)
	034474	010600		MOV	SP,R0
	034476	104414		TRAP	C\$PNTB
	034500	062706	000006	ADD	#6,SP
2084	034504			PRINTB	#L233,L2ELPB(R1)
	034504	016146	000160	MOV	L2ELPB(R1),-(SP)
	034510	012746	026431	MOV	#L233,-(SP)
	034514	012746	000002	MOV	#2,-(SP)
	034520	010600		MOV	SP,R0
	034522	104414		TRAP	C\$PNTB
	034524	062706	000006	ADD	#6,SP
2085	034530			PRINTB	#L234,<B,L2ELLB(R1)>
	034530	005046		CLR	-(SP)
	034532	156116	000162	BISB	L2ELLB(R1),(SP)
	034536	012746	026470	MOV	#L234,-(SP)
	034542	012746	000002	MOV	#2,-(SP)
	034546	010600		MOV	SP,R0
	034550	104414		TRAP	C\$PNTB
	034552	062706	000006	ADD	#6,SP
2086	034556			PRINTB	#L235,<B,L2ELT0(R1)>
	034556	005046		CLR	-(SP)
	034560	156116	000163	BISB	L2ELT0(R1),(SP)
	034564	012746	026532	MOV	#L235,-(SP)
	034570	012746	000002	MOV	#2,-(SP)
	034574	010600		MOV	SP,R0
	034576	104414		TRAP	C\$PNTB
	034600	062706	000006	ADD	#6,SP
2087	034604			PRINTB	#L236,<B,L2ELT1(R1)>
	034604	005046		CLR	-(SP)
	034606	156116	000164	BISB	L2ELT1(R1),(SP)
	034612	012746	026574	MOV	#L236,-(SP)
	034616	012746	000002	MOV	#2,-(SP)
	034622	010600		MOV	SP,R0
	034624	104414		TRAP	C\$PNTB
	034626	062706	000006	ADD	#6,SP
2088	034632			PRINTB	#L237,<B,L2ELT2(R1)>
	034632	005046		CLR	-(SP)
	034634	156116	000165	BISB	L2ELT2(R1),(SP)
	034640	012746	026636	MOV	#L237,-(SP)
	034644	012746	000002	MOV	#2,-(SP)
	034650	010600		MOV	SP,R0
	034652	104414		TRAP	C\$PNTB
	034654	062706	000006	ADD	#6,SP
2089	034660			PRINTB	#L238,L2ELDS(R1)
	034660	016146	000166	MOV	L2ELDS(R1),-(SP)
	034664	012746	026700	MOV	#L238,-(SP)
	034670	012746	000002	MOV	#2,-(SP)
	034674	010600		MOV	SP,R0
	034676	104414		TRAP	C\$PNTB
	034700	062706	000006	ADD	#6,SP
2090	034704			PRINTB	#L239,L2ELRW(R1)
	034704	016146	000170	MOV	L2ELRW(R1),-(SP)
	034710	012746	026737	MOV	#L239,-(SP)
	034714	012746	000002	MOV	#2,-(SP)
	034720	010600		MOV	SP,R0
	034722	104414		TRAP	C\$PNTB

GLOBAL ERROR REPORT SECTION

2091	034724	062706	000006		ADD	#6,SP	
	034730				PRINTB	#L240,L2ELOF(R1)	
	034730	016146	000172		MOV	L2ELOF(R1),-(SP)	
	034734	012746	026776		MOV	#L240,-(SP)	
	034740	012746	000002		MOV	#2,-(SP)	
	034744	010600			MOV	SP,R0	
	034746	104414			TRAP	C\$PNTB	
2092	034750	062706	000006		ADD	#6,SP	
2093	034754	000137	035034		JMP	25\$	
2094	034760	126127	000134	000000 10\$:	CMPB	L2ELFM(R1),#CNTFR	; "CONTROLLER ERROR" ERROR LOG ?
2095	034766	001022			BNE	25\$; NO, NO ERROR LOG TO PRINT, EXIT
	034770				PRINTB	#L241	; DUMP "CONTROLLER ERROR" ERROR LOG
	034770	012746	027035		MOV	#L241,-(SP)	
	034774	012746	000001		MOV	#1,-(SP)	
	035000	010600			MOV	SP,R0	
	035002	104414			TRAP	C\$PNTB	
	035004	062706	000004		ADD	#4,SP	
2096	035010				PRINTB	#L242,L2ELEC(R1)	
	035010	016146	000140		MOV	L2ELEC(R1),-(SP)	
	035014	012746	027077		MOV	#L242,-(SP)	
	035020	012746	000002		MOV	#2,-(SP)	
	035024	010600			MOV	SP,R0	
	035026	104414			TRAP	C\$PNTB	
	035030	062706	000006		ADD	#6,SP	
2097	035034			25\$:	EXIT	MSG	
	035034	000167			.WORD	J\$JMP	
	035036	000000			.WORD	L10004 2-.	
2098							
2099	035040				ENDMSG		
	035040			L10004:			
	035040	104423			TRAP	C\$MSG	

D5

```

2101 .SBTTL STATISTICAL REPORT SECTION
2102 035042 STARS
          ;:*****
2103 035042 STARS
          ;:*****
2104 :
2105 : STATISTICAL REPORT SECTION
2106 : THESE ARE THE "PRINTS" STATEMENTS USED TO DUMP THE LEVEL 2
2107 : MICRODIAGNOSTIC TEST RESULTS
2108 :
2109 035042 STARS
          ;:*****
2110 035042 STARS
          ;:*****
2111 :
2112 035042 BGNRPT
          035042 L$RPT::
2113 035042 PRINTS #L29,L2BWR1(R1)
          035042 MOV L2BWR1(R1),-(SP)
          035046 MOV #L29,-(SP)
          035052 MOV #2,-(SP)
          035056 MOV SP,R0
          035060 TRAP C$PNTS
          035062 ADD #6,SP
2114 035066 PRINTS #L210,L2BWR2(R1)
          035066 MOV L2BWR2(R1),-(SP)
          035072 MOV #L210,-(SP)
          035076 MOV #2,-(SP)
          035102 MOV SP,R0
          035104 TRAP C$PNTS
          035106 ADD #6,SP
2115 035112 PRINTS #L211,L2BRD1(R1)
          035112 MOV L2BRD1(R1),-(SP)
          035116 MOV #L211,-(SP)
          035122 MOV #2,-(SP)
          035126 MOV SP,R0
          035130 TRAP C$PNTS
          035132 ADD #6,SP
2116 035136 PRINTS #L212,L2BRD2(R1)
          035136 MOV L2BRD2(R1),-(SP)
          035142 MOV #L212,-(SP)
          035146 MOV #2,-(SP)
          035152 MOV SP,R0
          035154 TRAP C$PNTS
          035156 ADD #6,SP
2117 035162 PRINTS #L213,L2SWR1(R1)
          035162 MOV L2SWR1(R1),-(SP)
          035166 MOV #L213,-(SP)
          035172 MOV #2,-(SP)
          035176 MOV SP,R0
          035200 TRAP C$PNTS
          035202 ADD #6,SP
2118 035206 PRINTS #L214,L2SWR2(R1)
          035206 MOV L2SWR2(R1),-(SP)
          035212 MOV #L214,-(SP)
          035216 MOV #2,-(SP)
          035222 MOV SP,R0

```


	035224	104416		TRAP	C\$PNTS
	035226	062706	000006	ADD	#6,SP
2119	035232			PRINTS	#L215,L2ECC1(R1)
	035232	016146	000040	MOV	L2ECC1(R1),-(SP)
	035236	012746	025365	MOV	#L215,-(SP)
	035242	012746	000002	MOV	#2,-(SP)
	035246	010600		MOV	SP,R0
	035250	104416		TRAP	C\$PNTS
	035252	062706	000006	ADD	#6,SP
2120	035256			PRINTS	#L216,L2ECC2(R1)
	035256	016146	000042	MOV	L2ECC2(R1),-(SP)
	035262	012746	025431	MOV	#L216,-(SP)
	035266	012746	000002	MOV	#2,-(SP)
	035272	010600		MOV	SP,R0
	035274	104416		TRAP	C\$PNTS
	035276	062706	000006	ADD	#6,SP
2121	035302			PRINTS	#L219,L2REP1(R1)
	035302	016146	000050	MOV	L2REP1(R1),-(SP)
	035306	012746	025475	MOV	#L219,-(SP)
	035312	012746	000002	MOV	#2,-(SP)
	035316	010600		MOV	SP,R0
	035320	104416		TRAP	C\$PNTS
	035322	062706	000006	ADD	#6,SP
2122	035326			PRINTS	#L220,L2REP2(R1)
	035326	016146	000052	MOV	L2REP2(R1),-(SP)
	035332	012746	025541	MOV	#L220,-(SP)
	035336	012746	000002	MOV	#2,-(SP)
	035342	010600		MOV	SP,R0
	035344	104416		TRAP	C\$PNTS
	035346	062706	000006	ADD	#6,SP
2123	035352	026127	000124 000000	CMP	124(R1),#0
2124	035360	001002		BNE	1\$
2125	035362	000137	036112	JMP	10\$
2126	035366			PRINTS	#L244,<B,66(R1)>,<B,54(R1)>,100(R1)
	035366	016146	000100	MOV	100(R1),-(SP)
	035372	005046		CLR	-(SP)
	035374	156116	000054	BISB	54(R1),(SP)
	035400	005046		CLR	-(SP)
	035402	156116	000066	BISB	66(R1),(SP)
	035406	012746	027136	MOV	#L244,-(SP)
	035412	012746	000004	MOV	#4,-(SP)
	035416	010600		MOV	SP,R0
	035420	104416		TRAP	C\$PNTS
	035422	062706	000012	ADD	#12,SP
2127	035426			PRINTS	#L245,<B,67(R1)>,<B,55(R1)>,102(R1)
	035426	016146	000102	MOV	102(R1),-(SP)
	035432	005046		CLR	-(SP)
	035434	156116	000055	BISB	55(R1),(SP)
	035440	005046		CLR	-(SP)
	035442	156116	000067	BISB	67(R1),(SP)
	035446	012746	027232	MOV	#L245,-(SP)
	035452	012746	000004	MOV	#4,-(SP)
	035456	010600		MOV	SP,R0
	035460	104416		TRAP	C\$PNTS
	035462	062706	000012	ADD	#12,SP
2128	035466			PRINTS	#L246,<B,70(R1)>,<B,56(R1)>,104(R1)
	035466	016146	000104	MOV	104(R1),-(SP)

```

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

```

1\$:

	035472	005046		CLR	-(SP)
	035474	156116	000056	BISB	56(R1),(SP)
	035500	005046		CLR	-(SP)
	035502	156116	000070	BISB	70(R1),(SP)
	035506	012746	027326	MOV	#L246,-(SP)
	035512	012746	000004	MOV	#4,-(SP)
	035516	010600		MOV	SP,R0
	035520	104416		TRAP	C\$PNTS
	035522	062706	000012	ADD	#12,SP
2129	035526			PRINTS	#L247,<B,71(R1)>,<B,57(R1)>,106(R1)
	035526	016146	000106	MOV	106(R1),-(SP)
	035532	005046		CLR	-(SP)
	035534	156116	000057	BISB	57(R1),(SP)
	035540	005046		CLR	-(SP)
	035542	156116	000071	BISB	71(R1),(SP)
	035546	012746	027422	MOV	#L247,-(SP)
	035552	012746	000004	MOV	#4,-(SP)
	035556	010600		MOV	SP,R0
	035560	104416		TRAP	C\$PNTS
	035562	062706	000012	ADD	#12,SP
2130	035566			PRINTS	#L248,<B,72(R1)>,<B,60(R1)>,110(R1)
	035566	016146	000110	MOV	110(R1),-(SP)
	035572	005046		CLR	-(SP)
	035574	156116	000060	BISB	60(R1),(SP)
	035600	005046		CLR	-(SP)
	035602	156116	000072	BISB	72(R1),(SP)
	035606	012746	027516	MOV	#L248,-(SP)
	035612	012746	000004	MOV	#4,-(SP)
	035616	010600		MOV	SP,R0
	035620	104416		TRAP	C\$PNTS
	035622	062706	000012	ADD	#12,SP
2131	035626			PRINTS	#L249,<B,73(R1)>,<B,61(R1)>,112(R1)
	035626	016146	000112	MOV	112(R1),-(SP)
	035632	005046		CLR	-(SP)
	035634	156116	000061	BISB	61(R1),(SP)
	035640	005046		CLR	-(SP)
	035642	156116	000073	BISB	73(R1),(SP)
	035646	012746	027612	MOV	#L249,-(SP)
	035652	012746	000004	MOV	#4,-(SP)
	035656	010600		MOV	SP,R0
	035660	104416		TRAP	C\$PNTS
	035662	062706	000012	ADD	#12,SP
2132	035666			PRINTS	#L250,<B,74(R1)>,<B,62(R1)>,114(R1)
	035666	016146	000114	MOV	114(R1),-(SP)
	035672	005046		CLR	-(SP)
	035674	156116	000062	BISB	62(R1),(SP)
	035700	005046		CLR	-(SP)
	035702	156116	000074	BISB	74(R1),(SP)
	035706	012746	027706	MOV	#L250,-(SP)
	035712	012746	000004	MOV	#4,-(SP)
	035716	010600		MOV	SP,R0
	035720	104416		TRAP	C\$PNTS
	035722	062706	000012	ADD	#12,SP
2133	035726			PRINTS	#L251,<B,75(R1)>,<B,63(R1)>,116(R1)
	035726	016146	000116	MOV	116(R1),-(SP)
	035732	005046		CLR	-(SP)
	035734	156116	000063	BISB	63(R1),(SP)

STATISTICAL REPORT SECTION

	035740	005046		CLR	-(SP)
	035742	156116	000075	BISB	75(R1),(SP)
	035746	012746	030002	MOV	#L251,-(SP)
	035752	012746	000004	MOV	#4,-(SP)
	035756	010600		MOV	SP,R0
	035760	104416		TRAP	C\$PNTS
	035762	062706	000012	ADD	#12,SP
2134	035766			PRINTS	#L252,<B,76(R1)>,<B,64(R1)>,120(R1)
	035766	016146	000120	MOV	120(R1),-(SP)
	035772	005046		CLR	-(SP)
	035774	156116	000064	BISB	64(R1),(SP)
	036000	005046		CLR	-(SP)
	036002	156116	000076	BISB	76(R1),(SP)
	036006	012746	030076	MOV	#L252,-(SP)
	036012	012746	000004	MOV	#4,-(SP)
	036016	010600		MOV	SP,R0
	036020	104416		TRAP	C\$PNTS
	036022	062706	000012	ADD	#12,SP
2135	036026			PRINTS	#L253,<B,77(R1)>,<B,65(R1)>,122(R1)
	036026	016146	000122	MOV	122(R1),-(SP)
	036032	005046		CLR	-(SP)
	036034	156116	000065	BISB	65(R1),(SP)
	036040	005046		CLR	-(SP)
	036042	156116	000077	BISB	77(R1),(SP)
	036046	012746	030172	MOV	#L253,-(SP)
	036052	012746	000004	MOV	#4,-(SP)
	036056	010600		MOV	SP,R0
	036060	104416		TRAP	C\$PNTS
	036062	062706	000012	ADD	#12,SP
2136	036066			PRINTS	#L254,124(R1)
	036066	016146	000124	MOV	124(R1),-(SP)
	036072	012746	030266	MOV	#L254,-(SP)
	036076	012746	000002	MOV	#2,-(SP)
	036102	010600		MOV	SP,R0
	036104	104416		TRAP	C\$PNTS
	036106	062706	000006	ADD	#6,SP
2137	036112			ENDRPT	
	036112				
	036112	104425		TRAP	C\$RPT

10\$:
L10005:

GLOBAL SUBROUTINES SECTION

2139
 2143
 2144
 2145
 2146
 2147
 2148
 2149
 2150
 2151
 2152
 2153
 2154
 2155
 2156
 2157
 2158
 2159
 2160
 2161
 2162
 2163
 2164
 2165
 2166
 2167
 2168
 2169
 2170
 2174
 2175 036114
 036114
 2176
 2177 036114 005237 002276
 2178
 2179 036120
 036120
 036120 000002
 2180

```
.SBTTL GLOBAL SUBROUTINES SECTION
;*****
;*****
:GLOBAL SUBROUTINES SECTION
:THIS SECTION CONTAINS ALL SUBROUTINES AND
:INTERRUPT SERVICE ROUTINES THAT ARE AC-
:CESSED 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.
;*****
;*****
BGNSRV TRAP4
TRAP4::
INC TRP4FG ;SET THE FLAG - TRAP OCCURRED
ENDSRV
L10006:
RTI
```

GLOBAL SUBROUTINES SECTION

2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216

036122
036122

052764 000002 000014
036130
036130
036130 000002

```

;*****
;*****
;INTRCV
;   THIS IS THE TKxx INTERRUPT HANDLER USED BY THE PRO-
;   GRAM WHEN INTERRUPTS HAVE BEEN ENABLED.  IF THE
;   BRFLAG IS CLEAR, THE ROUTINE SETS A FLAG INDICATING
;   THE EXPECTED INTERRUPT OCCURRED.  IF BRFLAG IS SET,
;   IT INDICATES THAT PROCESOR PRIORITY WAS SET TO A
;   LEVEL THAT SHOULD HAVE INHIBITED THE INTERRUPT, SO
;   THE ROUTINE SETS AN ERROR INDICATOR.
;*****
;*****

```

```

BGNSRV INTRCV
INTRCV::
;   BIT   #BRFLAG,LUNFLG(R4)   ;IF NOT PRIORITY LEVEL TESTING
;   BEQ   5$                   ; THEN SKIP AROUND
;   MOV   #DRPFLG,LUNFLG(R4)   ; ELSE SET FAILED BIT
;   BR    EXTINT               ;RETURN
2212 036122 052764 000002 000014 5$: BIS   #INTFLG,LUNFLG(R4)   ;SET THE FLAG
EXTINT:
ENDSRV
L10007: RTI

```

GLOBAL SUBROUTINES SECTION

```

2221
2222      ;*****
2223      ;*****
2224      ;
2225      ;ILLINT
2226      ;      THIS HANDLER ROUTINE'S ADDRESS IS LOADED INTO THE
2227      ;      CURRENT UUT'S VECTOR FOR ALL TESTS THAT DO NOT EN-
2228      ;      ABLE DEVICE INTERRUPTS.
2229      ;
2230      ;*****
2231      ;*****
2235
2236 036132      BGNSRV ILLINT
      036132 ILLINT::
2237
2238 036132 052764 000001 000014      BIS      #DRPFLG,LUNFLG(R4)
2239
2240
2241 036140      ENDSRV
      036140 L10010:
      036140 000002      RTI

```

2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2262
2263
2264
036142
036142
036146
036152
036156
036162
036164
2265
2266
036174
036174
036200
036202
036206
036210
036214
036216
036222
2267
2268
2269
2270
2271
2272
2273
036242
036246
036250
036254
036256
036262
036264
036270
2274
2275
2276
2277
2278
2279
2280
2281
2282
036312
036312

```

;*****
;*****
;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

```

L5

GLOBAL SUBROUTINES SECTION

2283	036316	104436		TRAP	C\$CVEC	
2284	036320	005037	002276	CLR	TRP4FG	;MORE HOUSEKEEPING
2285	036324	000207		RTS	PC	
2286						

GLOBAL SUBROUTINES SECTION

2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2310
2311
2312
2313
2314

036326
036326 012746 000000
036332 012746 036132
036336 016446 000004
036342 012746 000003
036346 104437
036350 062706 000010
036354 000207

```
*****  
*****  
: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
```

GLOBAL SUBROUTINES SECTION

2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2342
2343
2344
2345

2346
2347
2348

2349
2350
2351
2352

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

036356
036356 032764 000004 000014
036364 001014
036366
036366 012746 000000
036372 012746 036122
036376 016446 000004
036402 012746 000003
036406 104437
036410 062706 000010
036414 000413

036416
036416 012746 000340
036422 012746 036122
036426 016446 000004
036432 012746 000003
036436 104437
036440 062706 000010

036444 000207

GLOBAL SUBROUTINES SECTION

2357
 2358
 2359
 2360
 2361
 2362
 2363
 2364
 2365
 2366
 2367
 2368
 2369
 2370
 2371
 2372
 2373
 2374
 2378
 2379
 2380
 2381
 2382
 2383
 2384
 2385
 2386
 2387
 2388
 2389

036446
 036446 005037 002330
 036452 005337 002324
 036456 001373
 036460 005337 002326
 036464 001002
 036466 005237 002330
 036472 000207

```

:*****
:*****
: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
    
```

GLOBAL SUBROUTINES SECTION

2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418

2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436

036474
036474 005037 002320
036500 012774 000000 000000
036506 012727 000001
036512 000000
036514 013727 002116
036520 000000
036522 005367 177772
036526 001375
036530 005367 177756
036534 001367

036536 017464 000002 000012
036544 022764 005700 000012

036552 001413
036554 022764 004700 000012
036562 001013
036564 022737 007776 002120
036572 002003
036574 012737 007776 002120

036602 013774 002250 000002 1\$:
036610 000402

036612 005237 002320
036616 000207

```
*****  
*****  
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)  
  
BEQ 1$ ;IF QBUS, BRANCH  
CMP #4700,TKSASV(R4) ;IF NOT UNIBUS  
BNE STP1ER ; THEN TAKE ERROR EXIT  
CMP #7776,L$HIME ; ELSE IF LESS THAN OR EQUAL TO 128K MEMORY  
BGE 1$ ; THEN BRANCH  
MOV #7776,L$HIME ; ELSE SET UP L$HIME  
  
1$: MOV STPTBL,@TKSA(R4); WRITE HOST'S STEP 1 RESPONSE  
BR STP1EX ; AND LEAVE SHOWING SUCCESS  
  
STP1ER: INC STEPST ;SET ERROR INDICATOR  
STP1EX: RTS PC
```

D6

GLOBAL SUBROUTINES SECTION

```

2441 ;:*****
2442 ;:*****
2443 ;:
2444 ;BAKPAT
2445 ; THIS SUBROUTINE WILL FILL THE COMMUNICATION WITH AN
2446 ; ALL 1'S DATA PATTERN. THE LENGTH OF THE AREA IN USE
2447 ; BY THE CURRENT TEST IS CONTAINED IN "CMARLG".
2448 ;:
2449 ;:*****
2450 ;:*****
2454 ;:

```

```

2455 036620
2456 036620 012702 060000
2457
2458 036624 012703 000024
2459 036630 006303
2460 036632 063703 002306
2461 036636 012722 177777
2462 036642 005303
2463 036644 001374
2464
2465 036646 000207
2466

```

```

BAKPAT::
MOV    #COMMBF,R2      ;STARTING ADDRESS OF COMM AREA
                ; -20 WORDS
MOV    #20.,R3        ;BUFFER LENGTH IN FRONT OF AREA
ASL    R3              ;MULTIPLIED BY 2
ADD    CMARLG,R3      ;ADD COMM AREA LENGTH USED
1$:    MOV    #-1,(R2)+ ;WRITE THE DATA
        DEC    R3      ;IF NOT DONE YET
        BNE   1$      ; THEN DO IT AGAIN
RTS    PC

```

2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2488
2489 036650
2490 036650 012701 177777
2491 036654 012702 060000
2492 036660 012703 000022
2493 036664 020122
2494 036666 001022
2495 036670 005303
2496 036672 001374
2497
2498 036674 005001
2499 036676 013703 002306
2500 036702 005722
2501 036704 001013
2502 036706 005303
2503 036710 001374
2504
2505 036712 012701 177777
2506 036716 012703 000024
2507 036722 020122
2508 036724 001003
2509 036726 005303
2510 036730 001374
2511 036732 000422
2512
2513 036734 162702 000002
2514 036740 022737 000010 002114
2515 036746 001405
2516 036750
036750 104455
036752 000001
036754 030705
036756 033062
2517 036760 000404
2518
2519 036762
036762 104455
036764 000002
036766 030734
036770 033106
2520
2521 036772
036772 013700 002312

```
*****  
*****  
:CHKCOM  
: THIS ROUTINE IS CALLED BY TESTS DOING THE PURGE/POLL  
: CHECK. IT IS USED TO VERIFY THAT THE PORT LEFT THE  
: COMMUNICATIONS AREA CLEARED. ADDITIONALLY, IT CHECKS  
: THE 20 WORDS PRECEDING AND SUCCEEDING THE COMM AREA  
: TO MAKE SURE THE PORT DIDN'T GO OUTSIDE THE COMM AREA.  
:*****  
:*****
```

```
CHKCOM: :  
:TEST DATA  
:STARTING ADDRESS  
:FIRST COUNT  
1$: MOV # -1,R1 ;TEST DATA  
MOV #COMMBF,R2 ;STARTING ADDRESS  
MOV #18,R3 ;FIRST COUNT  
CMP R1,(R2) ;IF NOT ALL 1'S  
BNE 15$ ; THEN GO REPORT ERROR  
DEC R3 ;IF NOT ALL DONE  
BNE 1$ ; THEN GO CHECK ANOTHER  
  
5$: CLR R1 ;TEST DATA FOR PRINTOUT  
MOV CMARLG,R3 ;SET UP COUNTER FOR COMM AREA  
TST (R2) ;IF NOT 0  
BNE 15$ ; THEN GO REPORT ERROR  
DEC R3 ;IF NOT ALL DONE  
BNE 5$ ; THEN GO CHECK ANOTHER  
  
10$: MOV # -1,R1 ;TEST DATA FOR PRINTOUT  
MOV #20,R3 ;SET UP COUNTER FOR POST COMM AREA  
CMP R1,(R2) ;IF NOT ALL 1'S  
BNE 15$ ; 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
```

F6

GLOBAL AREAS MACRO V05.05 Wednesday 27 Feb-91 11:55 Page 38-1

SEG 0070

GLOBAL SUBROUTINES SECTION

2522	036776	104451		TRAP	C4DODU
2523	037000	000207	CKCMEX:	RTS	PC
2524					

2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
037152
037152
037154

037002
037002 012703 172300
037006 012702 172340
037012 005001
037014 010122
037016 012723 077406
037022 062701 000200
037026 022701 002000
037032 001370
037034 010137 172346
037040 012737 007600 172356
037046 032737 000002 002274
037054 001406
037056 012737 177600 172356
037064 012737 000020 172516
037072 012737 000001 177572
037100 000207
037102 010174 000000
037106 012703 037324
037112 012701 004000
037116 005037 002316
037122 012737 000030 003026
037130 005002
037132 005202
037134 001016
037136 005337 003026
037142 001013
037144 017464 000002 000012
104455
000063

```

;*****
;*****
:INTMMU
: THIS SUBROUTINE IS CALLED FROM TEST 8 TO INITIALIZE
: MEMORY MANAGEMENT REGISTERS. ALL PAR'S EXCEPT ONE
: ARE SET UP TO MAP VIRTUAL ADDRESSES INTO THE LOWEST
: 32K OF PHYSICAL MEMORY. KPAR7 IS SET UP TO MAP TO
: THE I/O PAGE. THE PAR REGISTER THAT CORRESPONDS TO
: THE VIRTUAL ADDRESS OF THE COMMUNICATION AREA IS SET
: UP TO POINT TO THE SECOND 32K OF PHYSICAL MEMORY.
: ALL PDR'S ARE INITIALIZED TO THE SAME VALUE; NAMELY,
: UPWARD EXPANDABLE, READ/WRITE ACCESS ENABLED, AND THE
: FULL 8KBYTE PAGE IS ACCESSIBLE.
;*****
;*****
INTMMU::
MOV #KPDRO,R3 ;START OF PDR ADDRESS RANGE
MOV #KPARO,R2 ;START OF PAR ADDRESS RANGE
CLR R1 ;STARTING RELOCATION VALUE

1$: MOV R1,(R2)+ ;LOAD RELOCATION VALUE
MOV #77406,(R3)+ ;LOAD PDR
ADD #200,R1 ;ADJUST RELOCATION VALUE
CMP #2000,R1 ;IF NOT AT THE END
BNE 1$ ; THEN DO ANOTHER ONE

MOV R1,KPAR3 ; ELSE SET THIS REG TO NEXT 32K
MOV #7600,KPAR7 ;18 BIT I/O PAGE
BIT #BIT1,KTFLAG ;IF 22-BIT BUS NOT AVAILABLE
BEQ 2$ ; THEN GO TURN MMU ON
MOV #177600,KPAR7 ; ELSE SET 22 BIT I/O PAGE
MOV #MM22ON,MMUSR3 ; AND ENABLE 22 BIT MAPPING

2$: MOV #MMON,MMUSRO ;TURN ON THE WHOLE THING
RTS PC

PRTINT::
MOV R1,@TKIP(R4) ;INITIALIZE THE DRIVE
MOV #INTTBL,R3 ;PUT THE TABLE ADDRESS INTO R3
MOV #S1,R1 ;SET UP TO BEGIN AT STEP 1
CLR INISTP ;CLEAR THE STEP TRACKER
LOOP: MOV #24.,CNTHI ;SET UP THE TIME OUT COUNTER
CLR R2 ;CLEAR R2
ILOOP: INC R2 ;INCREMENT HI TIME OUT VALUE ?
BNE 2$ ;IF NOT, BRANCH
DEC CNTHI ;ELSE, DECREMENT LO TIMEOUT
BNE 2$ ;BRANCH IF NO TIME OUT
MOV @TKSA(R4),TKSASV(R4) ;SAVE THE SA FOR THE ERROR PRINTOUT
ERRDF 51.,WRER1,WRINTO ;PRINT PORT INIT FAILURE
TRAP C$ERDF
.WORD 51

```


GLOBAL SUBROUTINES SECTION

```

037156 032355 .WORD WRER1
037160 033236 .WORD WRINTO
2587 037162 DODU LOGUNT ;DROP THE UNIT
037162 013700 002312 MOV LOGUNT,R0
037166 104451 TRAP C$DODU
2588 037170 000454 BR 100$ ;EXIT ROUTINE
2589 037172 037401 000002 2$: BIT @TKSA(R4),R1 ;TEST FOR STEP BIT FROM DRIVE
2590 037176 001755 BEQ ILOOP ;LOOP UNTIL SOMETHING SETS
2591 037200 032774 100000 000002 BIT #ERR,@TKSA(R4) ;CHECK FOR ERROR
2592 037206 001413 BEQ 3$ ;NO ERROR, KEEP GOING
2593 037210 017464 000002 000012 MOV @TKSA(R4),TKSASV(R4) ;SAVE THE SA CONTENTS
2594 037216 ERRDF 52.,WRER2,WRPTE ;PRINT ERROR
037216 104455 TRAP C$ERDF
037220 000064 .WORD 52
037222 032407 .WORD WRER2
037224 033262 .WORD WRPTE
2595 037226 DODU LOGUNT ;DROP THE UNIT
037226 013700 002312 MOV LOGUNT,R0
037232 104451 TRAP C$DODU
2596 037234 000432 BR 100$ ;EXIT ROUTINE
2597 037236 005237 002316 3$: INC INISTP ;INCREMENT THE STEP TRACKER
2598 037242 012374 000002 MOV (R3)+,@TKSA(R4) ;WRITE WORD FROM TABLE TO CONTROLLER
2599 037246 006301 ASL R1 ;SHIFT TO NEXT STEP
2600 037250 100324 BPL LOOP ;IF NOT AT LAST STEP LOOP
2601 037252 012702 003006 MOV #RSPRNG,R2 ;PUT THE RESPONSE DESCRIPTOR ADD IN R2
2602 037256 012703 002576 MOV #RSPBUF,R3 ;PUT THE RESPONSE BUFFER ADDRESS IN R3
2603 037262 010322 5$: MOV R3,(R2)+ ;PUT THE BUFF ADD IN THE DESCRIPTOR
2604 037264 012722 100000 MOV #OWN,(R2)+ ;SET THE DESCRIPTOR TO THE CONTROLLER
2605 037270 062703 000104 ADD #RSPSTP,R3 ;STEP TO THE NEXT BUFFER SLOT
2606 037274 022703 003006 CMP #RSPEND,R3 ;ARE WE AT THE END OF THE BUFFER ?
2607 037300 001370 BNE 5$ ;NO, KEEP GOING
2608 037302 012737 003006 023042 MOV #RSPRNG,RSPSAV ;SET UP TO USE FIRST RESPONSE BUFFER
2609 037310 012737 003016 023040 MOV #CMDRNG,CMDSAV ;SET UP TO USE FIRST COMMAND BUFFER
2610 037316 005037 003034 CLR CMDREF ;SET THE COMMAND REFERENCE # TO 0
2611 037322 000207 100$: RTS PC ;RETURN
2612
2613 ;INIT DATA TABLE
2614 037324 104400 INTTBL: .WORD 104400
2615 037326 003006 .WORD RSPRNG
2616 037330 000000 .WORD 0
2617 037332 000001 .WORD GO

```

```

2619 037334          CLSDRV::
2620 037334 005237 003034 5$: INC CMDREF ;ADD 1 TO THE COMMAND REFERENCE NUMBER
2621 037340 004737 040004 JSR PC,PRTDRV ;GO SEND THE COMMAND
2622 037344 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
2623 037352 001006 BNE 80$ ;GET OUT IF NOT AVAILABLE
2624 037354 004737 040146 JSR PC,CDRECV ;GO CHECK FOR ANY NEW RESPONSES
2625 037360 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE ?
2626 037366 001402 BEQ 6$ ;YES - BRANCH
2627 037370 000137 040002 80$: JMP 100$ ;GET OUT IF NOT AVAILABLE
2628 037374 012702 024043 6$: MOV #WR2,R2 ;R2 GETS ERROR MESSAGE ADDRESS
2629 037400 005763 000012 TST P.STS(R3) ;WAS STATUS "NORMAL"?
2630 037404 001524 BEQ 15$ ;YES - BRANCH
2631 037406 032764 000010 000014 7$: BIT #DUPFLG,LUNFLG(R4) ;IS IT A DUP COMMAND ?
2632 037414 001110 BNE 12$ ;YES, PRINT DUP ERROR
2633 037416 022705 002410 CMP #ONLINE,R5 ;IS IT AN ONLINE COMMAND
2634 037422 001045 BNE 10$ ;NO, GO PRINT ERROR
2635
2636 ;-----
2637 ; Failing FRU printout fix ***FIX-RC
2638 ;-----
2639 037424 016337 000012 002334 MOV P.STS(R3),TEMP1 ;GET STATUS
2640 037432 042737 177740 002334 BIC #177740,TEMP1 ;LEAVE ERROR BITS SET
2641 037440 022737 000003 002334 CMP #3,TEMP1 ;DO WE HAVE A DRIVE ERROR ?
2642 037446 001404 BEQ 40$ ;YES-SET FRUIS
2643 037450 022737 000013 002334 CMP #13,TEMP1 ;NO-CHECK THE OTHER POSSIBILITY
2644 037456 001003 BNE 50$ ;NOT IT EITHER, FRUIS OK
2645 037460 012737 032715 002310 40$: MOV #DRVE,FRUIS ;FRU IS THE DRIVE
2646 ;-----
2647
2648 037466 122763 000003 000012 50$: CMPB #3,P.STS(R3) ;WAS IT A UNIT UNKNOWN ERROR ?
2649 037474 001003 BNE 8$ ;NO, CONTINUE
2650 037476 012702 024222 MOV #WR7,R2 ;R2 GETS ERROR MESSAGE ADDRESS
2651 037502 000415 BR 10$ ;PRINT ERROR MESSAGE
2652 037504 122763 000043 000012 8$: CMPB #43,P.STS(R3) ;WAS IT A NO MEDIA MOUNTED ERROR ?
2653 037512 001003 BNE 9$ ;NO, CONTINUE
2654 037514 012702 024273 MOV #WR8,R2 ;R2 GETS ERROR MESSAGE ADDRESS
2655 037520 000406 BR 10$ ;PRINT ERROR MESSAGE
2656 037522 122763 000103 000012 9$: CMPB #103,P.STS(R3) ;WAS IT A UNIT INOPERATIVE ERROR ?
2657 037530 001012 BNE 11$ ;NO, MUST BE INVALID STATUS
2658 037532 012702 024273 MOV #WR8,R2 ;R2 GETS ERROR MESSAGE ADDRESS
2659 037536 104455 104455 10$: ERRDF 57,WRER3,WRCMDE ;GET READY TO PRINT FAILURE
    037536 104455 TRAP C$ERDF
    037540 000071 .WORD 57
    037542 032430 .WORD WRER3
    037544 033316 .WORD WRCMDE
2660 037546 013700 002312 DODU LOGUNT ;DROP THE UNIT
    037546 013700 002312 MOV LOGUNT,R0
    037552 104451 TRAP C$DODU
2661 037554 000512 BR 100$ ;GET OUT ON ERROR
2662
2663 ;-----
2664 ; Data Safety Write Protect check ***FIX RC
2665 ;-----
2666 037556 022763 010000 000012 11$: CMP #10000,P.STS(R3) ;DSWP SET ?
2667 037564 001014 BNE 90$ ;NO-BRANCH OVER
2668 037566 012746 032646 PRINTF #WRER9 ;YES-PRINT MESSAGE
    MOV #WRER9,-(SP)

```

```

037572 012746 000001      MOV      #1,-(SP)
037576 010600      MOV      SP,R0
037600 104417      TRAP     C$PNTF
037602 062706 000004      ADD      #4,SP
2669 037606      DODU     LOGUNT      ;AND DROP UNIT
037606 013700 002312      MOV      LOGUNT,R0
037612 104451      TRAP     C$DODU
2670 037614 000472      BR       100$      ;GET OUT ON DSWP
2671
2672
2673 037616      90$:     ERRDF    58.,WRER7,WRCMDF      ;GET READY TO PRINT FAILURE
037616 104455      TRAP     C$ERDF
037620 000072      .WORD   58
037622 032554      .WORD   WRER7
037624 033316      .WORD   WRCMDE
2674 037626      DODU     LOGUNT      ;DROP THE UNIT
037626 013700 002312      MOV      LOGUNT,R0
037632 104451      TRAP     C$DODU
2675 037634 000462      BR       100$      ;GET OUT ON ERROR
2676 037636      12$:     ERRDF    59.,WRER8,WRCMDE      ;GET READY TO PRINT FAILURE
037636 104455      TRAP     C$ERDF
037640 000073      .WORD   59
037642 032622      .WORD   WRER8
037644 033316      .WORD   WRCMDE
2677 037646      DODU     LOGUNT      ;DROP THE UNIT
037646 013700 002312      MOV      LOGUNT,R0
037652 104451      TRAP     C$DODU
2678 037654 000452      BR       100$      ;GET OUT ON ERROR
2679 037656 022705 002460      15$:     CMP      #GDUST,R5      ;WAS IT A GET DUST STATUS COMMAND ?
2680 037662 001035      BNE     20$          ;NO, CONTINUE
2681 037664 126327 000014 000113      CMPB    P.EXT1(R3),#113      ;1ST BYTE OF PROGRAM EXTENSION = "K" ?
2682 037672 001321      BNE     10$          ;NO, ERROR
2683 037674 126327 000015 000065      CMPB    P.EXT2(R3),#65      ;2ND BYTE OF PROGRAM EXTENSION = "5" ?
2684 037702 001315      BNE     10$          ;NO, ERROR
2685 037704 126327 000016 000060      CMPB    P.EXT3(R3),#60      ;3RD BYTE OF PROGRAM EXTENSION = "0" ?
2686 037712 001311      BNE     10$          ;NO, ERROR
2687 037714 126327 000017 000007      CMPB    DUSTFL(R3),#7      ;FLAGS = 7 ?
2688 037722 001305      BNE     10$          ;NO, ERROR
2689 037724 026327 000020 000000      CMP     P.IND1(R3),#0      ;1ST WORD OF PROGRESS INDICATOR = 0 ?
2690 037732 001301      BNE     10$          ;NO, ERROR
2691 037734 026327 000022 000000      CMP     P.IND2(R3),#0      ;2ND WORD OF PROGRESS INDICATOR = 0 ?
2692 037742 001275      BNE     10$          ;NO, ERROR
2693 037744 026327 000024 000012      CMP     TIMOUT(R3),#12      ;TIMEOUT VALUE = 12 ?
2694 037752 001271      BNE     10$          ;NO, ERROR
2695 037754 000405      BR       30$          ;NO ERRORS, RETURN
2696 037756 022705 002526      20$:     CMP     #RCVDAT,R5      ;WAS IT A RECEIVE DATA COMMAND?
2697 037762 001002      BNE     30$          ;NO, CONTINUE
2698 037764 004737 040376      JSR     PC,L2DATA      ;YES, GO READ DIAG 2 PROGRAM STATUS
2699 037770 005337 002272      30$:     DEC     ITRCNT      ;SUBTRACT 1 FROM TIMES TO DO
2700 037774 001402      BEQ     100$         ;ALL DONE
2701 037776 000137 037334      JMP     5$           ;GO SEND IT AGAIN
2702 040002 000207      100$:    RTS      PC      ;RETURN
2703
2704 040004      PRTDRV::
2705 040004 013701 023040      MOV     CMDSAV,R1      ;SET UP COMMAND RING POINTER
2706 040010 013765 003034 000000      MOV     CMDREF,P.CRF(R5) ;PUT COMMAND REFERENCE # INTO PACKET
2707 040016 112765 000002 177776      MOVB   #2,CRD(R5)     ;PUT THE CREDIT LIMIT INTO THE PACKET

```

GLOBAL SUBROUTINES SECTION

2708	040024	032764	000010	000014		BIT	#DUPFLG,LUNFLG(R4)	;IS IT A DUP COMMAND?
2709	040032	001404				BEG	1\$;NO, CONNECTION TYFF = 1
2710	040034	112765	000002	177777		MOVB	#2,CONID(R5)	;YES,CONNECTION TYPE = 2
2711	040042	000403				BR	2\$;CONTINUE
2712	040044	112765	000001	177777	1\$:	MOVB	#1,CONID(R5)	;PUT THE CONNECTION TYPE INTO THE PACKET
2713	040052	010511			2\$:	MOV	R5,(R1)	;PUT THE PACKET ADDRESS INTO THE DESCRIPTOR
2714	040054	012761	100000	000002		MOV	#OWN,HIADDR(R1)	;SET THE OWNERSHIP BIT OF THE DESCRIPTOR
2715	040062	005774	000000			TST	@TKIP(R4)	;READ THE IP REGISTER
2716	040066	005774	000002			TST	@TKSA(R4)	;READ THE SA REGISTER
2717	040072	100013				BPL	10\$;BRANCH IF NO ERRORS
2718	040074	017464	000002	000012		MOV	@TKSA(R4),TKSASV(R4)	;SAVE THE SA FOR THE ERROR PRINTOUT
2719	040102					ERRDF	53,WRER4,WRPTE	;PRINT PORT DETECTED ERROR
	040102	104455				TRAP	C\$ERDF	
	040104	000065				.WORD	53	
	040106	032456				.WORD	WRER4	
	040110	033262				.WORD	WRPTE	
2720	040112					DODU	LOGUNT	;DROP THE UNIT
	040112	013700	002312			MOV	LOGUNT,R0	
	040116	104451				TRAP	C\$DODU	
2721	040120	000411				BR	100\$;GET OUT
2722	040122	062701	000004		10\$:	ADD	#RINGSTP,R1	;ADJUST RESPONCE POINTER FOR NEXT TIME
2723	040126	022701	003026			CMP	#DSCEND,R1	;ARE WE AT THE END ?
2724	040132	001002				BNE	15\$;NO, GET OUT
2725	040134	012701	003016			MOV	#CMDRNG,R1	;SET R1 TO TOP BUFFER
2726	040140	010137	023040		15\$:	MOV	R1,CMDSAV	;SAVE THE COMMAND RING LOCATION
2727	040144	000207			100\$:	RTS	PC	;RETURN

L6

GLOBAL AREAS MACRO V05.05 Wednesday 27-Feb-91 11:55 Page 41
GLOBAL SUBROUTINES SECTION

SEQ 0076

```

2729 040146          CDRECV:
2730 040146 004737 040254 1$: JSR    PC,PDRECV      ;CALL PORT DRIVER RECEIVE
2731 040152 032764 000001 000014 BIT    #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
2732 040160 001034          BNE    100$           ;GET OUT IF NOT AVAILABLE
2733 040162 011103          MOV    (R1),R3         ;SET UP RESPONSE BUFFER POINTER
2734 040164 026365 000000 000000 CMP    P.CRF(R3),P.CRF(R5) ;IS THIS THE COMMAND THAT IS EXPECTED ?
2735 040172 001015          BNE    10$           ;GET OUT IF WRONG RESPONSE
2736 040174 012761 100000 000002 MOV    #OWN,HIADDR(R1)  ;GIVE THE CONTROLLER THE RING BACK
2737 040202 062701 000004          ADD    #RNGSTP,R1     ;ADJUST RESPONSE POINTER FOR NEXT TIME
2738 040206 022701 003016          CMP    #CMDRNG,R1     ;ARE WE AT THE END ?
2739 040212 001002          BNE    15$           ;NO, GET OUT
2740 040214 012701 003006          MOV    #RSPRNG,R1     ;SET R1 TO TOP BUFFER
2741 040220 010137 023042 15$: MOV    R1,RSPSAV      ;SAVE THE POINTER FOR NEXT TIME
2742 040224 000412          BR     100$          ;
2743 040226 012737 032700 002310 10$: MOV    #CTRL,FRUIS    ;PRINT CONTROLLER ERROR
2744 040234          ERRDF 56.,WRERS,WRSEGE ;PRINT COMMAND OUT OF SEQUENCE ERROR
      040234 104455          TRAP  C$ERDF
      040236 000070          .WORD 56
      040240 032502          .WORD WRERS
      040242 033450          .WORD WRSEGE
2745 040244          DODU  LOGUNT          ;DROP THE UNIT
      040244 013700 002312          MOV    LOGUNT,R0
      040250 104451          TRAP  C$DODU
2746 040252 000207 100$: RTS    PC          ;RETURN
2747
2748
2749
2750 040254          PDRECV:
2751 040254 013701 023042 003026 1$: MOV    RSPSAV,R1     ;PUT THE RESPONSE RING SAVE IN R1
2752 040260 012737 000225          MOV    #225,CNTHI    ;SET UP THE TIME OUT COUNTER
2753 040266 005002          CLR    R2           ;CLEAR R2
2754 040270 005202          INC    R2           ;INCREMENT HI TIME OUT VALUE ?
2755 040272 001013          BNE    10$           ;IF NOT, BRANCH
2756 040274 005337 003026          DEC    CNTHI        ;ELSE, INCREMENT HI TIMEOUT
2757 040300 001010          BNE    10$           ;KEEP GOING ,NO TIME OUT YET
2758 040302          ERRDF 54.,WRER6,WRTOE ;PRINT COMMAND TIMEOUT MESSAGE
      040302 104455          TRAP  C$ERDF
      040304 000066          .WORD 54
      040306 032533          .WORD WRER6
      040310 033526          .WORD WRTOE
2759 040312          DODU  LOGUNT          ;GO DROP THE UNIT
      040312 013700 002312          MOV    LOGUNT,R0
      040316 104451          TRAP  C$DODU
2760 040320 000425          BR     100$          ;GET OUT ON ERROR
2761 040322 017464 000002 000012 10$: MOV    @1<SA(R4),TKSASV(R4) ;GET SA CONTENTS
2762 040330 005764 000012          TST    TKSASV(R4)   ;CHECK FOR SA ERROR
2763 040334 100013          BPL    20$           ;BRANCH IF NO ERROR
2764 040336 012737 032700 002310 MOV    #CTRL,FRUIS    ;LOAD FAILING FRU
2765 040344          ERRDF 55.,EMSG9,WRPRT ;PRINT SA CONTENTS IN ERROR MESSAGE
      040344 104455          TRAP  C$ERDF
      040346 000067          .WORD 55
      040350 030473          .WORD EMSG9
      040352 033262          .WORD WRPRT
2766 040354          DODU  LOGUNT          ;GO DROP THE UNIT
      040354 013700 002312          MOV    LOGUNT,R0
      040360 104451          TRAP  C$DODU
2767 040362 000404          BR     100$          ;GET OUT ON ERROR

```

M6

GLOBAL AREAS MACRO V05.05 Wednesday 27 Feb 91 11:55 Page 41-1
GLOBAL SUBROUTINES SECTION

SEQ 0077

```

2768 040364 032761 100000 000002 20$: BIT #OWN,HIADDR(R1) ;IS THE SLOT SET TO US ?
2769 040372 001336 BNE 5$ ;KEEP GOING TILL TIMEOUT OR SUCCESS
2770 040374 000207 100$: RTS PC ;RETURN
2771
2772
2773
2774
2775 040376 L2DATA::
2776 040376 010146 MOV R1,-(SP) ;SAVE REGISTERS
2777 040400 010246 MOV R2,-(SP)
2778 040402 010346 MOV R3,-(SP)
2779 040404 012701 060000 MOV #RDBUF,R1 ;GET START ADDRESS OF RECEIVE DATA
2780 040410 121127 000001 CMPB (R1),#1 ;ANY NEW INFORMATION ?
2781 040414 001162 BNE EXIT ;NO, RETURN
2782 040416 026127 000002 000000 CMP L2STA(R1),#0 ;YES, LOCAL PROGRAM FINISHED ?
2783 040424 001552 BEQ GOABO ;YES, SET ABORT FLAG
2784 040426 016103 000002 MOV L2STA(R1),R3 ;GET LEVEL 2 PROGRAM STATUS
2785 040432 042703 177740 BIC #177740,R3 ;MASK OFF UPPER 11 BITS OF STATUS
2786 040436 022703 000037 CMP #37,R3 ;L2 MESSAGE IF 1ST 5 BITS SET
2787 040442 001113 BNE MSCPER ;TMSCP COMMAND ERROR OTHERWISE
2788 040444 016103 000002 MOV L2STA(R1),R3 ;GET LEVEL 2 PROGRAM STATUS
2789 040450 012737 032700 002310 MOV #CTRL,FRUIS ;NO, SET FAILING FRU TO CONTROLLER
2790 040456 042703 000037 BIC #000037,R3 ;MASK OFF UNUSED BITS OF STATUS
2791 040462 006203 ASR R3
2792 040464 006203 ASR R3
2793 040466 006203 ASR R3
2794 040470 006203 ASR R3
2795 040472 006203 ASR R3
2796
2797 040474 022703 000001 CMP #1.,R3 ;IS IT A DATA COMPARE ERROR ?
2798 040500 001006 BNE 5$ ;NO TRY AGAIN
2799 040502 012702 023314 MOV #DCERR,R2 ;SET UP TO PRINT ERROR
2800 040506 012737 032715 002310 MOV #DRVE,FRUIS ;SET FRU CALLOUT TO DRIVE
2801 040514 000454 BR 40$ ;GO PRINT IT
2802
2803 040516 022703 000002 5$: CMP #2.,R3 ;IS IT A CONTROLLER ERROR ?
2804 040522 001003 BNE 10$ ;NO TRY AGAIN
2805 040524 012702 023324 MOV #CNTERR,R2 ;SET UP TO PRINT ERROR
2806 040530 000446 BR 40$ ;GO PRINT IT
2807
2808 040532 022703 000012 10$: CMP #10.,R3 ;IS IT AN INVAILID STATUS ERROR ?
2809 040536 001003 BNE 15$ ;NO TRY AGAIN
2810 040540 012702 023334 MOV #INVSTA,R2 ;SET UP TO PRINT ERROR
2811 040544 000440 BR 40$ ;GO PRINT IT
2812
2813 040546 022703 000024 15$: CMP #20.,R3 ;IS IT A BAD PATTERN NUMBER ERROR ?
2814 040552 001003 BNE 20$ ;NO TRY AGAIN
2815 040554 012702 023344 MOV #BPNERR,R2 ;SET UP TO PRINT ERROR
2816 040560 000432 BR 40$ ;GO PRINT IT
2817
2818 040562 022703 000025 20$: CMP #21.,R3 ;IS IT A RESPONSE ADDRESS ERROR ?
2819 040566 001003 BNE 25$ ;NO TRY AGAIN
2820 040570 012702 023354 MOV #RSPADD,R2 ;SET UP TO PRINT ERROR
2821 040574 000424 BR 40$ ;GO PRINT IT
2822
2823 040576 022703 000026 25$: CMP #22.,R3 ;IS IT A HOST BUFFER ADDRESS ERROR ?
2824 040602 001003 BNE 30$ ;NO TRY AGAIN

```

N6

GLOBAL AREAS MACRO V05.05 Wednesday 27-Feb-91 11:55 Page 41-2

SEQ 0078

GLOBAL SUBROUTINES SECTION

```

2825 040604 012702 023364      MOV    #HBFADD,R2      ;SET UP TO PRINT ERROR
2826 040610 000416      BR     40$            ;GO PRINT IT
2827
2828 040612 022703 000027      30$:  CMP    #23.,R3      ;IS IT A UNKNOWN ERROR LOG RECIEVED ?
2829 040616 001003      BNE    35$            ;NO TRY AGAIN
2830 040620 012702 023374      MOV    #UNERLG,R2     ;SET UP TO PRINT ERROR
2831 040624 000410      BR     40$            ;GO PRINT IT
2832
2833 040626 022703 000030      35$:  CMP    #24.,R3      ;IS IT A RESPONSE TIME OUT ERROR ?
2834 040632 001003      BNE    36$            ;NO TRY AGAIN
2835 040634 012702 023404      MOV    #RSPT0,R2     ;SET UP TO PRINT ERROR
2836 040640 000402      BR     40$            ;GO PRINT IT
2837
2838 040642 012702 023334      36$:  MOV    #INVSTA,R2   ;SET UP TO PRINT ERROR
2839
2840 040646 012237 023054      40$:  MOV    (R2)+,ERRTYP   ;LOAD ERROR TYPE
2841 040652 012237 023056      MOV    (R2)+,ERRNBR   ;LOAD ERROR NUMBER
2842 040656 012237 023060      MOV    (R2)+,ERRMSG   ;LOAD ERROR MESSAGE ADDRESS
2843 040662 012237 023062      MOV    (R2)+,ERRBLK   ;LOAD ERROR SUBROUTINE ADDRESS
2844 040666      ERROR                ;CALL "ERROR" MACRO
      040666 104460      TRAP   C$ERROR
2845 040670 000430      BR     GOABO         ;ABORT MICRODIAGNOSTIC
2846 040672 022703 000012      MSCPER: CMP    #12,R3   ;IS IT A CONTROLLER ERROR ?
2847 040676 001406      BEQ    10$           ;YES, SET FAILING FRU TO CONTROLLER
2848 040700 022703 000014      CMP    #14,R3       ;IS IT A FORMATTER ERROR ?
2849 040704 001403      BEQ    10$           ;YES, LEAVE FAILING FRU AS CONTRLLOER
2850 040706 012737 032715 002310 5$:  MOV    #DRVE,FRUIS   ;NO, SET FAILING FRU TO DRIVE
2851 040714 005303      10$:  DEC    R3
2852 040716 070327 000010      MUL    #10,R3        ;R3 IS OFFSET INTO TMSCP ERROR TABLE
2853 040722 012702 023064      MOV    #L2ETBL,R2   ;GET TMSCP ERROR TABLE START ADDRESS
2854 040726 060302      ADD    R3,R2         ;R2 POINTS TO ENTRY IN TMSCP ERROR TABLE.
2855 040730 012237 023054      MOV    (R2)+,ERRTYP   ;LOAD ERROR TYPE
2856 040734 012237 023056      MOV    (R2)+,ERRNBR   ;LOAD ERROR NUMBER
2857 040740 012237 023060      MOV    (R2)+,ERRMSG   ;LOAD ERROR MESSAGE ADDRESS
2858 040744 012237 023062      MOV    (R2)+,ERRBLK   ;LOAD ERROR SUBROUTINE ADDRESS
2859 040750      ERROR                ;CALL "ERROR" MACRO
      040750 104460      TRAP   C$ERROR
2860 040752 052764 000020 000014 GOABO: BIS    #ABTFLG,LUNFLG(R4) ;SET ABORT FLAG
2861 040760      DORPT                ;DUMP NORMAL COMPLETION STATISTICS
      040760 104424      TRAP   C$DRPT
2862 040762 012603      EXIT:  MOV    (SP)+,R3   ;RESTORE REGISTERS
2863 040764 012602      MOV    (SP)+,R2
2864 040766 012601      MOV    (SP)+,R1
2865 040770 000207      RTS    PC
2866
2867 040772      ENDMOD
2878      .TITLE MISCELLANEOUS SECTIONS
2879      .SBTTL REPORT CODING SECTION
2907
2908 040772      BGNMOD
2909

```

```

2911 .SBTTL INITIALIZE SECTION
2912
2913
2914 ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
2915 ; AT THE BEGINNING OF EACH PASS.
2916 ;--
2917
2918 040772 BGNINIT
      040772 L$INIT::
2919
2920
2921 040772 READDEF #EF.START ;IF THIS IS A FRESH START
      040772 012700 000040 MOV #EF.START,R0
      040776 104447 TRAP C$REFG
2922 041000 BCOMPLETE START ; THEN GO TO START
      041000 103421 BCS START
2923
2924 041002 READDEF #EF.RESTART ;IF THIS IS A RESTART
      041002 012700 000037 MOV #EF.RESTART,R0
      041006 104447 TRAP C$REFG
2925 041010 BCOMPLETE START ; THEN GO TO START
      041010 103415 BCS START
2926
2927 041012 READDEF #EF.PWR ;IF POWER-FAIL OCCURRED
      041012 012700 000034 MOV #EF.PWR,R0
      041016 104447 TRAP C$REFG
2928 041020 BCOMPLETE START ; THEN START FROM THE BEGINNING
      041020 103411 BCS START
2929
2930 041022 READDEF #EF.NEW ;IF THIS IS A NEW PASS
      041022 012700 000035 MOV #EF.NEW,R0
      041026 104447 TRAP C$REFG
2931 041030 BCOMPLETE NUPASS ; THEN SKIP START UP CODE
      041030 103422 BCS NUPASS
2932
2933 041032 READDEF #EF.CONTINUE ;IF THIS IS A CONTINUE
      041032 012700 000036 MOV #EF.CONTINUE,R0
      041036 104447 TRAP C$REFG
2934 041040 BCOMPLETE END ; THEN SKIP ALL INIT CODE
      041040 103465 BCS END
2935
2936 041042 BR NEXT ;JUST HERE FOR NEXT UUT
2937
2938 041044 START:
2939 041044 012737 000000 002270 MOV #0,PASCNT ;INITIALIZE PASS COUNT
2940 041052 005037 002274 CLR KTFLAG ;IN CASE WE'RE STARTED > THAN ONCE
2941 041056 012704 002212 MOV #LUNBLK,R4 ;R4 WILL ALWAYS POINT TO LUNBLK
2942 041062 022737 001400 002120 CMP #1400,L$HIME ;IF <= 28KWORDS OF MEMORY PRESENT
2943 041070 103002 BHS NUPASS ; THEN SKIP NEXT
2944 041072 004737 036142 JSR PC,KTTEST ; ELSE SEE IF MMU IS PRESENT
2945
2946 041076 NUPASS: BRESET ;CLEAR THE WORLD
      041076 104433 TRAP C$RESET
2947 041100 005237 002270 INC PASCNT ;UPDATE THE PASS COUNT
2948 041104 012737 177777 002312 MOV #-1,LOGUNT ;INITIALIZE LOGICAL UNIT COUNT
2949
2950 041112 005237 002312 NEXT: INC LOGUNT ;POINT TO NEXT UUT

```


C7

INITIALIZE SECTION

```

2951 041116 023737 002312 002012      CMP      LOGUNT,L$UNIT      ;IF WE'VE PASSED MAXIMUM UUT'S
2952 041124 001433                      BEQ      END                ; THEN LEAVE INIT
2953
2954 041126                      GPHARD  LOGUNT,R0          ;GET P-TABLE FOR THIS UNIT
      041126 013700 002312      MOV      LOGUNT,R0
      041132 104442      TRAP    C$GPHRD
2955 041134                      BNCOMPLETE NEXT          ;TRY AGAIN
      041134 103366      BCC     NEXT
2956
2957 041136 011064 000000      MOV      (R0),TKIP(R4)      ;PUT IP REG ADDRESS IN LUNBLK
2958 041142 012064 000002      MOV      (R0)+,TKSA(R4)    ; AND ANOTHER COPY IN LUNBLK
2959 041146 062764 000002 000002    ADD      #2,TKSA(R4)        ;MAKE IT THE SA REG ADDRESS
2960 041154 012064 000004      MOV      (R0)+,TKVEC(R4)   ;GET THE VECTOR INTO THE LUNBLK
2961 041160 011064 000006      MOV      (R0),MSCPUN(R4)   ;PUT THE T/MSCP UNIT # IN LUNBLK
2962 041164 004737 036326      JSR     PC,RSTVEC          ;SET UUT VECTOR FOR ILLEGAL INTRPTS.
2963 041170                      PRINTF  #IMSG,LOGUNT      ;"TESTING UNIT N"
      041170 013746 002312      MOV      LOGUNT,-(SP)
      041174 012746 041220      MOV      #IMSG,-(SP)
      041200 012746 000002      MOV      #2,-(SP)
      041204 010600      MOV      SP,R0
      041206 104417      TRAP    C$PNTF
      041210 062706 000006      ADD     #6,SP
2964
2965 041214                      END:
2966 041214                      EXIT   INIT
      041214 104432      TRAP    C$EXIT
      041216 000032      .WORD  L10011-.
2967
2979 041220      045      116      045  IMMSG:  .ASCIZ  ?N#ATESTING UNIT #D1#N?
2980                      .EVEN
2981
2982 041250                      ENDINIT
      041250                      L10011:
      041250 104411      TRAP    C$INIT

```

D7

CLEANUP CODING SECTION

```

2984 .SBTTL CLEANUP CODING SECTION
2985
2986 ;**
2987 ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
2988 ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
2989 ;--
2990
2991 041252 BGNCLN
041252 L$CLEAN:
2992
2999 041252 032764 000100 002212 BIT #T9FLAG,LUNBLK(R4) ;IF NOT HERE FROM TEST 9
3000 041260 001400 BEQ ENDCLE ; THEN SKIP THE REST
3001
3002 ;EVENTUALLY MORE CODE WILL BE PLACED HERE TO GUARANTEE THAT AN ABORT
3003 ;COMMAND IS ISSUED TO THE UUT TO STOP EXECUTION OF THE LOCAL PROGRAM.
3004
3005 041262 005064 000014 ENDCLE: CLR LUNFLG(R4) ;CLEAR OUT THE LUN FLAGS
3006
3007 ;NOTE: THIS LINE OF CODE MAY HAVE TO BE REMOVED TO HANDLE tC FOLLOWED
3008 ;BY A PROCEED COMMAND CORRECTLY.
3009 041266 CLRVEC TKVEC(R4) ;PUT "TRAP CATCHER" INTO VECTOR
041266 016400 000004 MOV TKVEC(R4),R0
041272 104436 TRAP C$CVEC
3010
3011 041274 EXIT CLN
041274 104432 TRAP C$EXIT
041276 000002 .WORD L10012-.
3012
3024
3025 .EVEN
3026
3027 041300 ENDCLN
041300 L10012:
041300 104412 TRAP C$CLEAN

```

E7

DROP UNIT SECTION

```

3029          .SBTTL  DROP UNIT SECTION
3030
3031          ;**
3032          ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
3033          ; TO NO LONGER BE TESTED.
3034          ;--
3035
3036 041302          BGNDU
041302          L$DU::
3037
3043
3044 041302 012764 000001 000014          MOV      #DRPFLG,LUNFLG(R4)          ;LETS PROGRAM KNOW IT'S DEAD
3045
3046 041310          EXIT      DU
041310 000167          .WORD   J$JMP
041312 000000          .WORD   L10013-2-.
3047
3059
3060          .EVEN
3061
3062 041314          ENDDU
041314          L10013:
041314 104453          TRAP    C$DU

```

F7

ADD UNIT SECTION

```

3064          .SBTTL  ADD UNIT SECTION
3065
3066          ;**
3067          ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
3068          ; TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
3069          ; TO THE TEST CYCLE.
3070          ;--
3071
3072 041316          BGN AU
041316          L$AU::
3073
3079
3080 041316          EXIT      AU
041316 000167        .WORD    J$JMP
041320 000000        .WORD    L10014-2-.
3081
3093
3094          .EVEN
3095
3096 041322          ENDAU
041322          L10014:
041322 104452        TRAP     C$AU
3097
3098 041324          BGN AUTO
041324          L$AUTO::
3099 041324          ENDAUTO
041324          L10015:
041324 104461        TRAP     C$AUTO
3100
3101 041326          ENDMOD
3102

```

G7

```

3105
3106      .TITLE HARDWARE TEST
3110      000000      HELP=0      ; CONTROL LISTING OF HELP INFORMATION
3111      ; HELP=0      NO LIST
3112      ; HELP=1      LIST
3113
3114      ;ONEFILE=      ; CONTROL USE OF SOURCE FILES
3115      ; ONEFILE IS NOT DEFINED      ASSEMBLE EACH SOURCE FILE SEPARATELY
3116      ; ONEFILE=ANYTHING      ASSEMBLE ALL SOURCE FILES TOGETHER
3120
3121      .SBTTL TEST 1: EXISTENCE VERIFICATION TEST
3134
3136      ;*****
3137      ;*****
3138      ;
3139      ;
3140      ;TEST 1 - EXISTENCE VERIFICATION TEST
3141      ; THIS TEST VERIFIES THE EXISTENCE OF THE UUT BY
3142      ; ATTEMPTING TO READ FIRST THE IP AND THEN THE SA
3143      ; REGISTERS OF THE TKxx. VECTOR 4 IS SET UP WITH
3144      ; A TRAP HANDLING ROUTINE IN CASE OF A NON-EXISTENT
3145      ; MEMORY TIMEOUT.
3146      ;
3147      ;*****
3148      ;*****
3149
3153      041326      BGNTST
3154      041326      T1::      NOP
3155      041326      000240      MOV      #1,ITRCNT      ;SET UP FOR ONE TEST ITERATION
3156      041330      012737      000001      002272      TST      PASCNT      ;IF PASS 0
3157      041336      005737      002270      BEQ      T1.1      ; THEN START TEST
3158      041342      001404      MOV      #10,ITRCNT      ; ELSE DO MULTIPLE ITERATIONS
3159      041344      012737      000010      002272      NOP
3160      041352      000240      BGNSUB
3161      041354      T1.1:      TRAP      C$BSUB
3162      041354      104402      1$:      CLR      TRP4FG      ;CLEAR NXM TRAP FLAG
3163      041356      005037      002276      SETVEC   #VEC4,#TRAP4,#PRI07      ;SET UP VECTOR 4 FOR NXM TRAP
3164      041362      041362      012746      000340      MOV      #PRI07,-(SP)
3165      041366      012746      036114      MOV      #TRAP4,(SP)
3166      041372      012746      000004      MOV      #VEC4,-(SP)
3167      041376      012746      000003      MOV      #3,-(SP)
3168      041402      104437      TRAP      C$SVEC
3169      041404      062706      000010      ADD      #10,SP
3170      041410      000240      NOP
3171      041412      005074      000000      CLR      @TKIP(R4)      ;WRITE THE IP REGISTER
3172      041416      000240      NOP
3173      041420      DELAY      1      ;MAKE SURE TIMEOUT CAN OCCUR
3174      041420      012727      000001      MOV      #1,(PC)+
3175      041424      000000      .WORD      0
3176      041426      013727      002116      MOV      L$DLY,(PC)+
3177      041432      000000      .WORD      0
3178      041434      005367      177772      DEC      -6(PC)
3179      041440      001375      BNE      . 4
3180      041442      005367      177756      DEC      -22(PC)

```

H7

HARDWARE TEST MACRO V05.05 Wednesday 27 Feb-91 11:55 Page 46-1
 TEST 1: EXISTENCE VERIFICATION TEST

SEQ 0085

```

041446 001367          BNE      .-20
3168
3169 041450 005737 002276    TST      TRP4FG          ;IF NO TRAP OCCURRED
3170 041454 001416          BEQ      5$              ; THEN CONTINUE TEST
3171 041456 000240          NOP
3172 041460 012737 032700 002310  MOV     #CTRL,FRUIS     ;IDENTIFY FAILING FRU FOR PRINTOUT
3173 041466          ERRDF   5,EMSG5,PRIERR ;"NXM ON READ TKIP"
      041466 104455          TRAP    C$ERDF
      041470 000005          .WORD  5
      041472 030342          .WORD  EMSG5
      041474 033206          .WORD  PRIERR
3174 041476          CKLOOP
      041476 104406          TRAP    C$CLP1          ;LOOP ON ERROR?
3175 041500          DODU    LOGUNT          ;DROP UNIT
      041500 013700 002312    MOV     LOGUNT,RO
      041504 104451          TRAP    C$DODU
3176 041506          ESCAPE  SUB              ;CAN'T CONTINUE
      041506 104410          TRAP    C$ESCAPE
      041510 000002          .WORD  L10017-.
3177
3178 041512          5$:      ENDSUB
      041512          L10017:
      041512 104403          TRAP    C$ESUB
3179 041514 000240          NOP
3180 041516          CLRVEC #VEC4              ;RESTORE VECTOR 4
      041516 012700 000004    MOV     #VEC4,RO
      041522 104436          TRAP    C$CVEC
3181 041524 032764 000001 000014  BIT     #DRPFLG,LUNFLG(R4) ;IF UNIT WAS NOT DROPPED
3182 041532 001402          BEQ     T1.2            ; THEN CONTINUE TESTING
3183 041534          ESCAPE  TST              ; ELSE LEAVE TEST
      041534 104410          TRAP    C$ESCAPE
      041536 000264          .WORD  L10016-.
3184
3185 041540          1.2:    BGNSUB
      041540          T1.2:
      041540 104402          TRAP    C$BSUB
3186 041542 005037 002276    10$:    CLR      TRP4FG          ;CLEAR NXM ERROR FLAG
3187
3188 041546          SETVEC #VEC4,#TRAP4,#PRI07 ;SET VECTOR 4 FOR NXM TRAPS
      041546 012746 000340    MOV     #PRI07,-(SP)
      041552 012746 036114    MOV     #TRAP4,-(SP)
      041556 012746 000004    MOV     #VEC4,-(SP)
      041562 012746 000003    MOV     #3,-(SP)
      041566 104437          TRAP    C$SVEC
      041570 062706 000010    ADD     #10,SP
3189 041574 000240          NOP
3190 041576 005774 000002    TST     @TKSA(R4)        ;READ THE SA REGISTER
3191 041602 000240          NOP
3192 041604          DELAY  25.              ;WAIT TO ALLOW NXM TRAP
      041604 012727 000031    MOV     #25.,(PC)+
      041610 000000          .WORD  0
      041612 013727 002116    MOV     L$DLY,(PC)+
      041616 000000          .WORD  0
      041620 005367 177772    DEC     -6(PC)
      041624 001375          BNE     . 4
      041626 005367 177756    DEC     22(PC)
      041632 001367          BNE     . 20

```

```

3193
3194 041634 005737 002276      TST      TRP4FG      ;IF NXM DID NOT OCCUR
3195 041640 001416              BEQ      15$        ; THEN CONTINUE TEST
3196 041642 000240              NOP
3197 041644 012737 032700 002310  MOV      #CTRL,FRUIS ;IDENTIFY FAILING FRU FOR PRINTOUT
3198 041652              ERRDF      7,EMSG7,PRIERR ;"NXM ON FIRST READ OF SA"
      041652 104455              TRAP      C$ERDF
      041654 000007              .WORD    7
      041656 030414              .WORD    EMSG7
      041660 033206              .WORD    PRIERR
3199 041662              CKLOOP                ;LOOP ON ERROR?
      041662 104406              TRAP      C$CLP1
3200 041664              DODU      LOGUNT      ;DROP UNIT IF NOT
      041664 013700 002312      MOV      LOGUNT,RO
      041670 104451              TRAP      C$DODU
3201 041672              ESCAPE    SUB          ;LEAVE TEST
      041672 104410              TRAP      C$ESCAPE
      041674 000062              .WORD    L10020-.
3202
3203 041676 017464 000002 000012 15$:  MOV      @TKSA(R4),TKSASV(R4) ;GET A COPY OF SA IN MEMORY
3204 041704 032764 004000 000012      BIT      #B.S1,TKSASV(R4) ;IF STEP 1 BIT IS SET
3205 041712 001021              BNE      16$        ; THEN TEST 1 IS COMPLETE
3206 041714 000240              NOP
3207 041716 012737 004000 002314      MOV      #B.S1,SAEXP ;LOAD "EXPECTED FOR PRINTOUT
3208 041724 012737 032700 002310      MOV      #CTRL,FRUIS ;IDENTIFY FAILING FRU FOR PRINTOUT
3209 041732              ERRDF      8.,EMSG8,PRISA ;"SA REG IN ERROR ON FIRST READ"
      041732 104455              TRAP      C$ERDF
      041734 000010              .WORD    8
      041736 030435              .WORD    EMSG8
      041740 033024              .WORD    PRISA
3210 041742              CKLOOP                ;LOOP ON ERROR?
      041742 104406              TRAP      C$CLP1
3211 041744              DODU      LOGUNT      ;DROP UNIT IF NOT
      041744 013700 002312      MOV      LOGUNT,RO
      041750 104451              TRAP      C$DODU
3212 041752              ESCAPE    SUB          ;LEAVE TEST
      041752 104410              TRAP      C$ESCAPE
      041754 000002              .WORD    L10020 .
3213 041756              ENDSUB
      041756              L10020:
      041756 104403              TRAP      C$ESUB
3214
3215 041760 005037 002314      20$:  CLR      SAEXP      ;CLEAR ERROR INDICATOR
3216 041764              CLRVEC    #VEC4      ;RESTORE VECTOR 4
      041764 012700 000004      MOV      #VEC4,RO
      041770 104436              TRAP      C$CVEC
3217 041772 032764 000001 000014      BIT      #DRPFLG,LUNFLG(R4) ;IF UNIT DROPPED
3218 042000 001006              BNE      25$        ; THEN LEAVE NOW
3219 042002 005337 002272      DEC      ITRCNT    ;IF ITERATIONS EQUAL 0
3220 042006 000240              NOP
3221 042010 001402              BEQ      25$        ; THEN LEAVE TEST
3222 042012 000137 041354      JMP      T1.1      ; ELSE GO BACK FOR MORE
3223
3224 042016              25$:  EXIT      TST
      042016 104432              TRAP      C$EXIT
      042020 000002              .WORD    L10016-.
3225

```

J7

TEST 1: EXISTENCE VERIFICATION TEST

```
3226  
3227  
3228  
3229 042022 .EVEN  
      042022 ENDTST  
      042022 104401 L10016: TRAP C$ETST
```


K7

3232
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278

042024
042024
042024
042032
042034
042034
042036
042042
042050
042052
042052
042056
042062
042066
042070
042072
042076
042076
042100
042102
042110
042116
042124
042126
042126
042134
042142
042150
042150
042154
042160
042162
042170
042170
042172
042174
042176
042200
042200

032764
001423
104421
010037
032737
001412
013746
012746
012746
010600
104417
062706
104432
000504
012737
012737
022737
001403
012737
140000
013737
004737
005737
001415
012737
042170
030473
033000
104406

.SBTTL TEST 2: SA REGISTER WRAP TEST

```
*****  
*****  
:TEST 2 - SA REGISTER WRAP TEST  
:THIS TEST WILL INITIALIZE THE UUT BY WRITING TO ITS  
:IP REGISTER. IT WILL FORCE THE UUT INTO DIAGNOSTIC  
:WRAP MODE, AND WRITE FIRST A FLOATING 0 DATA PATTERN,  
:FOLLOWED BY A FLOATING 1 DATA PATTERN TO THE SA REG.  
:EACH WRITE WILL BE FOLLOWED BY A READ AND COMPARE  
:OPERATION.  
*****  
*****
```

BGNTST
T2::

```
BIT #DRPFLG,LUNFLG(R4) ;IF UUT NOT DROPPED  
BEQ G02 ; 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 L10021-.  
  
G02: MOV #1,-NISTP ;STEP 1 FOR ERROR PRINTOUT  
MOV #1,IIRCNT ;SET UP FOR ONE TEST ITERATION  
CMP #1,PASCNT ;IF FIRST PASS  
BEQ 2$ ; THEN START TEST  
MOV #2,IIRCNT ; ELSE DO 2 ITERATIONS  
  
2$: MOV #BIT15!B.WR,SAEXP ;SET UP STEP 1 FOR DIAG. WRAP MODE  
MOV SAEXP,STPTBL ;PUT IT IN STEP 1 OF TABLE  
JSR PC,STEP1 ;GO DO IT  
  
TST STEPST ;IF STATUS OKAY  
BEQ 5$ ; THEN CONTINUE TEST  
  
MOV #CTRL,FRUIS ;FAILING FRU FOR PRINTOUT  
ERRDF 9,MSG9,PRIINI ;"SA CONTENTS IN ERROR"  
TRAP C$ERDF  
.WORD 9  
.WORD MSG9  
.WORD PRIINI  
CKLOOP ;LOOP ON ERROR?  
TRAP C$CLP1
```

L7

HARDWARE TEST MACRO V05.05 Wednesday 27-Feb-91 11:55 Page 47-1
 TEST 2: SA REGISTER WRAP TEST

SEQ 0089

```

3279 042202          DODU LOGUNT          ;DROP UUT
      042202 013700 002312 MOV LOGUNT,RO
      042206 104451 TRAP C$DODU
3280 042210          ESCAPE TST          ;LEAVE TST
      042210 104410 TRAP C$ESCAPE
      042212 000372 .WORD L10021-.
3281
3282 042214 012737 000100 002326 5$: MOV #100, OUTER ;SET UP FOR DELAY ROUTINE
3283 042222 012737 037200 002324 6$: MOV #16000., INNER ;SET UP INNER
3284 042230 017464 000002 000012 MOV @TKSA(R4), TKSASV(R4) ;GET SA CONTENTS
3285 042236 023764 002314 000012 CMP SAEXP, TKSASV(R4) ;IF SA IS WHAT WE EXPECT
3286 042244 001422 BEQ 10$ ; THEN MOVE ALONG
3287 042246 004737 036446 JSR PC, PDELAY ; ELSE GIVE UUT SOME TIME
3288 042252 005737 002330 TST TOUT ;IF NO TIMEOUT YET
3289 042256 001761 BEQ 6$ ; THEN GO TAKE ANOTHER LOOK
3290
3291 042260 012737 032700 002310 MOV #CTRL, FRUIS ;FAILING FRU FOR PRINTOUT
3292 042266 ERRDF 10., MSG9, PRIINI ;"SA CONTENTS IN ERROR"
      042266 104455 TRAP C$ERDF
      042270 000012 .WORD 10
      042272 030473 .WORD MSG9
      042274 033000 .WORD PRIINI
3293 042276 CKLOOP
      042276 104406 TRAP C$CLP1
3294 042300 DODU LOGUNT
      042300 013700 002312 MOV LOGUNT, RO
      042304 104451 TRAP C$DODU
3295 042306 ESCAPE TST
      042306 104410 TRAP C$ESCAPE
      042310 000274 .WORD L10021-.
3296
3297 042312 012737 177776 002322 10$: MOV #177776, WRDATA ;INITIALIZE WRAP DATA
3298 042320 013774 002322 000002 11$: MOV WRDATA, @TKSA(R4) ;SEND DATA TO UUT
3299 042326 013737 002322 002314 MOV WRDATA, SAEXP ;SAVE A COPY FOR COMPARE
3300 042334 012737 000100 002326 MOV #100, OUTER ;SET UP FOR DELAY ROUTINE
3301
3302 042342 013737 037200 002324 15$: MOV 16000., INNER ;INNER TOO
3303 042350 017464 000002 000012 MOV @TKSA(R4), TKSASV(R4) ;READ SA
3304 042356 023764 002314 000012 CMP SAEXP, TKSASV(R4) ;IF DATA MATCHES
3305 042364 001422 BEQ 20$ ; THEN CHANGE DATA
3306 042366 004737 036446 JSR PC, PDELAY ; ELSE GIVE UUT SOME TIME
3307 042372 005737 002330 TST TOUT ;IF NO TIMEOUT YET
3308 042376 001761 BEQ 15$ ; THEN GO TAKE ANOTHER LOOK
3309
3310 042400 012737 032700 002310 MOV #CTRL, FRUIS ;FAILING FRU FOR PRINTOUT
3311 042406 ERRDF 11., MSG10, PRIINI ;"SA WRONG IN DATA WRAP"
      042406 104455 TRAP C$ERDF
      042410 000013 .WORD 11
      042412 030520 .WORD MSG10
      042414 033000 .WORD PRIINI
3312 042416 CKLOOP
      042416 104406 TRAP C$CLP1
3313 042420 DODU LOGUNT
      042420 013700 002312 MOV LOGUNT, RO
      042424 104451 TRAP C$DODU
3314 042426 ESCAPE TST ;GET OUT IF NOT LOOPING
      042426 104410 TRAP C$ESCAPE

```

M7

HARDWARE TEST MACRO V05.05 Wednesday 27-Feb-91 11:55 Page 47-2

SEQ 0090

TEST 2: SA REGISTER WRAP TEST

```

042430 000154          .WORD  L10021-.
3315
3316 042432 006137 002322 20$:  ROL  WRDATA          ;SHIFT TEST PATTERN
3317 042436 103730          BCS  11$          ;WE'RE NOT DONE YET
3318
3319 042440 012737 000001 002322          MOV  #1,WRDATA      ;SET UP FOR FLOATING 1 PATTERN
3320 042446 013774 002322 000002 24$:  MOV  WRDATA,@TKSA(R4) ;SEND DATA TO UUT
3321 042454 013737 002322 002314          MOV  WRDATA,SAEXP   ;KEEP A COPY FOR COMPARE
3322 042462 012737 000100 002326          MOV  #100,OUTER    ;SET UP FOR DELAY ROUTINE
3323
3324 042470 012737 016000 002324 25$:  MOV  #16000,INNER   ;DELAY ROUTINE TOO
3325 042476 017464 000002 000012          MOV  @TKSA(R4),TKSASV(R4) ;READ THE SA
3326 042504 023764 002314 000012          CMP  SAEXP,TKSASV(R4) ;IF IT MATCHES
3327 042512 001422          BEQ  30$          ; THEN SEE IF WE'RE DONE
3328 042514 004737 036446          JSR  PC,PDELAY     ; ELSE GIVE UUT SOME MORE TIME
3329 042520 005737 002330          TST  TOUT         ;IF NO TIMEOUT YET
3330 042524 001761          BEQ  25$          ; THEN TAKE ANOTHER LOOK
3331
3332 042526 012737 032700 002310          MOV  #CTRL,FRUIS   ;FAILING FRU FOR PRINTOUT
3333 042534          ERRDF 12.,EMSG10,PRIINI ;"SA WRONG IN DATA WRAP"
042534 104455          TRAP C$ERRDF
042536 000014          .WORD 12
042540 030520          .WORD EMSG10
042542 033000          .WORD PRIINI
3334 042544          CKLOOP
042544 104406          TRAP C$CLP1
3335 042546          DODU  LOGUNT
042546 013700 002312          MOV  LOGUNT,RO
042552 104451          TRAP C$DODU
3336 042554          ESCAPE TST          ;LEAVE TEST IF NOT LOOPING
042554 104410          TRAP C$ESCAPE
042556 000026          .WORD  L10021-.
3337
3338 042560 006137 002322 30$:  ROL  WRDATA          ;SHIFT DATA PATTERN
3339 042564 103330          BCC  24$          ;WE'RE NOT DONE YET
3340 042566 005337 002272          DEC  ITRCNT       ;IF ITERATIONS = 0
3341 042572 001402          BEQ  T2EXT       ; THEN LEAVE TEST
3342 042574 000137 042134          JMP  2$          ; ELSE DO ANOTHER ONE
3343
3344          T2EXT:  EXIT  TST          ;GET OUTTA HERE
042600          TRAP C$EXIT
042602 000002          .WORD  L10021-.
3345
3346 042604          ENDTST
042604          L10021:
042604 104401          TRAP  C$ETST

```

N7

```

3349
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3371
3372 042606
      042606
3373
3374 042606 032764 000001 000014
3375 042614 001423
3376 042616
      042616 104421
      042620 010037 002336
3377 042624 032737 001000 002336
3378 042632 001412
3379 042634
      042634 013746 002114
      042640 012746 050026
      042644 012746 000002
      042650 010600
      042652 104417
      042654 062706 000006
3380 042660
      042660 104432
      042662 000410
3381
3382 042664 012737 032700 002310
3383 042672 012737 000001 002272
3384 042700 022737 000001 002270
3385 042706 001403
3386 042710 012737 000012 002272
3387
3388 042716 012705 000000
3389 042722 012737 000001 002316
3390 042730 016437 000004 002250
3391 042736 006237 002250
3392 042742 006237 002250
3393 042746 013737 002250 002264
3394 042754 052737 104400 002250
3395 042762
      042762 104407
3396 042764
      042764 103404
3397 042766 012737 004700 002260
3398 042774 000403

```

.SBTTL TEST 3: INITIALIZATION TEST

```

;*****
;*****
;TEST 3 - INITIALIZATION TEST
; THIS TEST COMMENCES THE UQ-PORT INITIALIZATION SEQUENCE
; WITH INTERRUPTS DISABLED. IT VERIFIES THAT ALL STEP
; TRANSITIONS OCCUR WITHIN THE ALLOTTED TIME, AND THAT ALL
; HOST SUPPLIED INFORMATION IS ECHOED BY THE UUT. THE
; PROGRAM FURTHER VERIFIES THAT NO INTERRUPTS OCCUR AS A
; RESULT OF THE STEP TRANSITIONS.
;*****
;*****

```

```

T3:: BGNTST
      BIT      #DRPFLG,LUNFLG(R4)      ;IF UUT NOT DROPPED
      BEQ      G03                      ; 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    L10022-.

G03:  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$:  MOV      #0,R5                       ;SET UP R5 AS INDEX TO STEP TABLE
      MOV      #1,INISTP                 ;STEP 1 FOR ERROR PRINTOUT
      MOV      TKVEC(R4),STPTBL          ;PUT VECTOR IN STEP 1
      ASR      STPTBL                     ;DIVIDE BY TWO
      ASR      STPTBL                     ;DIVIDE BY FOUR
      MOV      STPTBL,CMPTBL+4           ;PUT VECTOR IN STEP 3 COMPARE
      BIS      #104400,STPTBL           ;REST OF STEP ONE
      READBUS
      TRAP     C$RDBU                    ;IF QBUS
      BCOMPLETE 3$                       ; THEN BR FOR QBUS SET UP
      BCS      3$
      MOV      #4700,CMPTBL             ; ELSE SET UP FOR UNIBUS
      BR      4$

```

```

3399 042776 01273 005700 002260 3$: MOV #B.S1!B.QB!B.DI!B.OO!B.MP,CMP TBL
3400 ;STEP 1 COMPARE VALUE
3401 043004 012737 060050 002252 4$: MOV #COMMAR,STPTBL*2 ;STEP 2 - COMM AREA ADDRESS
3402 043012 012737 010211 002262 MOV #010211,CMP TBL*2 ;STEP 2 COMPARE
3403 043020 012737 000000 002254 MOV #0,STPTBL*4 ;STEP 3 - HIGH ADDRESS
3404 043026 112737 000040 002265 MOV B #40,CMP TBL*5 ;REST OF STEP 3 COMPARE
3405 043034 012737 000000 002256 MOV #0,STPTBL*6 ;STEP 4
3406 043042 012737 040000 002266 MOV #040000,CMP TBL*6 ;STEP 4 COMPARE
3407
3408 043050 004737 036474 JSR PC,STEP1 ;GO DO IT
3409 043054 005737 002320 TST STEPST ;IF STATUS OKAY
3410 043060 001412 BEQ 5$ ; THEN CONTINUE TEST
3411
3412 043062 ERRDF 9.,EMSG9,PRIINI ;"SA CONTENTS IN ERROR"
043062 104455 TRAP C$ERDF
043064 000011 .WORD 9
043066 030473 .WORD EMSG9
043070 033000 .WORD PRIINI
3413 043072 CKLOOP ;LOOP ON ERROR?
043072 104406 TRAP C$CLP1
3414 043074 DODU LOGUNT ;DROP UUT
043074 013700 002312 MOV LOGUNT,R0
043100 104451 TRAP C$DODU
3415 043102 ESCAPE TST ;LEAVE TST
043102 104410 TRAP C$ESCAPE
043104 000166 .WORD L10022-.
3416
3417 043106 005237 002316 5$: INC INISTP ;ADJUST STEP COUNTER
3418 043112 062705 000002 ADD #2,R5 ;ADJUST TABLE INDEX
3419 043116 012737 000100 002326 6$: MOV #100,OUTER ;SET UP FOR DELAY ROUTINE
3420 043124 016537 002260 002314 MOV CMP TBL(R5),SAEXP ;SET UP FOR COMPARE
3421 043132 012737 037200 002324 7$: MOV #16000.,INNER ;SET UP INNER
3422 043140 017464 000002 000012 MOV @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
3423 043146 022705 000006 CMP #6,R5 ;ARE WE IN STEP 4?
3424 043152 001005 BNE 8$ ;BRANCH IF NOT
3425 043154 033764 002314 000012 BIT SAEXP,TKSASV(R4) ;JUST LOOK FOR STEP 4 BIT
3426 043162 001024 BNE 10$ ;IT'S SET SO LET'S GO
3427 043164 000404 BR 9$ ;STAY IN LOOP OTHERWISE
3428 043166 023764 002314 000012 8$: CMP SAEXP,TKSASV(R4) ;IF SA IS WHAT WE EXPECT
3429 043174 001417 BEQ 10$ ; THEN MOVE ALONG
3430 043176 004737 036446 9$: JSR PC,PDELAY ; ELSE GIVE UUT SOME TIME
3431 043202 005737 002330 TST TOUT ;IF NO TIMEOUT YET
3432 043206 001751 BEQ 7$ ; THEN GO TAKE ANOTHER LOOK
3433
3434 043210 ERRDF 13.,EMSG9,PRIINI ;"SA CONTENTS IN ERROR"
043210 104455 TRAP C$ERDF
043212 000015 .WORD 13
043214 030473 .WORD EMSG9
043216 033000 .WORD PRIINI
3435 043220 CKLOOP
043220 104406 TRAP C$CLP1
3436 043222 DODU LOGUNT ;DROP UUT
043222 013700 002312 MOV LOGUNT,R0
043226 104451 TRAP C$DODU
3437 043230 ESCAPE TST
043230 104410 TRAP C$ESCAPE
043232 000040 .WORD L10022-.

```

C8

TEST 3: INITIALIZATION TEST

```

3438
3439 043234 016574 002250 000002 10$: MOV STPTBL(R5),@TKSA(R4) ;WRITE NEXT STEP TO UJT
3440 043242 022705 000006 CMP #6,R5 ;IF NOT IN STEP 4
3441 043246 001317 BNE 5$ ;GO BACK TO MAIN LOOP
3442
3443 043250 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;HAS UJT BEEN DROPPED
3444 043256 001003 BNE T3EXT ;LEAVE NOW IF SO
3445 043260 005337 002272 DEC ITRCNT ;IF MORE ITERATIONS LEFT
3446 043264 001214 BNE 2$ ; THEN GO DO IT AGAIN
3447
3448 043266 T3EXT: EXIT TST
043266 104432 TRAP C$EXIT
043270 000002 .WORD L10022-.
3449
3450 043272 L10022: ENDTST
043272 TRAP C$ETST
043272 104401

```

```

3453
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3471
3472
3473 043274
      043274
3474
3475 043274 032764 000001 000014
3476 043302 001423
3477 043304
      043304 104421
      043306 010037 002336
3478 043312 032737 001000 002336
3479 043320 001412
3480 043322
      043322 013746 002114
      043326 012746 050026
      043332 012746 000002
      043336 010600
      043340 104417
      043342 062706 000006
3481 043346
      043346 104432
      043350 000516
3482
3483 043352 042764 000004 000014
3484 043360 042764 000002 000014
3485 043366 012737 032700 002310
3486 043374 012737 000001 002372
3487 043402 022737 000001 002270
3488 043410 001403
3489 043412 012737 000012 002272
3490
3491 043420 004737 036356
3492 043424 012705 000000
3493 043430 012737 000001 002316
3494 043436 016437 000004 002250
3495 043444 006237 002250
3496 043450 006237 002250
3497 043454 013737 002250 002264
3498 043462 052737 104600 002250
3499 043470
      043470 104407
3500 043472
      043472 103404
3501 043474 012737 004700 002260
3502 043502 000403

```

```

.SBTTL TEST 4: VECTOR AND INTERRUPT TEST
;*****
;*****
;TEST 4 - VECTOR AND INTERRUPT TEST
;TEST 3 IS REPEATED, BUT WITH INTERRUPTS ENABLED.
;THE PROGRAM VERIFIES THAT AN INTERRUPT OCCURS AT
;THE END OF STEPS 1 - 3.
;*****
;*****
T4:: BGNTST
1$:
BIT #DRPFLG,LUNFLG(R4) ;IF UUT NOT DROPPED
BEQ G04 ; THEN DO TEST
RFLAGS FLAGS ;ELSE GO GET SUPRVISOR FLAGS
TRAP C$RFLA
MOV R0,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,R0
TRAP C$PNTF
ADD #6,SP
EXIT TST ;BYPASSED MESSAGE AND GET OUT
TRAP C$EXIT
.WORD L10023-.
G04:
BIC #BPFLAG,LUNFLG(R4) ;DO TEST WITH PRIORITY SET TO 0
BIC #INTFLG,LUNFLG(R4) ;CLEAR THE INTERRUPT FLAG
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$:
JSR PC,VECTOR ;SET UP VECTOR WITH INTERRUPT HANDLER
MOV #0,R5 ;SET UP R5 AS INDEX TO STEP TABLES
MOV #1,INISTP ;STEP 1 FOR ERROR PRINTOUT
MOV TKVEC(R4),STPTBL ;PUT VECTOR IN STEP 1
ASR STPTBL ;DIVIDE BY TWO
ASR STPTBL ;DIVIDE BY FOUR
MOV STPTBL,CMPTBL+4 ;PUT VECTOR IN STEP 3 COMPARE
BIS #104600,STPTBL ;REST OF STEP ONE
READBUS ;IF QBUS
TRAP C$RDBU
BCOMLETE 3$ ; THEN BR FOR QBUS SET UP
BCS 3$
MOV #4700,CMPTBL ; ELSE SET UP FOR UNIBUS
BR 4$

```

E8

HARDWARE TEST MACRO V05.05 Wednesday 27 Feb-91 11:55 Page 49-1
TEST 4: VECTOR AND INTERRUPT TEST

SEQ 0095

```

3503 043504 012737 005700 002260 3$: MOV #B.S1!B.QB!B.DI!B.OO!B.MP,CMPTBL
3504 ;STEP 1 COMPARE VALUE
3505 043512 012737 060050 002252 4$: MOV #COMMAR,STPTBL+2 ;STEP 2 - COMM AREA ADDRESS
3506 043520 012737 010211 002262 MOV #010211,CMPTBL+2 ;STEP 2 COMPARE
3507 043526 012737 000000 002254 MOV #0,STPTBL+4 ;STEP 3 - HIGH ADDRESS
3508 043534 052737 000200 002264 BIS #B.IE,CMPTBL+4 ;SET THE INTERRUPT ENABLE BIT
3509 043542 112737 000040 002265 MOV #40,CMPTBL+5 ;REST OF STEP 3 COMPARE
3510 043550 012737 000000 002256 MOV #0,STPTBL+6 ;STEP 4
3511 043556 012737 040000 002266 MOV #040000,CMPTBL+6 ;STEP 4 COMPARE
3512
3513 043564 004737 036474 JSR PC,STEP1 ;GO DO IT
3514 043570 005737 002320 TST STEPST ;IF STATUS OKAY
3515 043574 001412 BEQ 5$ ; THEN CONTINUE TEST
3516
3517 043576 ERRDF 14.,EMSG9,PRIINI ;"SA CONTENTS IN ERROR"
043576 104455 TRAP C$ERDF
043600 000016 .WORD 14
043602 030473 .WORD EMSG9
043604 033000 .WORD PRIINI
3518 043606 CKLOOP ;LOOP ON ERROR?
043606 104406 TRAP C$CLP1
3519 043610 DODU LOGUNT ;DROP UUT
043610 013700 002312 MOV LOGUNT,R0
043614 104451 TRAP C$DODU
3520 043616 ESCAPE TST ;LEAVE TST
043616 104410 TRAP C$ESCAPE
043620 000246 .WORD L10023-.
3521
3522 043622 012737 000100 002326 5$: MOV #100,OUTER ;SET UP FOR DELAY ROUTINE
3523 043630 016537 002260 002314 MOV CMPTBL(R5),SAEXP ;SET UP FOR COMPARE
3524 043636 012737 037200 002324 7$: MOV #16000.,INNER ;SET UP INNER
3525 043644 032764 000002 000014 BIT #INTFLG,LUNFLG(R4) ;IF INTERRUPT OCCURRED
3526 043652 001017 BNE 10$ ; THEN SEE IF SA IS CORRECT
3527 043654 004737 036446 9$: JSR PC,PDELAY ; ELSE GIVE UUT SOME TIME
3528 043660 005737 002330 TST TOUT ;IF NO TIMEOUT YET
3529 043664 001764 BEQ 7$ ; THEN GO TAKE ANOTHER LOOK
3530
3531 043666 ERRDF 15.,EMSG11,PRIERR ;"EXPECTED INTERRUPT DID NOT OCCUR"
043666 104455 TRAP C$ERDF
043670 000017 .WORD 15
043672 030546 .WORD EMSG11
043674 033206 .WORD PRIERR
3532 043676 CKLOOP
043676 104406 TRAP C$CLP1
3533 043700 DODU LOGUNT
043700 013700 002312 MOV LOGUNT,R0
043704 104451 TRAP C$DODU
3534 043706 ESCAPE TST
043706 104410 TRAP C$ESCAPE
043710 000156 .WORD L10023-.
3535
3536 043712 042764 000002 000014 10$: BIC #INTFLG,LUNFLG(R4) ;CLEAR THE INTERRUPT FLAG
3537 043720 005237 002316 INC INISTP ;ADJUST THE STEP COUNTER
3538 043724 062705 000002 ADD #2,R5 ;ADJUST TABLE INDEX
3539 043730 016537 002260 002314 MOV CMPTBL(R5),SAEXP ;GET THE COMPARISON VALUE
3540 043736 017464 000002 000012 MOV @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
3541 043744 022705 000006 CMP #6,R5 ;ARE WE IN STEP 4?

```


TEST 4: VECTOR AND INTERRUPT TEST

```

3542 043750 001005          BNE      15$          ;BRANCH IF NOT
3543 043752 033764 002314 000012  BIT      SAEXP,TKSASV(R4) ;JUST LOOK FOR STEP 4 BIT
3544 043760 001017          BNE      20$          ;IT'S SET SO LET'S GO
3545 043762 000404          BR       16$          ;ERROR
3546 043764 023764 002314 000012 15$:  CMP      SAEXP,TKSASV(R4) ;IF SA IS WHAT WE EXPECT
3547 043772 001412          BEQ      20$          ; THEN MOVE ALONG
3548
3549 043774          16$:  ERRDF   16.,EMSG9,PRIINI ;"SA CONTENTS IN ERROR"
      043774 104455      TRAP   C$ERDF
      043776 000020      .WORD  16
      044000 030473      .WORD  EMSG9
      044002 033000      .WORD  PRIINI
3550 044004          CKLOOP
      044004 104406      TRAP   C$CLP1
3551 044006          DODU   LOGUNT
      044006 013700 002312  MOV     LOGUNT,R0
      044012 104451      TRAP   C$DODU
3552 044014          ESCAPE  TST
      044014 104410      TRAP   C$ESCAPE
      044016 000050      .WORD  L10023-.
3553
3554 044020 016574 002250 000002 20$:  MOV     STPTBL(R5),@TKSA(R4) ;WRITE NEXT STEP TO UUT
3555 044026 022705 000006          CMP     #6,R5          ;IF NOT IN STEP 4
3556 044032 001273          BNE     5$            ;GO BACK TO MAIN LOOP
3557
3558 044034 032764 000001 000014  BIT     #DRPFLG,LUNFLG(R4) ;HAS UUT BEEN DROPPED
3559 044042 001005          BNE     T4EXT         ;LEAVE NOW IF SO
3560 044044 005377 002272          DEC     ITRCNT        ;IF NO MORE ITERATIONS LEFT
3561 044050 001402          BEQ     T4EXT         ; THEN EXIT
3562 044052 000137 043420          JMP     2$            ; ELSE DO IT AGAIN
3563
3564 044056 004737 036326          T4EXT: JSR     PC,RSTVEC ;CATCH ILLEGAL INTERRUPTS
3565 044062          EXIT   TST
      044062 104432      TRAP   C$EXIT
      044064 000002      .WORD  L10023-.
3566
3567 044066          ENDTST
      044066          L10023:
      044066 104401      TRAP   C$ETST

```

3570
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619

044070
044070
044070
044076
044100
044100
044102
044106
044114
044116
044116
044122
044126
044132
044134
044136
044142
044142
044144
044146
044154
044162
044170
044176
044200
044206
044206
044212
044214
044220
044224
044232
044240
044244
044250
044256
044264
044270

032764
001423
104421
010037
032737
001412
013746
012746
012746
010600
104417
062706
104432
000370
052764
012737
012737
022737
001403
012737
012700
012700
104441
004737
012705
012737
016437
006237
006237
052737
016437
004737
005737

.SBTTL TEST 5: BR LEVEL TEST

```
*****  
*****  
:TEST 5 - BR LEVEL TEST  
:THIS TEST INSURES THAT THE TKxx CAN NOT INTERRUPT  
:WHEN THE CPU PRIORITY IS SET TO 7. THE TEST GOES  
:ONLY THROUGH THE FIRST STEP OF THE INIT SEQUENCE  
:SINCE THE CONTROLLER WILL "HANG" WAITING FOR THE  
:INTERRUPT ACKNOWLEDGE.  
*****  
*****
```

BGNTST
T5::

```
BIT #DRPFLG,LUNFLG(R4) ;IF UUT NOT DROPPED  
BEQ G05 ; 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$PNTF  
ADD #6,SP  
3599 044142 1$: EXIT TST ;BYPASSED MESSAGE AND GET OUT  
TRAP C$EXIT  
.WORD L10024-.  
G05: BIS #BRFLAG,LUNFLG(R4) ;DO TEST WITH HIGH PRIORITY  
MOV #CTRL,FRUIS ;FAILING FRU IN CASE OF ERROR  
MOV #1,ITRCNT ;SET UP FOR ONE TEST ITERATION  
CMP #1,PASCNT ;IF FIRST PASS  
BEQ 2$ ; THEN START TEST  
MOV #2,ITRCNT ; ELSE DO 10 ITERATIONS  
2$: SETPRI #PRI07 ;CPU PRIORITY 7  
MOV #PRI07,RO  
TRAP C$SPRI  
JSR PC,VECTOR ;SET UP VECTOR WITH INTERRUPT HANDLER  
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  
BIS #104600,STPTBL ;REST OF STEP ONE  
MOV TKVEC(R4),CMPTBL ;STEP 1 COMPARE VALUE  
JSR PC,STEP1 ;GO DO IT  
TST STEPST ;IF STATUS OKAY
```

H8

```

3620 044274 001412          BEQ      5$          ; THEN CONTINUE TEST
3621
3622 044276          ERRDF    14.,EMSG9,PRIINI ; "SA CONTENTS IN ERROR"
    044276 104455      TRAP     C$ERDF
    044300 000016      .WORD    14
    044302 030473      .WORD    EMSG9
    044304 033000      .WORD    PRIINI
3623 044306          CKLOOP           ; LOOP ON ERROR?
    044306 104406      TRAP     C$CLP1
3624 044310          DODU     LOGUNT          ; DROP UUT
    044310 013700 002312 MOV     LOGUNT,R0
    044314 104451      TRAP     C$DODU
3625 044316          ESCAPE   TST          ; LEAVE TST
    044316 104410      TRAP     C$ESCAPE
    044320 000214      .WORD    L10024-.
3626
3627 044322 012737 000100 002326 5$:   MOV     #100,OUTER      ; SET UP FOR DELAY ROUTINE
3628 044330 016537 002260 002314      MOV     CMPTBL(R5),SAEXP ; SET UP FOR COMPARE
3629 044336 012737 037200 002324 7$:   MOV     #16000.,INNER  ; SET UP INNER
3630 044344 004737 036446 9$:   JSR     PC,PDELAY      ; ELSE GIVE UUT SOME TIME
3631 044350 005737 002330          TST     TOUT          ; IF NO TIMEOUT YET
3632 044354 001770          BEQ     7$          ; THEN GO TAKE ANOTHER LOOK
3633
3634 044356 017464 000002 000012      MOV     @TKSA(R4),TKSASV(R4) ; GET SA CONTENTS
3635 044364 023764 002314 000012      CMP     SAEVP,TKSASV(R4) ; IF CONTENTS OKAY
3636 044372 001412          BEQ     10$         ; THEN CHECK FOR INTERRUPT
3637
3638 044374          ERRDF    17.,EMSG9,PRIINI ; "SA CONTENTS IN ERROR"
    044374 104455      TRAP     C$ERDF
    044376 000021      .WORD    17
    044400 030473      .WORD    EMSG9
    044402 033000      .WORD    PRIINI
3639 044404          CKLOOP           ; LOOP ON ERROR?
    044404 104406      TRAP     C$CLP1
3640 044406          DODU     LOGUNT          ; DROP UUT
    044406 013700 002312 MOV     LOGUNT,R0
    044412 104451      TRAP     C$DODU
3641 044414          ESCAPE   TST          ; LEAVE TST
    044414 104410      TRAP     C$ESCAPE
    044416 000116      .WORD    L10024-.
3642
3643 044420 032764 000002 000014 10$:  BIT     #INTFLG,LUNFLG(R4) ; IF NO INTERRUPT OCCURRED
3644 044426 001415          BEQ     20$         ; THEN CARRY ON WITH TEST
3645 044430 042764 000002 000014      BIC     #INTFLG,LUNFLG(R4) ; CLEAR FLAG IN CASE WE'RE LOOPING
3646 044436          ERRDF    18.,EMSG12,PRIINI ; "INTRPT WITH CPU PRIORITY =7"
    044436 104455      TRAP     C$ERDF
    044440 000022      .WORD    18
    044442 030607      .WORD    EMSG12
    044444 033000      .WORD    PRIINI
3647 044446          CKLOOP           ; LOOP ON ERROR?
    044446 104406      TRAP     C$CLP1
3648 044450          DODU     LOGUNT          ; DROP UUT
    044450 013700 002312 MOV     LOGUNT,R0
    044454 104451      TRAP     C$DODU
3649 044456          ESCAPE   TST          ; LEAVE TST
    044456 104410      TRAP     C$ESCAPE
    044460 000054      .WORD    L10024 .

```

TEST 5: BR LEVEL TEST

```

3650
3651 044462          20$:  SETPRI  #PRI00          ;CPU PRIORITY = 0
      044462 012700 000000
      044466 104441
3652 044470 000240
3653 044472 000240
3654 044474 042764 000002 000014
      3655
      3656 044502 032764 000001 000014
      3657 044510 001005
      3658 044512 005337 002272
      3659 044516 001402
      3660 044520 000137 044206
      3661
      3662 044524 004737 036326
      3663 044530
      044530 104432
      044532 000002
      3664
      3665 044534
      044534
      044534 104401

```

```

20$:  SETPRI  #PRI00          ;CPU PRIORITY = 0
      MOV     #PRI00,R0
      TRAP   C$SPRI
      NOP
      NOP
      BIC    #INTFLG,LUNFLG(R4) ;DELAY FOR PENDING INTERRUPT
                                          ;CLEAR THE FLAG NOW
      BIT    #DRPFLG,LUNFLG(R4) ;HAS UUT BEEN DROPPED
      BNE    TSEXT             ;LEAVE NOW IF SO
      DEC    ITRCNT           ;IF NO MORE ITERATIONS LEFT
      BEQ    TSEXT            ; THEN EXIT
      JMP    2$              ; ELSE DO IT AGAIN

TSEXT: JSR    PC,RSTVEC       ;CATCH ILLEGAL INTERRUPTS
      EXIT   TST
      TRAP   C$EXIT
      .WORD  L10024-.

L10024: ENDTST
      TRAP   C$ETST

```

3668
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718

044536
044536
044536 032764 000001 000014
044544 001423
044546 104421
044550 010037 002336
044554 032737 001000 002336
044562 001412
044564 013746 002114
044570 012746 050026
044574 012746 000002
044600 010600
044602 104417
044604 062706 000006
044610 104432
044612 000542
044614 012737 032700 002310
044622 012737 000001 002272
044630 022737 000001 002270
044636 001403
044640 012737 000012 002272
044646 012705 000000
044652 012737 000001 002316
044660 016437 000004 002250
044666 006237 002250
044672 006237 002250
044676 013737 002250 002264
044704 052737 104400 002250
044712 104407

.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. WHEN 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 WUT NOT DROPPED  
BEQ G06 ; THEN DO TEST  
RFLAGS FLAGS ;ELSE GO GET SUPRVISOR FLAGS  
TRAP C$RFLA  
MOV R0,FLAGS  
BIT #PNT,FLAGS ;SEE IF WE'RE PRINTING TEST NUMBERS  
BEQ 1$ ;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,R0  
TRAP C$PNTF  
ADD #6,SP  
EXIT TST ;BYPASSED MESSAGE AND GET OUT  
TRAP C$EXIT  
.WORD L10025-.  
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$: MOV #0,R5 ;SET UP R5 AS INDEX TO STEP TABLES  
MOV #1,INISTP ;STEP 1 FOR ERROR PRINTOUT  
MOV TKVEC(R4),STPTBL ;PUT VECTOR IN STEP 1  
ASR STPTBL ;DIVIDE BY TWO  
ASR STPTBL ;DIVIDE BY FOUR  
MOV STPTBL,CMPTBL+4 ;PUT VECTOR IN STEP 3 COMPARE  
BIS #104400,STPTBL ;REST OF STEP ONE  
READBUS ;IF QBUS  
TRAP C$RDBU
```

```

3719 044714          BCOMPLETE      3$          ; THEN BR FOR GBUS SET UP
      044714 103404          BCS      3$
3720 044716 012737 004700 002260      MOV      #4700,CMPTBL      ; ELSE SET UP FOR UNIBUS
3721 044724 000403          BR      4$
3722 044726 012737 005700 002260 3$:      MOV      #B.S1!B.GB!B.DI!B.OD!B.MP,CMPTBL
3723          ;STEP 1 COMPARE VALUE
3724 044734 012737 060050 002252 4$:      MOV      #COMMAR,STPTBL+2      ;STEP 2 - COMM AREA ADDRESS
3725 044742 012737 010211 002262          MOV      #010211,CMPTBL+2      ;STEP 2 COMPARE
3726 044750 012737 100000 002254          MOV      #B.PP,STPTBL+4      ;STEP 3 - HIGH ADDRESS AND PRGE/POLL
3727 044756 112737 000040 002265          MOV      #40,CMPTBL+5      ;REST OF STEP 3 COMPARE
3728 044764 012737 000000 002256          MOV      #0,STPTBL+6      ;STEP 4
3729 044772 012737 040000 002266          MOV      #040000,CMPTBL+6      ;STEP 4 COMPARE
3730          ;LENGTH OF COMM AREA FOR THIS TEST
3731 045000 012737 000012 002306          MOV      #10,CMARLG
3732 045006 004737 036620          JSR      PC,BAKPAT      ;FILL COMM AREA WITH ALL 1'S DATA
3733          ;GO DO IT
3734 045012 004737 036474          JSR      PC,STEP1
3735 045016 005737 002320          TST      STEPST      ;IF STATUS OKAY
3736 045022 001412          BEQ      5$          ; THEN CONTINUE TEST
3737          ;"SA CONTENTS IN ERROR"
3738 045024          ERRDF      19,EMSG9,PRIINI
      045024 104455          TRAP      C$ERDF
      045026 000023          .WORD      19
      045030 030473          .WORD      EMSG9
      045032 033000          .WORD      PRIINI
3739 045034          CKLOOP
      045034 104406          TRAP      C$CLP1      ;LOOP ON ERROR?
3740 045036          DODU      LOGUNT      ;DROP UUT
      045036 013700 002312          MOV      LOGUNT,R0
      045042 104451          TRAP      C$DODU
3741 045044          ESCAPE      TST      ;LEAVE TST
      045044 104410          TRAP      C$ESCAPE
      045046 000306          .WORD      L10025-.
3742          ;ADJUST STEP COUNTER
3743 045050 005237 002316          5$:      INC      INISTP
3744 045054 062705 000002          ADD      #2,R5      ;ADJUST TABLE INDEX
3745 045060 012737 000100 002326 6$:      MOV      #100,OUTER      ;SET UP FOR DELAY ROUTINE
3746 045066 016537 002260 002314          MOV      CMPTBL(R5),SAEXP      ;SET UP FOR COMPARE
3747 045074 012737 037200 002324 7$:      MOV      #16000,INNER      ;SET UP INNER
3748 045102 017464 000002 000012          MOV      @TKSA(R4),TKSASV(R4)      ;GET SA CONTENTS
3749 045110 022705 000006          CMP      #6,R5      ;ARE WE IN STEP 4?
3750 045114 001005          BNE      8$          ;BRANCH IF NOT
3751 045116 033764 002314 000012          BIT      SAEXP,TKSASV(R4)      ;JUST LOOK FOR STEP 4 BIT
3752 045124 001024          BNE      10$          ;IT'S SET SO LFT'S GO
3753 045126 000404          BR      9$          ;STAY IN LOOP OTHERWISE
3754 045130 023764 002314 000012 8$:      CMP      SAEXP,TKSASV(R4)      ;IF SA IS WHAT WE EXPECT
3755 045136 001417          BEQ      10$          ; THEN MOVE ALONG
3756 045140 004737 036446          9$:      JSR      PC,PDELAY      ; ELSE GIVE UUT SOME TIME
3757 045144 005737 002330          TST      TOUT      ;IF NO TIMEOUT YET
3758 045150 001751          BEQ      7$          ; THEN GO TAKE ANOTHER LOOK
3759          ;"SA CONTENTS IN ERROR"
3760 045152          ERRDF      20,EMSG9,PRIINI
      045152 104455          TRAP      C$ERDF
      045154 000024          .WORD      20
      045156 030473          .WORD      EMSG9
      045160 033000          .WORD      PRIINI
3761 045162          CKLOOP

```

L8

```

3762 045162 104406 TRAP C$CLP1
045164 DODU LOGUNT
045164 013700 002312 MOV LOGUNT,R0
045170 104451 TRAP C$DODU
3763 045172 ESCAPE TST
045172 104410 TRAP C$ESCAPE
045174 000160 .WORD L10025-.

3764
3765 045176 016574 002250 000002 10$: MOV STPTBL(R5),@TKSA(R4) ;WRITE NEXT STEP TO UUT
3766 045204 022705 000004 CMP #4,R5 ;IF STEP 3
3767 045210 001404 BEQ 15$ ; THEN DO PURGE/POLL STUFF
3768 045212 022705 000006 CMP #6,R5 ;IF NOT IN STEP 4
3769 045216 001314 BNE 5$ ; THEN GO BACK TO MAIN LOOP
3770 045220 000440 BR 20$ ; ELSE GO CHECK RESULTS
3771
3772 045222 15$: DELAY 1 ;GIVE PORT SOME TIME
045222 012727 000001 MOV #1,(PC)+
045226 000000 .WORD 0
045230 013727 002116 MOV L$DLY,(PC)+
045234 000000 .WORD 0
045236 005367 177772 DEC -6(PC)
045242 001375 BNE .-4
045244 005367 177756 DEC -22(PC)
045250 001367 BNE .-20
3773 045252 017464 000002 000012 MOV @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
3774 045260 001412 BEQ 16$ ;BRANCH IF OKAY
3775
3776 045262 ERRDF 21,EMSG13,PRIINI ;SA NOT 0 IN PURGE/POLL
045262 104455 TRAP C$ERDF
045264 000025 .WORD 21
045266 030656 .WORD EMSG13
045270 033000 .WORD PRIINI
3777 045272 CKLOOP
045272 104406 TRAP C$CLP1
3778 045274 DODU LOGUNT
045274 013700 002312 MOV LOGUNT,R0
045300 104451 TRAP C$DODU
3779 045302 ESCAPE TST
045302 104410 TRAP C$ESCAPE
045304 000050 .WORD L10025-.

3780
3781 045306 012774 000000 000002 16$: MOV #0,@TKSA(R4) ;WRITE 0'S TO SA
3782 045314 005774 000000 TST @TKIP(R4) ;AND READ IP
3783 045320 000653 BR 5$ ;GO WAIT FOR NEXT TRANSITION
3784
3785 045322 004737 036650 20$: JSR PC,CHKCOM ;GO CHECK COMM AREA
3786 045326 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;HAS UUT BEEN DROPPED
3787 045334 001005 BNE T6EXT ;LEAVE NOW IF SO
3788 045336 005337 002272 DEC ITRCNT ;IF NO MORE ITERATIONS LEFT
3789 045342 001402 BEQ T6EXT ; THEN LEAVE TEST
3790 045344 000137 044646 JMP 2$ ; ELSE DO IT AGAIN
3791
3792 045350 T6EXT: EXIT TST
045350 104432 TRAP C$EXIT
045352 000002 .WORD L10025-.

3793
3794 045354 END1ST
  
```

M8

HARDWARE TEST MACRO V05.05 Wednesday 27-Feb-91 11:55 Page 51-3

SEQ 0103

TEST 6: PURGE AND POLL TEST

045354
045354 104401

L10025:
TRAP C\$ETST

N8

```

3797 .SBTTL TEST 7: MAXIMUM RING BUFFER TEST
3798
3799 045356 STARS
3800 045356 ;*****
STARS
;*****
3801 ;
3802 ;TEST 7 - MAXIMUM RING BUFFER TEST
3803 ; THIS TEST IS SIMILAR TO TEST 5, BUT IT WILL UTILIZE
3804 ; THE MAXIMUM ALLOWABLE RING DEPTH AS SPECIFIED IN
3805 ; UQSSP. THIS VALUE IS EQUAL TO 128 COMMAND AND 128
3806 ; RESPONSE SLOTS OF 32 BITS PER SLOT.
3807 ;
3808 045356 STARS
3809 045356 ;*****
STARS
;*****
3810
3811 045356 BGNTST
045356 T7::
3812
3813 045356 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;IF UUT NOT DROPPED
3814 045364 001423 BEQ G07 ; THEN DO TEST
3815 045366 RFLAGS FLAGS ;ELSE GO GET SUPRVISOR FLAGS
045366 104421 TRAP C$RFLA
045370 010037 002336 MOV R0,FLAGS
3816 045374 032737 001000 002336 BIT #PNT,FLAGS ;SEE IF WE'RE PRINTING TEST NUMBERS
3817 045402 001412 BEQ 1$ ;NO, DON'T PRINT BYPASSED
3818 045404 PRINTF #BYPASS,L$TEST ; ELSE PRINT THE TEST
045404 013746 002114 MOV L$TEST,-(SP)
045410 012746 050026 MOV #BYPASS,-(SP)
045414 012746 000002 MOV #2,-(SP)
045420 010600 MOV SP,R0
045422 104417 TRAP C$PNTF
045424 062706 000006 ADD #6,SP
3819 045430 1$: EXIT TST ;BYPASSED MESSAGE AND GET OUT
045430 104432 TRAP C$EXIT
045432 000542 .WORD L10026-.
3820
3821 045434 012737 032700 002310 G07: MOV #CTRL,FRUIS ;FAILING FRU IN CASE OF ERROR
3822 045442 012737 000001 002272 MOV #1,ITRCNT ;SET UP FOR ONE TEST ITERATION
3823 045450 022737 000001 002270 CMP #1,PASCNT ;IF FIRST PASS
3824 045456 001403 BEQ 2$ ; THEN START TEST
3825 045460 012737 000012 002272 MOV #10.,ITRCNT ; ELSE DO 10 ITERATIONS
3826
3827 045466 012705 000000 2$: MOV #0,R5 ;SET UP R5 AS INDEX TO STEP TABLES
3828 045472 012737 000001 002316 MOV #1,INISTP ;STEP 1 FOR ERROR PRINTOUT
3829 045500 016437 000004 002250 MOV TKVEC(R4),STPTBL ;PUT VECTOR IN STEP 1
3830 045506 006237 002250 ASR STPTBL ;DIVIDE BY TWO
3831 045512 006237 002250 ASR STPTBL ;DIVIDE BY FOUR
3832 045516 013737 002250 002264 MOV STPTBL,CMPTBL+4 ;PUT VECTOR IN STEP 3 COMPARE
3833 045524 052737 137400 002250 BIS #137400,STPTBL ;REST OF STEP ONE
3834 045532 READBUS ;IF QBUS
045532 104407 TRAP C$RDBU
3835 045534 BCOMPLETE 3$ ; THEN BR FOR QBUS SET UP
045534 103404 BCS 3$
3836 045536 012737 004700 002260 MOV #4700,CMPTBL ; ELSE SET UP FOR UNIBUS

```

```

3837 045544 000403          BR      4$
3838 045546 012737 005700 002260 3$:  MOV    #B.S1!B.QB!B.DI!B.OO!B.MP,CMP TBL
3839                                ;STEP 1 COMPARE VALUE
3840 045554 012737 060050 002252 4$:  MOV    #COMMAR,STPTBL+2          ;STEP 2 - COMM AREA ADDRESS
3841 045562 012737 010277 002262          MOV    #010277,CMP TBL+2          ;STEP 2 COMPARE
3842 045570 012737 100000 002254          MOV    #B.PP,STPTBL+4          ;STEP 3 HIGH ADDRESS AND PRGE/POLL
3843 045576 112737 000040 002265          MOV    #40,CMP TBL+5          ;REST OF STEP 3 COMPARE
3844 045604 012737 000000 002256          MOV    #0,STPTBL+6          ;STEP 4
3845 045612 012737 040000 002266          MOV    #040000,CMP TBL+6      ;STEP 4 COMPARE
3846
3847 045620 012737 001002 002306          MOV    #514.,CMARLG          ;LENGTH OF COMM AREA FOR THIS TEST
3848 045626 004737 036620          JSR    PC,BAKPAT            ;FILL COMM AREA WITH ALL 1'S DATA
3849
3850 045632 004737 036474          JSR    PC,STEP1            ;GO DO IT
3851 045636 005737 002320          TST   STEPST              ;IF STATUS OKAY
3852 045642 001412          BEQ   5$                  ; THEN CONTINUE TEST
3853
3854 045644          ERRDF  22.,EMSG9,PRIINI    ;"SA CONTENTS IN ERROR"
3855 045644 104455          TRAP  C$ERDF
3856 045646 000026          .WORD 22
3857 045650 030473          .WORD EMSG9
3858 045652 033000          .WORD PRIINI
3859 045654          CKLOOP
3860 045654 104406          TRAP  C$CLP1              ;LOOP ON ERROR?
3861 045656          DODU  LOGUNT              ;DROP UUT
3862 045656 013700 002312          MOV    LOGUNT,R0
3863 045662 104451          TRAP  C$DODU
3864 045664          ESCAPE TST                ;LEAVE TST
3865 045664 104410          TRAP  C$ESCAPE
3866 045666 000306          .WORD L10026-.
3867
3868 045670 005237 002316          5$:  INC    INISTP              ;ADJUST STEP COUNTER
3869 045674 062705 000002          ADD   #2,R5                ;ADJUST TABLE INDEX
3870 045700 012737 000100 002326 6$:  MOV    #100,OUTER          ;SET UP FOR DELAY ROUTINE
3871 045706 016537 002260 002314          MOV    CMP TBL(R5),SAEXP    ;SET UP FOR COMPARE
3872 045714 012737 037200 002324 7$:  MOV    #16000.,INNER       ;SET UP INNER
3873 045722 017464 000002 000012          MOV    @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
3874 045730 022705 000006          CMP   #6,R5                ;ARE WE IN STEP 4?
3875 045734 001005          BNE   8$                  ;BRANCH IF NOT
3876 045736 033764 002314 000012          BIT   SAEXP,TKSASV(R4)     ;JUST LOOK FOR STEP 4 BIT
3877 045744 001024          BNE   10$                 ;IT'S SET SO LET'S GO
3878 045746 000404          BR    9$                  ;STAY IN LOOP OTHERWISE
3879 045750 023764 002314 000012 8$:  CMP   SAEXP,TKSASV(R4)     ;IF SA IS WHAT WE EXPECT
3880 045756 001417          BEQ   10$                 ; THEN MOVE ALONG
3881 045760 004737 036446          9$:  JSR    PC,PDELAY          ; ELSE GIVE UUT SOME TIME
3882 045764 005737 002330          TST   TOUT                ;IF NO TIMEOUT YET
3883 045770 001751          BEQ   7$                  ; THEN GO TAKE ANOTHER LOOK
3884
3885 045772          ERRDF  23.,EMSG9,PRIINI    ;"SA CONTENTS IN ERROR"
3886 045772 104455          TRAP  C$ERDF
3887 045774 000027          .WORD 23
3888 045776 030473          .WORD EMSG9
3889 046000 033000          .WORD PRIINI
3890 046002          CKLOOP
3891 046002 104406          TRAP  C$CLP1              ;LOOP ON ERROR?
3892 046004          DODU  LOGUNT              ;DROP UUT
3893 046004 013700 002312          MOV    LOGUNT,R0
    
```

C9

TEST 7: MAXIMUM RING BUFFER TEST

```

3879 046010 104451 TRAP C$DODU
046012 ESCAPE TST
046012 104410 TRAP C$ESCAPE
046014 000160 .WORD L10026-.

3880
3881 046016 016574 002250 000002 10$: MOV STPTBL(R5),@TKSA(R4) ;WRITE NEXT STEP TO UUT
3882 046024 022705 000004 CMP #4,R5 ;IF STEP 3
3883 046030 001404 BEQ 15$ ; THEN DO PURGE/POLL STUFF
3884 046032 022705 000006 CMP #6,R5 ;IF NOT IN STEP 4
3885 046036 001314 BNE 5$ ; THEN GO BACK TO MAIN LOOP
3886 046040 000440 BR 20$ ; ELSE GO CHECK RESULTS
3887
3888 046042 15$: DELAY 1 ;GIVE PORT SOME TIME
046042 012727 000001 MOV #1,(PC)+
046046 000000 .WORD 0
046050 013727 002116 MOV L$DLY,(PC)+
046054 000000 .WORD 0
046056 005367 177772 DEC -6(PC)
046062 001375 BNE -.4
046064 005367 177756 DEC -22(PC)
046070 001367 BNE -.20
3889 046072 017464 000002 000012 MOV @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
3890 046100 001412 BEQ 16$ ;BRANCH IF OKAY
3891
3892 046102 ERRDF 24.,EMSG13,PRIINI ;SA NOT 0 IN PURGE/POLL
046102 104455 TRAP C$ERDF
046104 000030 .WORD 24
046106 030656 .WORD EMSG13
046110 033000 .WORD PRIINI
3893 046112 CKLOOP
046112 104406 TRAP C$CLP1
3894 046114 DODU LOGUNT
046114 013700 002312 MOV LOGUNT,R0
046120 104451 TRAP C$DODU
3895 046122 ESCAPE TST
046122 104410 TRAP C$ESCAPE
046124 000050 .WORD L10026-.

3896
3897 046126 012774 000000 000002 16$: MOV #0,@TKSA(R4) ;WRITE 0'S TO SA
3898 046134 005774 000000 TST @TKIP(R4) ;AND READ IP
3899 046140 000653 BR 5$ ;GO WAIT FOR NEXT TRANSITION
3900
3901 046142 004737 036650 20$: JSR PC,CHKCOM ;GO CHECK COMM AREA
3902 046146 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;HAS UUT BEEN DROPPED
3903 046154 001005 BNE T7EXT ;LEAVE NOW IF SO
3904 046156 005337 002272 DEC ITRCNT ;IF NO MORE ITERATIONS LEFT
3905 046162 001402 BEQ T7EXT ; THEN LEAVE TEST
3906 046164 000137 045466 JMP 2$ ; ELSE DO IT AGAIN
3907
3908 046170 T7EXT: EXIT TST
046170 104432 TRAP C$EXIT
046172 000002 .WORD L10026-.

3909
3910 046174 ENDTST
046174 L10026: TRAP C$ETST
046174 104401

```

```

3913 .SBTTL TEST 8: EXTENDED ADDRESS TEST
3914
3915 046176 STARS
;*****
3916 046176 STARS
;*****
3917
3918 ;TEST 8 - EXTENDED ADDRESS TEST
3919 ; THE FORMAT OF THIS TEST IS SIMILAR TO TEST 6, BUT THE
3920 ; PROGRAM WILL ESTABLISH THE COMMUNICATION AREA IN THE
3921 ; HIGHEST AVAILABLE MEMORY LOCATIONS. THIS WILL ALLOW
3922 ; TESTING OF THE UPPER SIX BITS OF ADDRESS LOGIC ON THE
3923 ; CONTROLLER BOARD.
3924
3925 046176 STARS
;*****
3926 046176 STARS
;*****
3927
3928 046176 BGNTST
046176 T8::
3929
3930 046176 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;IF UUT NOT DROPPED
3931 046204 001423 BEQ G08 ; THEN DO TEST
3932 046206 RFLAGS FLAGS ;ELSE GO GET SUPRVISOR FLAGS
046206 104421 TRAP C$RFLA
046210 010037 002336 MOV R0,FLAGS
3933 046214 032737 001000 002336 BIT #PNT,FLAGS ;SEE IF WE'RE PRINTING TEST NUMBERS
3934 046222 001412 BEQ 1$ ;NO, DON'T PRINT BYPASSED
3935 046224 PRINTF #BYPASS,L$TEST ; ELSE PRINT THE TEST
046224 013746 002114 MOV L$TEST,-(SP)
046230 012746 050026 MOV #BYPASS,-(SP)
046234 012746 000002 MOV #2,-(SP)
046240 010600 MOV SP,R0
046242 104417 TRAP C$PNTF
046244 062706 000006 ADD #6,SP
3936 046250 1$: EXIT TST ;BYPASSED MESSAGE AND GET OUT
046250 104432 TRAP C$EXIT
046252 000656 .WORD L10027-.
3937
3938 046254 005737 002274 G08: TST KTFLAG ;IF MEMORY MANAGEMENT AVAILABLE
3939 046260 001002 BNE 1$ ; THEN DO TEST
3940 046262 EXIT TST ; ELSE GET OUT
046262 104432 TRAP C$EXIT
046264 000644 .WORD L10027-.
3941 046266 012737 032700 002310 1$: MOV #CTRL,FRUIS ;FAILING FRU IN CASE OF ERROR
3942 046274 012737 000001 002272 MOV #1,IIRCNT ;SET UP FOR ONE TEST ITERATION
3943 046302 022737 000001 002270 CMP #1,PASCI ;IF FIRST PASS
3944 046310 001403 BEQ 2$ ; THEN START TEST
3945 046312 012737 000012 002272 MOV #10.,IIRCNT ; ELSE DO 10 ITERATIONS
3946
3947 046320 004737 037002 2$: JSR PC,INTMMU ;INITIALIZE MMU REGISTERS
3948 046324 012705 000000 3$: MOV #0,R5 ;SET UP R5 AS INDEX TO STEP TABLES
3949 046330 012737 000001 002316 MOV #1,INISTP ;STEP 1 FOR ERROR PRINTOUT
3950 046336 016437 000004 002250 MOV TKVEC(R4),STPTBL ;PUT VECTOR IN STEP 1
3951 046344 006237 002250 ASR STPTBL ;DIVIDE BY TWO
3952 046350 006237 002250 ASR STPTBL ;DIVIDE BY FOUR

```

E9

HARDWARE TEST MACRO V05.05 Wednesday 27-Feb-91 11:55 Page 53-1
TEST 8: EXTENDED ADDRESS TEST

SEQ 0108

```

3953 046354 013737 002250 002264      MOV      STPTBL,CMPTBL+4      ;PUT VECTOR IN STEP 3 COMPARE
3954 046362 052737 111000 002250      BIS      #111000,STPTBL      ;REST OF STEP ONE
3955 046370                                READBUS                       ;IF QBUS
      046370 104407      TRAP     C$RDBU
3956 046372                                BCOMPLETE      11$           ; THEN BR FOR QBUS SET UP
      046372 103404      BCS     11$
3957 046374 012737 004700 002260      MOV      #4700,CMPTBL      ; ELSE SET UP FOR UNIBUS
3958 046402 000403                                BR      12$
3959 046404 012737 005700 002260 11$:  MOV      #B.S1!B.QB!B.DI!B.OO!B.MP,CMPTBL
3960                                ;STEP 1 COMPARE VALUE
3961 046412 012737 060050 002252 12$:  MOV      #COMMAR,STPTBL+2      ;STEP 2 - COMM AREA ADDRESS
3962 046420 042737 160000 002252      BIC      #BIT15!BIT14!BIT13,STPTBL+2
3963                                ;CLEAR THE ACTIVE PAGE FIELD
3964 046426 012737 010222 002262      MOV      #010222,CMPTBL+2      ;STEP 2 COMPARE
3965 046434 013737 172346 002332      MOV      KPAR3,TEMP          ;GET RELOCATION VALUE
3966 046442 005037 002254                                CLR      STPTBL+4            ;CLEAR IT OUT *****RC
3967 046446 113737 002333 002254      MOVB    TEMP+1,STPTBL+4      ;JUST THE HGH BYTE
3968 046454 006237 002254                                ASR      STPTBL+4            ;MAKE IT THE EXTENDED
3969 046460 006237 002254                                ASR      STPTBL+4            ; ADDRESS OF THE COMM AREA
3970 046464 052737 100000 002254      BIS      #B.PP,STPTBL+4      ;NOW SET PURGE/POLL BIT
3971 046472 112737 000040 002265      MOVB    #40,CMPTBL+5        ;REST OF STEP 3 COMPARE
3972 046500 012737 000000 002256      MOV      #0,STPTBL+6        ;STEP 4
3973 046506 012737 040000 002266      MOV      #040000,CMPTBL+6    ;STEP 4 COMPARE
3974
3975 046514 012737 000022 002306      MOV      #18.,CMARLG         ;LENGTH OF COMM AREA FOR THIS TEST
3976 046522 004737 036620                                JSR      PC,BAKPAT          ;FILL COMM AREA WITH ALL 1'S DATA
3977
3978 046526 004737 036474                                JSR      PC,STEP1          ;GO DO IT
3979 046532 005737 002320                                TST     STEPST             ;IF STATUS OKAY
3980 046536 001412                                BEQ     5$                 ; THEN CONTINUE TEST
3981
3982 046540                                ERRDF   25.,MSG9,PRIINI     ;"SA CONTENTS IN ERROR"
      046540 104455      TRAP     C$ERDF
      046542 000031      .WORD   25
      046544 030473      .WORD   MSG9
      046546 033000      .WORD   PRIINI
3983 046550                                CKLOOP                                ;LOOP ON ERROR?
      046550 104406      TRAP     C$CLP1
3984 046552                                DODU      LOGUNT           ;DROP UUT
      046552 013700 002312      MOV      LOGUNT,R0
      046556 104451      TRAP     C$DODU
3985 046560                                ESCAPE   TST              ;LEAVE TST
      046560 104410      TRAP     C$ESCAPE
      046562 000346      .WORD   L10027-.
3986
3987 046564 005237 002316 5$:      INC     INISTP             ;ADJUST STEP COUNTER
3988 046570 062705 000002                                ADD     #2,R5              ;ADJUST TABLE INDEX
3989 046574 012737 000100 002326 6$:      MOV     #100,OUTER        ;SET UP FOR DELAY ROUTINE
3990 046602 016537 002260 002314      MOV     CMPTBL(R5),SAEXP   ;SET UP FOR COMPARE
3991 046610 012737 037200 002324 7$:      MOV     #16000.,INNER     ;SET UP INNER
3992 046616 017464 000002 000012      MOV     @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
3993 046624 022705 000006      CMP     #6,R5             ;ARE WE IN STEP 4?
3994 046630 001005                                BNE     8$                 ;BRANCH IF NOT
3995 046632 033764 002314 000012      BIT     SAEXP,TKSASV(R4)   ;JUST LOOK FOR STEP 4 BIT
3996 046640 001024                                BNE     10$                ;IT'S SET SO LET'S GO
3997 046642 000404                                BR      9$                 ;STAY IN LOOP OTHERWISE
3998 046644 023764 002314 000012 8$:      CMP     SAEXP,TKSASV(R4)   ;IF SA IS WHAT WE EXPECT

```

```

3999 046652 001417          BEQ      10$          ; THEN MOVE ALONG
4000 046654 004737 036446    9$:     JSR      PC,PDELAY ; ELSE GIVE UUT SOME TIME
4001 046660 005737 002330    TST      TOUT       ; IF NO TIMEOUT YET
4002 046664 001751          BEQ      7$          ; THEN GO TAKE ANOTHER LOOK
4003
4004 046666          ERRDF   26.,EMSG9,PRIINI ; "SA CONTENTS IN ERROR"
      046666 104455      TRAP    C$ERDF
      046670 000032      .WORD   26
      046672 030473      .WORD   EMSG9
      046674 033000      .WORD   PRIINI
4005 046676          CKLOOP
      046676 104406      TRAP    C$CLP1
4006 046700          DODU    LOGUNT
      046700 013700 002312    MOV     LOGUNT,R0
      046704 104451      TRAP    C$DODU
4007 046706          ESCAPE  TST
      046706 104410      TRAP    C$ESCAPE
      046710 000220      .WORD   L10027-.
4008
4009 046712 016574 002250 000002 10$:    MOV     STPTBL(R5),@TKSA(R4) ; WRITE NEXT STEP TO UUT
4010 046720 022705 000004    CMP     #4,R5          ; IF STEP 3
4011 046724 001404          BEQ     15$          ; THEN DO PURGE/POLL STUFF
4012 046726 022705 000006    CMP     #6,R5          ; IF NOT IN STEP 4
4013 046732 001314          BNE    5$            ; THEN GO BACK TO MAIN LOOP
4014 046734 000440          BR     20$          ; ELSE GO CHECK RESULTS
4015
4016 046736          15$:    DELAY   1            ; GIVE PORT SOME TIME
      046736 012727 000001    MOV     #1,(PC)+
      046742 000000      .WORD   0
      046744 013727 002116    MOV     L$DLY,(PC)+
      046750 000000      .WORD   0
      046752 005367 177772    DEC     -6(PC)
      046756 001375      BNE    -.4
      046760 005367 177756    DEC     -22(PC)
      046764 001367      BNE    -.20
4017 046766 017464 000002 000012    MOV     @TKSA(R4),TKSASV(R4) ; GET SA CONTENTS
4018 046774 001412          BEQ     16$          ; BRANCH IF OKAY
4019
4020 046776          ERRDF   27.,EMSG13,PRIINI ; SA NOT 0 IN PURGE/POLL
      046776 104455      TRAP    C$ERDF
      047000 000033      .WORD   27
      047002 030656      .WORD   EMSG13
      047004 033000      .WORD   PRIINI
4021 047006          CKLOOP
      047006 104406      TRAP    C$CLP1
4022 047010          DODU    LOGUNT
      047010 013700 002312    MOV     LOGUNT,R0
      047014 104451      TRAP    C$DODU
4023 047016          ESCAPE  TST
      047016 104410      TRAP    C$ESCAPE
      047020 000110      .WORD   L10027-.
4024
4025 047022 012774 000000 000002 16$:    MOV     #0,@TKSA(R4)    ; WRITE 0'S TO SA
4026 047030 005774 000000    TST     @TKIP(R4)      ; AND READ IP
4027 047034 000653          BR     5$            ; GO WAIT FOR NEXT TRANSITION
4028
4029 047036 004737 036650    20$:    JSR     PC,CHKCOM     ; GO CHECK COMM AREA
    
```

TEST 8: EXTENDED ADDRESS TEST

```

4030 047042 032764 000001 000014      BIT      #DRPFLG,LUNFLG(R4)      ;HAS UUT BEEN DROPPED
4031 047050 001023                    BNE      T8EXT                  ;LEAVE NOW IF SO
4032
4033 047052 062737 002000 172346      ADD      #2000,KPAR3           ;POINT TO NEXT 32KWORDS
4034 047060 005737 172346      TST      KPAR3                 ;DID THE ADDITION CAUSE AN OVERFLOW?
4035 047064 001406                    BEQ      25$                    ;STOP MMU IF OVERFLOW (4 MBYTES).
4036 047066 023737 002120 172346      CMP      $HIME,KPAR3          ;IF THERE'S NO MORE MEMORY AVAILABLE
4037 047074 103402                    BLO     _25$                   ; THEN CHECK FOR MORE ITERATIONS
4038 047076 000137 046324      JMP      3$                    ; ELSE DO IT AGAIN
4039
4040 047102 005037 177572      25$:  CLR      MMUSRO           ;SHUT DOWN MEMORY MANAGEMENT
4041 047106 005337 002272      DEC      ITRCNT              ;IF NO MORE ITERATIONS LEFT
4042 047112 001402                    BEQ      T8EXT                ; THEN LEAVE TEST
4043 047114 000137 046320      JMP      2$                    ; ELSE DO IT AGAIN
4044
4045 047120 005037 177572      T8EXT: CLR      MMUSRO        ;MAKE SURE IT'S OFF
4046 047124                    EXIT      TST
      047124 104432      TRAP     C$EXIT
      047126 000002      .WORD   L10027-.
4047
4048 047130                    ENDTST
      047130      L10027:
      047130 104401      TRAP     C$ETST

```

4051
4052
4053 047132
4054 047132
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066 047132

```
.SBTTL TEST 9:GET DUST STATUS
STARS
;*****
STARS
;*****
;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 AND 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.
STARS
;*****
STARS
;*****
```

4068
4069 047132
047132

```
BGNTST
T9::
```

4070
4071 047132 032764 000001 000014
4072 047140 001423
4073 047142
047142 104421
047144 010037 002336
4074 047150 032737 001000 002336
4075 047156 001412
4076 047160
047160 013746 002114
047164 012746 050026
047170 012746 000002
047174 010600
047176 104417
047200 062706 000006
4077 047204
047204 104432
047206 000146

```
BIT #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
BEQ G09 ;GO DO TEST IF AVAILABLE
RFLAGS FLAGS ;ELSE GO GET SUPRVISOR FLAGS
TRAP C$RFLA
MOV R0,FLAGS
BIT #PNT,FLAGS ;SEE IF WE'RE PRINTING TEST NUMBERS
BEQ 1$ ;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,R0
TRAP C$PNTF
ADD #6,SP
EXIT TST ;BYPASSED MESSAGE AND GET OUT
TRAP C$EXIT
.WORD L10030-
```

4078
4079 047210 052764 000100 002212
4080 047216 042764 000010 000014
4081 047224 012737 032700 002310
4082 047232 004737 037102
4083 047236 032764 000001 000014
4084 047244 001036
4085 047246 012705 002344
4086 047252 012737 000001 002272
4087 047260 012737 032723 023044
4088 047266 004737 037334
4089 047272 032764 000001 000014
4090 047300 001020
4091 047302 052764 000010 000014
4092 047310 012705 002460

```
G09: BIS #T9FLAG,LUNBLK(R4) ;SET T9FLAG ***FIX-RC
BIC #DUPFLG,LUNFLG(R4) ;CLEAR DUP FLAG
MOV #CTRL,FRUIS ;SET UP FRU POINTER
JSR PC,PRINT ;GO DO A PORT INITIALIZE
BIT #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
BNE T9EXT ;GET OUT IF NOT AVAILABLE
MOV #SCTRLC,R5 ;SET UP TO DO THE SCC COMMAND
MOV #1,I TRCNT ;DO IT ONCE
MOV #SCCCMD,CURCMD ;SET UP COMMAND ASCII
JSR PC,CLSDRV ;GO ISSUE THE COMMAND
BIT #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
BNE T9EXT ;GET OUT IF NOT AVAILABLE
BIS #DUPFLG,LUNFLG(R4) ;SET DUP FLAG FOR FOLLOWING COMMAND
MOV #GDUST,R5 ;SET UP TO DO GET DUST STATUS COMMAND
```


TEST 9:GET DUST STATUS

4093	047314	012737	000001	002272		MOV	#1,ITRCNT	
4094	047322	012737	032754	023044		MOV	#GDSCMD,CURCMD	;SET UP COMMAND ASCII
4095	047330	004737	037334			JSR	PC,CLSDRV	;GO ISSUE THE COMMAND
4096	047334	042764	000010	000014		BIC	#DUPFLG,LUNFLG(R4)	;CLEAR DUP FLAG
4097	047342	042764	000100	002212	T9EXT:	BIC	#T9FLAG,LUNBLK(R4)	;CLEAR T9FLAG
4098	047350					EXIT	TST	
	047350	104432				TRAP	C\$EXIT	
	047352	000002				.WORD	L10030-	
4099	047354					ENDTST		
	047354				L10030:			
	047354	104401				TRAP	C\$ETST	

```

4101 .SBTTL TEST 10:EXECUTE LOCAL PROGRAM (Level II microdiagnostics)
4102
4103 047356 STARS
4104 047356 ;:*****
STARS
;:*****
4105 ;
4106 ;TEST 10 - EXECUTE LOCAL PROGRAM
4107 ; THIS TEST WILL INVOKE, VIA THE DUP EXECUTE LOCAL
4108 ; PROGRAM COMMAND, THE CONTROLLER RESIDENT LEVEL 2
4109 ; MICRODIAGNOSTICS.
4110 ;
4111 047356 STARS
4112 047356 ;:*****
STARS
;:*****
4113
4114 047356 BGNTST
047356 T10::
4115
4116 047356 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
4117 047364 001423 BEQ GO10 ;GO DO TEST IF AVAILABLE
4118 047366 RFLAGS FLAGS ;ELSE GO GET SUPRVISOR FLAGS
047366 104421 TRAP C$RFLA
047370 010037 002336 MOV RO,FLAGS
4119 047374 032737 001000 002336 BIT #PNT,FLAGS ;SEE IF WE'RE PRINTING TEST NUMBERS
4120 047402 001412 BEQ 1$ ;NO, DON'T PRINT BYPASSED
4121 047404 PRINTF #BYPASS,L$TEST ; ELSE PRINT THE TEST
047404 013746 002114 MOV L$TEST,-(SP)
047410 012746 050026 MOV #BYPASS,-(SP)
047414 012746 000002 MOV #2,-(SP)
047420 010600 MOV SP,RO
047422 104417 TRAP C$PNTF
047424 062706 000006 ADD #6,SP
4122 047430 1$: EXIT TST ;BYPASSED MESSAGE AND GET OUT
047430 104432 TRAP C$EXIT
047432 000372 .WORD L10031-.
4123
4124 047434 042764 000010 000014 GO10: BIC #DUPFLG,LUNFLG(R4) ;CLEAR DUP FLAG
4125 047442 042764 000020 000014 BIC #ABTFLG,LUNFLG(R4) ;CLEAR ABORT FLAG
4126 047450 042764 000040 000014 BIC #CNTRLC,LUNFLG(R4) ;CLEAR CONTROL C FLAG
4127 047456 012737 032700 002310 MOV #CTRL,FRUIS ;SET UP FRU POINTER
4128 047464 004737 037102 JSR PC,PRINT ;GO DO A PORT INITIALIZE
4129 047470 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
4130 047476 001150 BNE TSTXEX ;GET OUT IF NOT AVAILABLE
4131 047500 012705 002344 MOV #SCTRLC,R5 ;SET UP TO DO THE SCC COMMAND
4132 047504 012737 000001 002272 MOV #1,I TRCNT ;DO IT ONCE
4133 047512 012737 032723 023044 MOV #SCCCMD,CURCMD ;SET UP COMMAND ASCII
4134 047520 004737 037334 JSR PC,CLSDRV ;GO ISSUE THE COMMAND
4135 047524 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
4136 047532 001132 BNE TSTXEX ;GET OUT IF NOT AVAILABLE
4137 047534 012705 002410 MOV #ONLINE,R5 ;SET UP TO DO THE ONLINE COMMAND
4138 047540 012737 000001 002272 MOV #1,I TRCNT ;DO IT ONCE
4139 047546 012737 032730 023044 MOV #ONLCMD,CURCMD ;SET UP COMMAND ASCII
4140 047554 016465 000006 000004 MOV MSCPUN(R4),P.UNIT(R5) ;PUT THE UNIT NUMBER IN THE PACKET
4141 047562 004737 037334 JSR PC,CLSDRV ;GO ISSUE THE COMMAND
4142 047566 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE

```

```

4143 047574 001111          BNE      TSTXEX          ;GET OUT IF NOT AVAILABLE
4144 047576 052764 000010 000014  BIS      #DUPFLG,LUNFLG(R4) ;ALL FOLLOWING COMMANDS ARE DUP
4145 047604 012705 002500          MOV      #XLOCP,R5        ;SET UP TO DO ELP COMMAND
4146 047610 012737 000001 002272  MOV      #1,ITRCNT        ;DO IT ONCE
4147 047616 012737 032761 023044  MOV      #ELPCMD,CURCMD   ;SET UP COMMAND ASCII
4148 047624 004737 037334          JSR      PC,CLSDRV        ;GO ISSUE THE COMMAND
4149 047630 032764 000001 000014  BIT      #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
4150 047636 001070          BNE      TSTXEX          ;GET OUT IF NOT AVAILABLE
4151 047640 032737 000200 177560 1$:  BIT      #BIT7,RCSR       ;CHECK FOR INPUT FROM KEYBOARD
4152 047646 001413          BEQ      2$              ;IF NONE, CONTINUE
4153 047650 013705 177562          MOV      RBUF,R5         ;GET DATA INPUT FROM KEYBOARD
4154 047654 042705 000200          BIC      #BIT7,R5        ;STRIP PARITY
4155 047660 022705 000003          CMP      #3,R5          ;CHECK FOR "CONTROL C" INPUT
4156 047664 001004          BNE      2$              ;IF NOT, CONTINUE
4157 047666 052764 000040 000014  BIS      #CNTRLC,LUNFLG(R4) ;SET "CONTROL C" FLAG
4158 047674 000432          BR       20$            ;GO ABORT MICRODIAGNOSTICS
4159 047676 012705 002526          MOV      #RCV DAT,R5     ;SET UP TO DO RECEIVE DATA COMMAND
4160 047702 012737 000001 002272 2$:  MOV      #1,ITRCNT        ;DO IT ONCE
4161 047710 012737 032766 023044  MOV      #RCV CMD,CURCMD  ;SET UP COMMAND ASCII
4162 047716 004737 037334          JSR      PC,CLSDRV        ;GO ISSUE THE COMMAND
4163 047722 032764 000001 000014  BIT      #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
4164 047730 001033          BNE      TSTXEX          ;GET OUT IF NOT AVAILABLE
4165 047732 032764 000020 000014  BIT      #ABTFLG,LUNFLG(R4) ;ABORT LOCAL PROGRAM ?
4166 047740 001010          BNE      20$            ;YES, ISSUE ABORT COMMAND
4167 047742 005002          CLR      R2             ;NO, SET UP DELAY LOW COUNT
4168 047744 012703 000020          MOV      #20,R3         ;SET UP DELAY HIGH COUNT
4169 047750 005202          INC      R2             ;DELAY
4170 047752 001376          BNE      10$            ;
4171 047754 005303          DEC      R3             ;
4172 047756 001374          BNE      10$            ;
4173 047760 000727          BR       1$              ;
4174 047762 012705 002556          MOV      #ABORT,R5       ;POLL DIAGL2 PROGRESS
4175 047766 012737 000001 002272 20$: MOV      #1,ITRCNT        ;SET UP TO DO ABORT COMMAND
4176 047774 012737 032773 023044  MOV      #ABT CMD,CURCMD  ;DO IT ONCE
4177 050002 004737 037334          JSR      PC,CLSDRV        ;SET UP COMMAND ASCII
4178 050006 032764 000040 000014  BIT      #CNTRLC,LUNFLG(R4) ;GO ISSUE THE COMMAND
4179 050014 001401          BEQ      TSTXEX          ;CHECK IF ABORTING DUE TO "CONTROL C"
4180 050016          BREAK          ;IF NOT, EXIT
4181 050020 104422          TRAP     C$BRK          ;
4182 050024          TSTXEX: EXIT          ;
4183 050026          TRAP     C$EXIT        ;
4184          .WORD  L10031- . ;
4185          ENDTST ;
4186          L10031: ;
4187          TRAP     C$ETST ;
4188          .ASCIZ  /%N%A TEST %Z3%A BYPASSED%N/ ;
4189          .EVEN ;
4190          ENDMOD ;
4191          .TITLE  PARAMETER CODING ;
4192          .SBTTL  HARDWARE PARAMETER CODING SECTION ;
4193          BGNMOD ;
4194          ;+ ;
4195          ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS ;
4196          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE ;
4197          ;
4198          ;
4199          ;
4200          ;
4201          ;
4202          ;
4203          ;
4204          ;
4205          ;
4206          ;
4207          ;
4208          ;
4209          ;
4210          ;
4211          ;
4212          ;
4213          ;
4214          ;
4215          ;
4216          ;
4217          ;
4218          ;
4219          ;
4220          ;
4221          ;
4222          ;
4223          ;
4224          ;
4225          ;
4226          ;
4227          ;
4228          ;
4229          ;
4230          ;
4231          ;

```

L9

PARAMETER CODING

MACRO V05.05 Wednesday 27-Feb-91 11:55 Page 56-2

SEQ 0115

HARDWARE PARAMETER CODING SECTION

```

4232 ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
4233 ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
4234 ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
4235 ; WITH THE OPERATOR.
4236 ; --
4237
4238 050062          BGNHRD
      050062 000044 .WORD L10032-L$HARD/2
      050064          L$HARD::

4239
4245
4246 050064          GPRMA   TKIPAD,0,0,160002,177564,YES
      050064 000031 .WORD   T$CODE
      050066 050122 .WORD   TKIPAD
      050070 160002 .WORD   T$LLOLIM
      050072 177564 .WORD   T$HILIM
4247 050074          GPRMD   TKVECT,2,0,777,60,776,YES
      050074 001032 .WORD   T$CODE
      050076 050137 .WORD   TKVECT
      050100 000777 .WORD   777
      050102 000060 .WORD   T$LLOLIM
      050104 000776 .WORD   T$HILIM
4248 050106          GPRMD   TKUNT,4,0,777,0,251,YES
      050106 002032 .WORD   T$CODE
      050110 050151 .WORD   TKUNT
      050112 000777 .WORD   777
      050114 000000 .WORD   T$LLOLIM
      050116 000251 .WORD   T$HILIM

4249
4250 050120          EXIT HRD
      050120 026004 .WORD   T$CODE

4251
4252 050122          124      113      111 TKIPAD: .ASCIZ  ?TKIP ADDRESS?
4253 050137          124      113      040 TKVECT: .ASCIZ  ?TK VECTOR?
4254 050151          124      057      115 TKUNT:  .ASCIZ  ?T/MSCP UNIT NUMBER?
4255                                     .EVEN
4256
4257
4258 050174          ENDHRD
      050174          .EVEN
      L10032:

4259
4266

```

M9

SOFTWARE PARAMETER CODING SECTION

4269
 4270
 4271
 4272
 4273
 4274
 4275
 4276
 4277
 4278
 4279
 4280 050174
 050174 000000
 050176
 4281
 4288
 4289
 4290
 4291 050176

 050176
 4292
 4293
 4303
 4304
 4305
 4306
 4307
 4308
 4309
 4310
 4311
 4312
 4313
 4314
 4315
 4316
 4317
 4318 060000
 4319
 4320
 4321 060000
 4322 060000
 4323 060000
 4324 060050
 4325 060050
 4326 062054
 4327 062054
 4331
 4332 062124

 062124 000000
 062126 000000
 062130
 4333 062130
 4334 000001

```
.SBTTL SOFTWARE PARAMETER CODING SECTION

; **
; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
; WITH THE OPERATOR.
; --

      BGNSFT
      .WORD L10033-L$SOFT/2
L$SOFT::

      .EVEN

      ENDSFT
      .EVEN

L10033:

; *****
; *****
; COMMUNICATIONS AREA
; THIS IS THE COMMUNICATIONS AREA THAT IS USED
; THROUGHOUT THE PROGRAM IN TESTING THE PERMUTATIONS
; OF THE UQ-PORT INIT SEQUENCE. IT IS ESSENTIAL THAT
; THIS AREA RESIDE IN AN 8KBYTE AREA OF MEMORY FREE
; OF DIAGNOSTIC CODE SO THAT IT MAY BE SUCCESSFULLY
; RELOCATED THROUGHOUT UPPER MEMORY VIA MEMORY MAN-
; AGEMENT.
; *****
; *****

      .=60000      ;START OF THE THIRD 8KBYTE BLOCK
                  ;OF VIRTUAL MEMORY SPACE. ACCESSIBLE
                  ;VIA PAR/PDR 2.

RDBUF::
COMMBF::
COMMAR::
LASTBF::

      LASTAD
      .EVEN
      .WORD 0
      .WORD 0
L$LAST::

      ENDMOD

      .END
```

N9

PARAMETER CODING
Symbol table

ABORT 002556
 ABTCMD 032773 G
 ABTFLG= 000020 G
 ADR = 000020 G
 ASSEMB= 000010
 BAKPAT 036620 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 023344
 BRFLAG= 000004 G
 BYPASS 050026 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 040146 G
 CHKCOM 036650 G
 CKCMEX 037000
 CLSDRV 037334 G
 CMARLG 002306 G

CMDCNT 003036 G
 CMDREF 003034 G
 CMDRNG 003016 G
 CMDSAV 023040 G
 CMMERR 002302 G
 CMPTBL 002260 G
 CMTBLG 002304 G
 CNTER = 000000 G
 CNTERR 023324
 CNTFLG 003030 G
 CNTHI 003026 G
 CNTRLC= 000040 G
 COMMAR 060050 G
 COMMBF 060000 G
 CONID = 177777 G
 CRD = 177776 G
 CTRL 032700 G
 CURCMD 023044 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\$ERRO= 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 023314
 DFPTBL 002202 G
 DIAGMC= 000000
 DRPFLG= 000001 G
 DRVE 032715 G
 DRIVER = 000011 G
 DSCEND 003026 G
 DSCRNG 003002 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 032761 G
 EMSG10 030520 G
 EMSG11 030546 G
 EMSG12 030607 G
 EMSG13 030656 G
 EMSG14 030705 G
 EMSG15 030734 G
 EMSG5 030342 G
 EMSG6 030363 G
 EMSG7 030414 G
 EMSG8 030435 G
 EMSG9 030473 G
 END 041214
 ENDCLE 041262
 ERR = 100000 G
 ERRBLK 023062 G
 ERRMSG 023060 G
 ERRNBR 023056 G
 ERRTP 023054 G
 EVL = 000004 G

EXIT 040762
 EXTINT 036130
 EXTVEC 036444
 E\$END = 002100
 E\$LOAD= 000035
 FLAG = 040000 G
 FLAGS 002336 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 032754 G
 GDUST 002460
 GO = 000001 G
 GOABO 040752
 GO10 047434
 GO2 042102
 GO3 042664
 GO4 043352
 GO5 044146
 GO6 044614
 GO7 045434
 GO8 046254
 GO9 047210
 G\$CNTO= 000200
 G\$DELM= 000372
 G\$DISP= 000003
 G\$EXCP= 000400
 G\$HILI= 000002
 G\$LOLI= 000001
 G\$NO = 000000
 G\$OFFS= 000400
 G\$OFSI= 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 023364
 HELP = 000000
 HIADDR= 000002 G
 HOE = 100000 G
 HSTIMO= 000000 G
 IBE = 010000 G
 IDU = 000040 G
 IER = 020000 G
 ILLINT 036132 G
 ILOOP 037132
 IMM = 000200 G
 IMSG 041220
 INISTP 002316 G
 INNER 002324 G
 INTFLG= 000002 G
 INTMMU 037002 G
 INTRCV 036122 G
 INTTBL 037324
 INVSTA 023334
 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

BIC

PARAMETER CODING
Symbol table

KPDR4 = 172310 G
 KPDR5 = 172312 G
 KPDR6 = 172314 G
 KPDR7 = 172316 G
 KTEXT 036312
 KTFLAG 002274 G
 KTTEST 036142 G
 LASTBF 062054 G
 LINE1 023426 G
 LINE2 023462 G
 LINE3 023542 G
 LINE4 023572 G
 LINE5 023634 G
 LINE6 023711 G
 LINE7 023754 G
 LOE = 040000 G
 LOGUNT 002312 G
 LOOP 037122
 LOT = 000010 G
 LUNBLK 002212 G
 LUNFLG= 000014 G
 L\$ACP 002110 G
 L\$APT 002036 G
 L\$AU 041316 G
 L\$AUT 002070 G
 L\$AUTO 041324 G
 L\$CCP 002106 G
 L\$CLEA 041252 G
 L\$CO 002032 G
 L\$DEPO 002011 G
 L\$DESC 002150 G
 L\$DESP 002076 G
 L\$DEVP 002060 G
 L\$DISP 002124 G
 L\$DLY 002116 G
 L\$DTP 002040 G
 L\$DTYP 002034 G
 L\$DU 041302 G
 L\$DUT 002072 G
 L\$DVTY 023414 G
 L\$EF 002052 G
 L\$ENVI 002044 G
 L\$ERRT 023054 G
 L\$ETP 002102 G
 L\$EXP1 002046 G
 L\$EXP4 002064 G
 L\$EXP5 002066 G
 L\$HARD 050064 G
 L\$HIME 002120 G
 L\$HPCP 002016 G
 L\$HPTP 002022 G
 L\$HW 002202 G
 L\$ICP 002104 G
 L\$INIT 040772 G
 L\$LADP 002026 G
 L\$LAST 062130 G
 L\$LOAD 002100 G

L\$LUN 002074 G
 L\$MREV 002050 G
 L\$NAME 002000 G
 L\$PRIO 002042 G
 L\$PROT 023046 G
 L\$PRT 002112 G
 L\$REPP 002062 G
 L\$REV 002010 G
 L\$RPT 035042 G
 L\$SOFT 050176 G
 L\$SPC 002056 G
 L\$SPCP 002020 G
 L\$SPTP 002024 G
 L\$STA 002030 G
 L\$SW 002212 G
 L\$TEST 002114 G
 L\$TIML 002014 G
 L\$UNIT 002012 G
 L10000 002210
 L10001 002212
 L10003 033602
 L10004 035040
 L10005 036112
 L10006 036120
 L10007 036130
 L10010 036140
 L10011 041250
 L10012 041300
 L10013 041314
 L10014 041322
 L10015 041324
 L10016 042022
 L10017 041512
 L10020 041756
 L10021 042604
 L10022 043272
 L10023 044066
 L10024 044534
 L10025 045354
 L10026 046174
 L10027 047130
 L10030 047354
 L10031 050024
 L10032 050174
 L10033 050176
 L2BRD1= 000024 G
 L2BRD2= 000030 G
 L2BWR1= 000014 G
 L2BWR2= 000020 G
 L2CMD = 000010 G
 L2CRC1= 000044 G
 L2CRC2= 000046 G
 L2DATA 040376 G
 L2DRV = 000004 G
 L2DUMP 033604 G
 L2ECC1= 000040 G
 L2ECC2= 000042 G

L2ELDE= 000155 G
 L2ELDF= 000156 G
 L2ELDS= 000166 G
 L2ELEC= 000140 G
 L2ELEV= 000136 G
 L2ELFL= 000135 G
 L2ELFM= 000134 G
 L2ELLB= 000162 G
 L2ELOF= 000172 G
 L2ELPB= 000160 G
 L2ELP1= 000146 G
 L2ELP2= 000150 G
 L2ELRL= 000144 G
 L2ELRT= 000145 G
 L2ELRW= 000170 G
 L2ELST= 000154 G
 L2ELTN= 000157 G
 L2ELT0= 000163 G
 L2ELT1= 000164 G
 L2ELT2= 000165 G
 L2ER1 030771 G
 L2ER10 031217 G
 L2ER11 031240 G
 L2ER12 031254 G
 L2ER13 031274 G
 L2ER14 031314 G
 L2ER15 031342 G
 L2ER16 031372 G
 L2ER17 031420 G
 L2ER18 031436 G
 L2ER19 031457 G
 L2ER2 031011 G
 L2ER20 031475 G
 L2ER21 031557 G
 L2ER22 031637 G
 L2ER23 031726 G
 L2ER24 032016 G
 L2ER25 032104 G
 L2ER26 032175 G
 L2ER27 032267 G
 L2ER3 031031 G
 L2ER4 031046 G
 L2ER5 031065 G
 L2ER6 031115 G
 L2ER7 031135 G
 L2ER8 031153 G
 L2ER9 031166 G
 L2ETBL 023064 G
 L2MSG 023314
 L2REP1= 000050 G
 L2REP2= 000052 G
 L2RSP = 000011 G
 L2STA = 000002 G
 L2SWR1= 000034 G
 L2SWR2= 000036 G
 L2TRK = 000007 G
 L2TST = 000006 G

L2UNT = 000012 G
 L21 024425 G
 L210 025101 G
 L211 025145 G
 L212 025211 G
 L213 025255 G
 L214 025321 G
 L215 025365 G
 L216 025431 G
 L219 025475 G
 L22 024501 G
 L220 025541 G
 L221 025607 G
 L222 025654 G
 L223 025716 G
 L224 025760 G
 L225 026017 G
 L226 026061 G
 L227 026123 G
 L228 026162 G
 L229 026221 G
 L23 024537 G
 L230 026263 G
 L231 026325 G
 L232 026367 G
 L233 026431 G
 L234 026470 G
 L235 026532 G
 L236 026574 G
 L237 026636 G
 L238 026700 G
 L239 026737 G
 L24 024577 G
 L240 026776 G
 L241 027035 G
 L242 027077 G
 L244 027136 G
 L245 027232 G
 L246 027326 G
 L247 027422 G
 L248 027516 G
 L249 027612 G
 L25 024637 G
 L250 027706 G
 L251 030002 G
 L252 030076 G
 L253 030172 G
 L254 030266 G
 L26 024677 G
 L27 024737 G
 L28 024777 G
 L29 025034 G
 MMON = 000001 G
 MMUSR0= 177572 G
 MMUSR1= 177574 G
 MMUSR2= 177576 G
 MMUSR3= 177516 G

MM22ON= 000020 G
 MSCPER 040672
 MSCPJN= 000006 G
 MSCPVR= 000000 G
 MSGLEN= 177774 G
 NEXT 041112
 NOKT 036306
 NUPASS 041076
 ONEFIL = 000001
 ONLCMD 032730 G
 ONLINE 002410
 OP.ABT= 000006 G
 OP.ELP= 000003 G
 OP.END= 000200 G
 OP.GDS= 000001 G
 OP.ONL = 000011 G
 OP.REC= 000005 G
 OP.SCC= 000004 G
 OUTER 002326 G
 OWIN = 100000 G
 O\$APTS= 000000
 O\$AU = 000000
 O\$BGNR= 000001
 O\$BGNS= 000000
 O\$DU = 000001
 O\$ERRT= 000001
 O\$GNSW= 000000
 O\$POIN= 000001
 O\$SETU= 000000
 PAROFF 002300 G
 PASCNT 002270 G
 PCKSIZ 003032 G
 PDELAY 036446 G
 PDLYEX 036472
 PDRECV 040254 G
 PNT = 001000 G
 PRI = 002000 G
 PRIDAT 033134 G
 PRIERR 033206 G
 PRIEX 033576
 PRIINI 033000 G
 PRIIP 033162 G
 PRIPAD 033062 G
 PRISA 033024 G
 PRIVAD 033106 G
 PRI00 = 000000 G
 PRI01 = 000040 G
 PRI02 = 000100 G
 PRI03 = 000140 G
 PRI04 = 000200 G
 PRI05 = 000240 G
 PRI06 = 000300 G
 PRI07 = 000340 G
 PRTDRV 040004 G
 PRTINT 037102 G
 P.BCNT= 000014 G
 P.BUFF= 000020 G

Symbol table

P.CRF = 000000 G	START 041044	T\$ERRN= 000033	T\$\$RPT= 010005	WRCMD 032742 G
P.EXT1= 000014 G	STEPST 002320 G	T\$EXCP= 000000	T\$\$SOF= 010033	WRCMDE 033316 G
P.EXT2= 000015 G	STEP1 036474 G	T\$FLAG= 000041	T\$\$SRV= 010010	WRDATA 002322 G
P.EXT3= 000016 G	STPTBL 002250 G	T\$GMAN= 000000	T\$\$SUB= 010020	WRER1 032355 G
P.FLGS= 000011 G	STPIER 036612	T\$HILI= 000251	T\$\$SW = 010001	WRER2 032407 G
P.IND1= 000020 G	STPIEX 036616	T\$LAST= 000001	T\$\$TES= 010031	WRER3 032430 G
P.IND2= 000022 G	SVCGBL= 000000	T\$LOLI= 000000	T1 041326 G	WRER4 032456 G
P.MOD = 000012 G	SVCINS= 000000	T\$LSYM= 010000	T1.1 041354	WRER5 032502 G
P.OPCD= 000010 G	SVCSUB= 000000	T\$LTNO= 000012	T1.2 041540	WRER6 032533 G
P.STS = 000012 G	SVCTAG= 000000	T\$NEST= 177777	T10 047356 G	WRER7 032554 G
P.UNIT= 000004 G	SVCTST= 000000	T\$NSO = 000000	T2 042024 G	WRER8 032622 G
RBUF = 177562 G	S\$LSYM= 010000	T\$NS1 = 000005	T2EXT 042600	WRER9 032646 G
RCSR = 177560 G	S1 = 004000 G	T\$NS2 = 000002	T3 042606 G	WREX 033552 G
RCVCMO 032766 G	TEMP 002332 G	T\$PTNU= 000000	T3EXT 043266	WRINTO 033236 G
RCVDAT 002526	TEMP1 002334 G	T\$SAVL= 177777	T4 043274 G	WRPRTE 033262 G
RDBUF 060000 G	TF.BLK= 000010 G	T\$SEGL= 177777	T4EXT 044056	WRSEGE 033450 G
RDCMD 032747 G	TIMOUT= 000024 G	T\$SUBN= 000000	T5 044070 G	WRTOE 033526 G
RESPBF 002572 G	TKIP = 000000 G	T\$TAGL= 177777	T5EXT 044524	WR1 024000 G
REWCMO 032735 G	TKIPAD 050122	T\$TAGN= 010034	T6 044536 G	WR2 024043 G
RNGSTP= 000004 G	TKIPSV= 000010 G	T\$TEMP= 000000	T6EXT 045350	WR3 024077 G
RSPADD 023354	TKSA = 000002 G	T\$TEST= 000012	T7 045356 G	WR4 024130 G
RSPBUF 002576 G	TKSASV= 000012 G	T\$TSTM= 177777	T7EXT 046170	WR5 024155 G
RSPEND 003006 G	TKUNT 050151	T\$TSTS= 000001	T8 046176 G	WR6 024217 G
RSPRNG 003006 G	TKVEC = 000004 G	T\$\$AU = 010014	T8EXT 047120	WR7 024222 G
RSPSAV 023042 G	TKVECT 050137	T\$\$AUT= 010015	T9 047132 G	WR8 024273 G
RSPSTP= 000104 G	TOUT 002330 G	T\$\$CLE= 010012	T9EXT 047342	WR9 024350 G
RSPTO 023404	TRAP4 036114 G	T\$\$DU = 010013	T9FLAG= 000100 G	W1 033362
RSTVEC 036326 G	TRP4FG 002276 G	T\$\$HAR= 010032	UAM = 000200 G	XLOCPR 002500
SAEXP 002314 G	TSTXEX 050020	T\$\$HW = 010000	UNERLG 023374	X\$ALWA= 000000
SCCCMD 032723 G	TXFER = 000005 G	T\$\$INI= 010011	VECTOR 036356 G	X\$FALS= 000040
SCTRLC 002344	T\$ARGC= 000002	T\$\$MSG= 010004	VEC4 = 000004 G	X\$OFFS= 000400
SFPTBL 002212 G	T\$CODE= 026004	T\$\$PRO= 010002	WRBUF 003040 G	X\$TRUE= 000020

. ABS. 062130 000 (RW,I,GBL,ABS,OVR)
 000000 001 (RW,I,LCL,REL,CON)

Errors detected: 0

*** Assembler statistics

Work file reads: 354
 Work file writes: 354
 Size of work file: 34208 Words (134 Pages)
 Size of core pool: 19136 Words (73 Pages)
 Operating system: RSX-11M/M-PLUS (Under VAX/VMS)

Elapsed time: 01:09:01.27
 CZTKAF.BIN,CZTKAF/-SP=SVC40R.MLB/ML,CZTKAF