

TK50

TK50 FRONT END FUNC  
CZTKAB0

AH-T772B-MC  
1 OF 1 OCT 1985  
COPYRIGHT © 1985

digital  
MADE IN USA

The image displays a grid of 100 small, illegible technical diagrams or data tables arranged in 10 rows and 10 columns on a dark background. Each cell in the grid contains a small, light-colored graphic or text that is too faint to read. The overall appearance is that of a technical manual or a data sheet for a specific system, likely related to the 'TK50 FRONT END FUNC' mentioned in the header.

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 8

IDENTIFICATION  
-----

PRODUCT CODE: AC - T771B MC  
PRODUCT NAME: CZTKABO TK50 FRONT END FUNCTIONAL  
PRODUCT DATE: 26 - JUL - 1985  
MAINTAINER: TAPE AND OPTICAL DIAGNOSTIC ENGINEERING  
AUTHOR: RAYMOND CHANG

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

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

COPYRIGHT (C) 1985 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

46  
47  
48  
49  
50  
51  
52  
53

REVISION HISTORY

APRIL 1985

NEW RELEASE

JUNE 1985

REVISION B  
ADDED CODE SO THAT PROGRAM CAN RUN ON  
PDP - 11 UNIBUS FAMILY CPUS.

55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111

## 1 GENERAL INFORMATION

### i.1 Product Description

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

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

### 1.2 Product Users And Uses

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

### 1.3 Performance Goals

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

### 1.4 Pass/Fail Criteria

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

112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165

#### 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 TK50 tape drives with controllers
7. 1 to 4 TK50 scratch cartridges (optional)
8. LCP-5 UFD software (optional)



167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216

## 2 USER INTERFACE

### 2.1 User Dialogue

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

#### 2.1.1 Hardware Questions -

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

CHANGE HARDWARE (L)? no default

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

UNIT x

BASE ADDRESS (O) 774500?

VECTOR (O) 224?

UNIT NUMBER (O)?

#### 2.1.2 Definition Of Hardware Questions -

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

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

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

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

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

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

#### 2.1.3 Software Questions -

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

218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274

### 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
```

```

275
276      **** LEVEL 2 MICRODIAGNOSTIC DUMP ****
277
278      Program Status      *****
279      Drive Status        ***
280      Test Number         ***
281      Track Number        ***
282      TMSCP Command       ***
283      TMSCP Response Flag ***
284      TMSCP Unit Flags    *****
285
286      Error Log (Tape Transfer Error)
287      Error Format         ***
288      Error Flags         ***
289      Event Code          *****
290      Error Retry Level   ***
291      Number of Reties    ***
292      Position (Low Order) *****
293      Position (High Order) *****
294      Controller Status   ***
295      Drive Error Code    ***
296      Drive Flags        ***
297      Track number        ***
298      Physical Block Number *****
299      Logical Block Number ***
300      Tape Count 0        ***
301      Tape Count 1        ***
302      Tape Count 2        ***
303      Drive State         *****
304      Read/Write State    *****
305      Operation Flags     *****
306
307      Blocks Written Channel 1 *****
308      Blocks Written Channel 2 *****
309      Blocks Read Channel 1 *****
310      Blocks Read Channel 2 *****
311      Soft Write Channel 1 *****
312      Soft Write Channel 2 *****
313      Ecc Corrected Channel 1 *****
314      Ecc Corrected Channel 2 *****
315      Read Repositions Channel 1 *****
316      Read Repositions Channel 2 *****
317
318
319
320
321

```

### 3.2.1 Program Status -

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

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

```

322
323
324
325
326
327
328
329
330
331

```



332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388

### 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).

389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445

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 "TK50 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.

446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502

### 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.

503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559

### 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.

560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616

#### 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.

617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648

#### 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.

650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706

## 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



707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763

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

#### 5.4 Test 4 - Vector And Interrupt Test

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

#### 5.5 Test 5 - BR Level Test

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

#### 5.6 Test 6 - Purge And Poll Test

The initialization sequence will again be started by the host, but in step 3, the host will set the "PP" (Purge and Poll) bit. The first part of this test will simulate the UQSSP handshake for a bus adapter purge. The host will then request the controller to commence "poll" testing. The controller will now begin DMA activity in both directions to the host-identified communication area. The controller must end this test by leaving the communication area cleared and by transitioning to step 4. The host will verify that the entire communication area is cleared. (Note: At the start of this test the host will have filled the communication area with a non-zero data pattern.) Ring depth in this test will be set to the minimal value (i.e., one), reducing potential impact on host memory in event of failure. Failures encountered will again be attributed to the controller.

#### 5.7 Test 7 - Maximum Ring Buffer Test

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

764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820

### 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

821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877

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

878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928

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

#### 5.11.6 Test 6 - Overwrite Test

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

#### 5.11.7 Test 7 - Track Access Test -

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

#### 5.11.8 Test 8 - Positioning test -

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

#### 5.12 Retry Algorithms

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

@

```

940      .TITLE PROGRAM HEADER AND TABLES
941      .SBTTL PROGRAM HEADER
967
969 000000      .ENABL ABS,AMA
970      002000      =          2000
971      .NLIST BEX
973
974 002000      BGNMOD
975
976      ;**
977      ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
978      ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
979      ;--
980
981 002000      POINTER BGNDU,ERRTBL,BGNRPT
982
990
991 002000      HEADER CZTKA,B,0,120.,0,PRI00
002000      L$NAME::      ;DIAGNOSTIC NAME
002000      103      .ASCII /C/
002001      132      .ASCII /Z/
002002      124      .ASCII /T/
002003      113      .ASCII /K/
002004      101      .ASCII /A/
002005      000      .BYTE 0
002006      000      .BYTE 0
002007      000      .BYTE 0
002010      L$REV::      ;REVISION LEVEL
002010      102      .ASCII /B/
002011      L$DEPO::      ;0
002011      060      .ASCII /O/
002012      L$UNIT::      ;NUMBER OF UNITS
002012      000000      .WORD 0
002014      L$TIML::      ;LONGEST TEST TIME
002014      000170      .WORD 120.
002016      L$HPCP::      ;PTR. TO H.W. QUES.
002016      047664      .WORD L$HARD
002020      L$SPCP::      ;PTR. TO S.W. QUES.
002020      000000      .WORD 0
002022      L$HPTP::      ;PTR. TO DEF. H.W. PTABLE
002022      002202      .WORD L$HW
002024      L$SPTP::      ;PTR. TO S.W. PTABLE
002024      000000      .WORD 0
002026      L$LADP::      ;DIAG. END ADDRESS
002026      062130      .WORD L$LAST
002030      L$STA::      ;RESERVED FOR APT STATS
002030      000000      .WORD 0
002032      L$CO::      .WORD 0
002032      000000      .WORD 0
002034      L$DTYP::      ;DIAGNOSTIC TYPE
002034      000000      .WORD 0
002036      L$APT::      ;APT EXPANSION
002036      000000      .WORD 0
002040      L$DTP::      ;PTR. TO DISPATCH TABLE
002040      002124      .WORD L$DISPATCH
002042      L$PRIO::      ;DIAGNOSTIC RUN PRIORITY
002042      000000      .WORD PRI00

```

002044		L\$ENVI::		;FLAGS DESCRIBE HOW IT WAS SETUP
002044	000000	.WORD	0	
002046		L\$EXP1::		;EXPANSION WORD
002046	000000	.WORD	0	
002050		L\$MREV::		;SVC REV AND EDIT #
002050	004	.BYTE	C\$REVISION	
002051	000	.BYTE	C\$EDIT	
002052		L\$EF::		;DIAG. EVENT FLAGS
002052	000000	.WORD	0	
002054	000000	.WORD	0	
002056		L\$SPC::		
002056	000000	.WORD	0	
002060		L\$DEVP::		; POINTER TO DEVICE TYPE LIST
002060	023412	.WORD	L\$DVTYP	
002062		L\$REPP::		;PTR. TO REPORT CODE
002062	035002	.WORD	L\$RPT	
002064		L\$EXP4::		
002064	000000	.WORD	0	
002066		L\$EXP5::		
002066	000000	.WORD	0	
002070		L\$AUT::		;PTR. TO ADD UNIT CODE
002070	000000	.WORD	0	
002072		L\$DUT::		;PTR. TO DROP UNIT CODE
002072	041130	.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	023052	.WORD	L\$ERRRTBL	
002104		L\$ICP::		;PTR. TO INIT CODE
002104	040620	.WORD	L\$INIT	
002106		L\$CCP::		;PTR. TO CLEAN-UP CODE
002106	041100	.WORD	L\$CLEAN	
002110		L\$ACP::		;PTR. TO AUTO CODE
002110	000000G	.WORD	L\$AUTO	
002112		L\$PRT::		;PTR. TO PROTECT TABLE
002112	023044	.WORD	L\$PROT	
002114		L\$TEST::		;TEST NUMBER
002114	000000	.WORD	0	
002116		L\$DLY::		;DELAY COUNT
002116	000000	.WORD	0	
002120		L\$HIME::		;PTR. TO HIGH MEM
002120	000000	.WORD	0	

999  
1000  
1001  
1002  
1003  
1004  
1005  
1006

.SBTTL DISPATCH TABLE

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

002122  
002122 000012  
002124  
002124 041152  
002126 041650  
002130 042432  
002132 043120  
002134 043714  
002136 044362  
002140 045202  
002142 046022  
002144 046746  
002146 047156

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

1007  
1014

1015 002150  
002150  
002150

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

103 132 124

1016



```
1018          .SBTTL  DEFAULT HARDWARE P-TABLE
1019
1020          ;**
1021          ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
1022          ; THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
1023          ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES.
1024          ;--
1025
1026 002200          BGNHW  DFPTBL
          002200 000003          .WORD  L10000 L$HW/2
          002202          L$HW::
          002202          DFPTBL::
1027
1033 002202 174500          .WORD  174500          ;TKIP BASE ADDRESS
1034 002204 000260          .WORD  260          ;VECTOR
1035 002206 000000          .WORD  0          ;T/MSCP UNIT NUMBER
1036 002210          ENDPW
          002210          L10000:
```

```
1039          .SBTTL  SOFTWARE P TABLE
1040
1041          ;**
1042          ; THE SOFTWARE P TABLE CONTAINS THE VALUES OF THE PROGRAM
1043          ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
1044          ;--
1045
1046 002210          BGNSW  SFPTBL
1047          002210 000000          .WORD  L10001 L$SW/2
1048          002212          L$SW::
1049          002212          SFPTBL::
1050
1051
1052
1053
1054
1055          002212          ENDSW
1056          002212          L10001:
1057          002212          ENDMOD
```

1060  
1071  
1072  
1100  
1101  
1102 002212  
1103  
1104  
1105  
1106  
1107  
1108  
1109 002212

.TITLE GLOBAL AREAS  
.SBTTL GLOBAL EQUATES SECTION

BGNMOD

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

EQUALS

; BIT DIFINITIONS

100000	BIT15== 100000
040000	BIT14== 40000
020000	BIT13== 20000
010000	BIT12== 10000
004000	BIT11== 4000
002000	BIT10== 2000
001000	BIT09== 1000
000400	BIT08== 400
000200	BIT07== 200
000100	BIT06== 100
000040	BIT05== 40
000020	BIT04== 20
000010	BIT03== 10
000004	BIT02== 4
000002	BIT01== 2
000001	BIT00== 1
001000	BIT9== BIT09
000400	BIT8== BIT08
000200	BIT7== BIT07
000100	BIT6== BIT06
000040	BIT5== BIT05
000020	BIT4== BIT04
000010	BIT3== BIT03
000004	BIT2== BIT02
000002	BIT1== BIT01
000001	BIT0== BIT00

; EVENT FLAG DEFINITIONS

; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

000040	EF.START== 32.	; BIT POSITION IN SECOND STATUS WORD
000037	EF.RESTART== 31.	; (100000) START COMMAND WAS ISSUED
000036	EF.CONTINUE== 30.	; (040000) RESTART COMMAND WAS ISSUED
000035	EF.NEW== 29.	; (020000) CONTINUE COMMAND WAS ISSUED
000034	EF.PWR== 28.	; (010000) A NEW PASS HAS BEEN STARTED
		; (004000) A POWER-FAIL/POWER-UP OCCURRED

; PRIORITY LEVEL DEFINITIONS

000340	PRI07== 340
000300	PRI06== 300
000240	PRI05== 240
000200	PRI04== 200
000140	PRI03== 140
000100	PRI02== 100
000040	PRIC1== 40
000000	PRIOC== 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

```
1114 ;:*****
1115 ;:*****
1116 ;
1117 ;LUN_BLOCK OFFSETS
1118 ; THESE LITERALS ARE USED AS WORD OFFSETS INTO THE LUNBLK, WHICH
1119 ; IS POINTED TO THROUGHOUT THE PROGRAM BY R4.
1120 ;
1121 ;:*****
1122 ;:*****
1123
1124 000000 TKIP == 0 ;TKIP REGISTER ADDRESS
1125 000002 TKSA == 2 ;TKSA REGISTER ADDRESS
1126 000004 TKVEC == 4 ;TK INTERRUPT VECTOR
1127 000006 MSCPUN == 6 ;T/MSCP UNIT NUMBER
1128 000010 TKIPSV == 10 ;SAVE LOCATION FOR IP CONTENTS
1129 000012 TKSASV == 12 ;SAVE LOCATION FOR SA CONTENTS
1130 000014 LUNFLG == 14 ;BIT-SPECIFIC MEANINGS AS DEFINED BELOW
1131
1132
1133 ;:*****
1134 ;:*****
1135 ;
1136 ;LUNFLG
1137 ; THIS WORD IN LUNBLK IS USED TO CONVEY VARIOUS INFORMATION
1138 ; IN A BIT-SPECIFIC MANNER. BITS USED BY THE PROGRAM ARE
1139 ; DEFINED AS FOLLOWS.
1140 ;
1141 ;:*****
1142 ;:*****
1143
1144 000001 DRPFLG == BIT0 ;=0 UUT AVAILABLE FOR TEST
1145 ;=1 UUT HAS BEEN DROPPED
1146 000002 INTFLG == BIT1 ;=1 EXPECTED INTERRUPT OCCURRED
1147
1148 000004 BRFLAG == BIT2 ;=1 INTERRUPT PRIORITY TEST
1149
1150 000010 DUPFLG == BIT3 ;=1 DUP COMMAND
1151
1152 000020 ABTFLG == BIT4 ;=1 ABORT LOCAL PROGRAM
1153
1154 000040 CNTRLC == BITS ;=1 RETURN TO DRS
1155
```

```

1157      ;:*****
1158      ;:*****
1159      ;
1160      ;UQ PORT EQUATES
1161      ;   THIS SECTION DEFINES THOSE LITERALS USED
1162      ;   BY THE DIAGNOSTIC IN THE UQ-PORT PROTOCOL.
1163      ;   IN GENERAL THEY HAVE BEEN FORMED BY USING
1164      ;   THE TWO LETTER MNEMONIC DEFINED IN UQSSP,
1165      ;   PRECEDED BY "B." INDICATING THEY ARE BITS.
1166      ;
1167      ;:*****
1168      ;:*****
1169
1170      ;READ-ONLY BITS
1171
1172      004000      B.S1    ==    BIT11    ;STEP 1
1173      010000      B.S2    ==    BIT12    ;STEP 2
1174      020000      B.S3    ==    BIT13    ;STEP 3
1175      040000      B.S4    ==    BIT14    ;STEP 4
1176
1177      100000      B.ER    ==    BIT15    ;ERROR INDICATION
1178      002000      B.NV    ==    BIT10    ;=0 VECTOR IS HOST SETTABLE
1179      001000      B.QB    ==    BIT9     ;=1 SUPPORTS 22 BIT HOST BUS
1180      000400      B.DI    ==    BIT8     ;=1 SUPPORTS ENHANCED DIAGNOSTICS
1181      000200      B.OD    ==    BIT7     ;=1 SUPPORTS ODD BUFFER ADDRESSES
1182      000100      B.MP    ==    BIT6     ;=1 SUPPORTS ADDRESS MAPPING
1183
1184      ;WRITE-ONLY BITS
1185
1186      100000      B.PP    ==    BIT15    ;PERFORM PURGE AND POLL TESTS
1187      040000      B.WR    ==    BIT14    ;ENTER DIAGNOSTIC WRAP MODE
1188      000002      B.LF    ==    BIT1     ;LAST FAIL REQUEST
1189      000001      B.PI    ==    BIT0     ;ENABLE ADAPTER PURGE INTERRUPTS
1190      000001      B.GO    ==    BIT0     ;GO BIT - START RUNNING
1191
1192      ;READ/WRITE BITS
1193
1194      000200      B.IE    ==    BIT7     ;STEP X-TION INTERRUPT ENABLE
1195
    
```

```
1197 ;:*****
1198 ;:*****
1199 ;
1200 ;GENERAL PURPOSE EQUATES
1201 ;
1202 ;:*****
1203 ;:*****
1204
1205 000004 VEC4 == 4 ;VECTOR FOUR - NXM TIMEOUTS, ETC.
1206 177560 RCSR == 177560 ;TERMINAL RECEIVE CONTROL/STATUS REGISTER ADDRESS
1207 177562 RBUF == 177562 ;TERMINAL RECEIVE BUFFER ADDRESS
1208
```



```
1210 ;:*****
1211 ;:*****
1212 ;
1213 ;MEMORY MANAGEMENT EQUATES
1214 ;
1215 ;:*****
1216 ;:*****
1217
1218 MMUSRO == 177572 ;STATUS REG 0
1219 MMUSR1 == 177574
1220 MMUSR2 == 177576
1221 MMUSR3 == 172516 ;SHOULD ONLY BE PRESENT ON 22 BIT CPU'S
1222
1223 KPAR0 == 172340 ;KERNEL MODE PAGE ADDRESS REG 0
1224 KPAR1 == 172342
1225 KPAR2 == 172344
1226 KPAR3 == 172346
1227 KPAR4 == 172350
1228 KPAR5 == 172352
1229 KPAR6 == 172354
1230 KPAR7 == 172356 ;ALWAYS FOR I/O PAGE
1231
1232 KPDR0 == 172300 ;KERNEL MODE PAGE DESCRIPTOR REG 0
1233 KPDR1 == 172302
1234 KPDR2 == 172304
1235 KPDR3 == 172306
1236 KPDR4 == 172310
1237 KPDR5 == 172312
1238 KPDR6 == 172314
1239 KPDR7 == 172316
1240
1241 MMON == 000001
1242 MM22ON == 000020
1243
```

```

1245      ;:*****
1246      ;:*****
1247      ;
1248      ;COMMAND PACKET OPCODES
1249      ;
1250      ;:*****
1251      ;:*****
1252
1253      000004      OP.SCC  ==      04      ;SET CONTROLLER CHARACTERISTICS OPCODE
1254      000011      OP.ONL  ==      11      ;ONLINE OPCODE
1255      000001      OP.GDS  ==      01      ;GET DUST STATUS OPCODE
1256      000003      OP.FLP  ==      03      ;EXECUTE LOCAL PROGRAM OPCODE
1257      000005      OP.REC  ==      05      ;RECEIVE DATA OPCODE
1258      000006      OP.ABT  ==      06      ;ABORT PROGRAM OPCODE
1259      000200      OP.END  ==      200     ;END MESSAGE FLAG OPCODE
1260
1261
1262      ;:*****
1263      ;:*****
1264      ;
1265      ;GENERIC COMMAND PACKET OFFSETS
1266      ;
1267      ;:*****
1268      ;:*****
1269
1270      000000      P.CRF   ==      0      ;COMMAND REFERENCE NUMBER
1271      000004      P.UNIT  ==      4      ;UNIT NUMBER
1272      000010      P.OPCD  ==      10     ;OPCODE
1273      000011      P.FLGS  ==      11     ;END MESSAGE FLAGS
1274      000012      P.MOD   ==      12     ;MODIFIERS / STATUS
1275      000012      P.STS   ==      12     ;MODIFIERS / STATUS
1276      000014      P.BCNT  ==      14     ;BYTE COUNT
1277      000020      P.BUFF  ==      20     ;BUFFER DESCRIPTOR
1278
1279
1280      ;:*****
1281      ;:*****
1282      ;
1283      ;GET DUST STATUS END PACKET OFFSETS
1284      ;
1285      ;:*****
1286      ;:*****
1287
1288      000014      P.EXT1  ==      14     ;1ST BYTE OF PROGRAM EXTENSION
1289      000015      P.EXT2  ==      15     ;2ND BYTE OF PROGRAM EXTENSION
1290      000016      P.EXT3  ==      16     ;3RD BYTE OF PROGRAM EXTENSION
1291      000017      DUSTFL  ==      17     ;FLAGS
1292      000020      P.IND1  ==      20     ;1ST WORD OF PROGRESS INDICATOR
1293      000022      P.IND2  ==      22     ;2ND WORD OF PROGRESS INDICATOR
1294      000024      TIMEOUT ==      24     ;TIMEOUT VALUE
1295
    
```

```
1297 ;:*****
1298 ;:*****
1299 ;
1300 ;TKSA BIT DEFINITIONS
1301 ;
1302 ;:*****
1303 ;:*****
1304 ;
1305 100000 ERR == 100000 ;ERROR
1306 004000 S1 == 004000 ;STEP 1
1307 000001 GO == 000001 ;GO
1308 ;
1309 ;
1310 ;:*****
1311 ;:*****
1312 ;
1313 ;U/Q PORT LITERALS
1314 ;
1315 ;:*****
1316 ;:*****
1317 ;
1318 100000 OWN == 100000 ;DESCRIPTOR OWNERSHIP BIT
1319 040000 FLAG == 040000 ;DESCRIPTOR INTERRUPT FLAG BIT
1320 000200 IMM == 000200 ;IMMEDIATE COMMAND FLAG
1321 000010 TF.BLK == 10 ;TAPE FORMAT
1322 000000 HSTIMO == 0 ;HOST TIMEOUT VALUE
1323 000000 MSCPVR == 0 ;MSCP VERSION NUMBER
1324 000004 RNGSTP == 4. ;DESCRIPTOR RING STEP
1325 000104 RSPSTP == 68. ;RESPONCE BUFFER STEP
1326 ;
1327 ;
```

```
1329 ;*****
1330 ;*****
1331 ;
1332 ;TMSCP DRIVER BUFFER OFFSETS
1333 ;
1334 ;*****
1335 ;*****
1336
1337 000002 HIADDR == 2. ;DESCRIPTOR ADDRESS OFFSET
1338 177777 CONID == -1. ;COMMAND/RESPONSE CONNECTION TYPE I.D.
1339 177776 CRD == 2. ;COMMAND/RESPONSE CREDIT LIMIT OFFSET
1340 177774 MSGLEN == -4. ;COMMAND/RESPONSE MESSAGE LENGTH
1341 000005 TXFER == 5. ;ERROR FORMAT FOR "TAPE TRANSFER" ERROR LOG
1342 000011 DRVER == 9. ;ERROR FORMAT FOR "DRIVE ERROR" ERROR LOG
1343 000000 CNTER == 0. ;ERROR FORMAT FOR "CONTROLLER ERROR" ERROR LOG
1344
```

```

1346      ;;*****
1347      ;;*****
1348      :
1349      :OFFSET VALUES INTO LEVEL 2 MICRODIAGNOSTIC RECEIVE DATA BUFFER
1350      :
1351      ;;*****
1352      ;;*****
1353
1354      000002      L2STA ==      2      ;OFFSET OF PROGRAM STATUS
1355      000004      L2DRV ==      4      ;OFFSET OF DRIVE STATUS
1356      000006      L2TST ==      6      ;OFFSET OF TEST NUMBER
1357      000007      L2TRK ==      7      ;OFFSET OF TRACK NUMBER
1358      000010      L2CMD ==     10      ;OFFSET OF COMMAND IN ERROR
1359      000011      L2RSP ==     11      ;OFFSET OF RESPONSE FLAG
1360      000012      L2UNT ==     12      ;OFFSET OF UNIT FLAGS
1361      000014      L2BWR1 ==    14      ;OFFSET OF BLOCKS WRITTEN CH. 1
1362      000020      L2BWR2 ==    20      ;OFFSET OF BLOCKS WRITTEN CH. 2
1363      000024      L2BRD1 ==    24      ;OFFSET OF BLOCKS READ CH. 1
1364      000030      L2BRD2 ==    30      ;OFFSET OF BLOCKS READ CH. 2
1365      000034      L2SWR1 ==    34      ;OFFSET OF SOFT WRITE ERRORS CH. 1
1366      000036      L2SWR2 ==    36      ;OFFSET OF SOFT WRITE ERRORS CH. 2
1367      000040      L2ECC1 ==    40      ;OFFSET OF ECC CORRECTED ERRORS CH. 1
1368      000042      L2ECC2 ==    42      ;OFFSET OF ECC CORRECTED ERRORS CH. 2
1369      000044      L2CRC1 ==    44      ;OFFSET OF CRC ERRORS ON F.C BLOCK CH. 1
1370      000046      L2CRC2 ==    46      ;OFFSET OF CRC ERRORS ON C.C BLOCK CH. 2
1371      000050      L2REP1 ==    50      ;OFFSET OF READ REPOSITIONS CH. 1
1372      000052      L2REP2 ==    52      ;OFFSET OF READ REPOSITIONS CH. 2
1373      000134      L2ELFM ==   134      ;OFFSET OF ERROR LOG ERROR FORMAT
1374      000135      L2ELFL ==   135      ;OFFSET OF ERROR LOG FLAGS
1375      000136      L2ELEV ==   136      ;OFFSET OF ERROR LOG EVENT CODE
1376      000144      L2ELRL ==   144      ;OFFSET OF ERROR LOG RETRY LEVEL
1377      000145      L2ELRT ==   145      ;OFFSET OF ERROR LOG NUMBER OF RETRIES
1378      000146      L2ELP1 ==   146      ;OFFSET OF ERROR LOG POSITION (low order)
1379      000150      L2ELP2 ==   150      ;OFFSET OF ERROR LOG POSITION (high order)
1380      000154      L2ELST ==   154      ;OFFSET OF CONTROLLER STATUS
1381      000155      L2ELDE ==   155      ;OFFSET OF ERROR LOG DRIVE ERROR CODE
1382      000156      L2ELDF ==   156      ;OFFSET OF ERROR LOG DRIVE FLAGS
1383      000157      L2ELTN ==   157      ;OFFSET OF ERROR LOG TRACK NUMBER
1384      000160      L2ELPB ==   160      ;OFFSET OF ERROR LOG PHYSICAL BLOCK NUMBER
1385      000162      L2ELLB ==   162      ;OFFSET OF ERROR LOG LOGICAL BLOCK NUMBER
1386      000163      L2ELT0 ==   163      ;OFFSET OF ERROR LOG LSB OF TAPE LENGTH COUNT
1387      000164      L2ELT1 ==   164      ;OFFSET OF ERROR LOG BYTE 1 OF TAPE LENGTH COUNT
1388      000165      L2ELT2 ==   165      ;OFFSET OF ERROR LOG MSB OF TAPE LENGTH COUNT
1389      000166      L2ELDS ==   166      ;OFFSET OF ERROR LOG DRIVE STATE
1390      000170      L2ELRW ==   170      ;OFFSET OF ERROR LOG READ/WRITE STATE
1391      000172      L2ELOF ==   172      ;OFFSET OF ERROR LOG OPERATION FLAGS
1392      000140      L2ELEC ==   140      ;OFFSET OF ERROR LOG CONTROLLER ERROR CODE
    
```

1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420 002212 000000 000000 000000  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436 002250  
1437  
1438 002260  
1439

.SBTTL GLOBAL DATA SECTION

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

\*\*\*\*\*  
\*\*\*\*\*

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

\*\*\*\*\*  
\*\*\*\*\*

LUNBLK:: .WORD 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

\*\*\*\*\*  
\*\*\*\*\*

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

\*\*\*\*\*  
\*\*\*\*\*

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

CMPTBL:: .BLKW 4 ;COMPARISON VALUES

```
1441 ;:*****
1442 ;:*****
1443 ;
1444 ;PROGRAM CONTROL VARIABLES
1445 ; THESE GLOBAL VARIABLES ARE GENERALLY USED TO CONTROL THE
1446 ; OVERALL EXECUTION OF THE DIAGNOSTIC.
1447 ;
1448 ;:*****
1449 ;:*****
1450
1451 002270 000000 PASCNT:: .WORD 0 ;CUMULATIVE PROGRAM PASS COUNTER
1452 002272 000000 ITRCNT:: .WORD 0 ;LOADED BY EACH TEST TO SPECIFY # OF ITERATIONS
1453 002274 000000 KTFLAG:: .WORD 0 ;=0 MEMORY MANAGEMENT NOT AVAILABLE
1454 ; ;=1 MEMORY MANAGEMENT IS AVAILABLE
1455 002276 000000 TRP4FG:: .WORD 0 ;=1 TRAP TO VECTOR OCCURRED
1456 002300 000000 PAROFF:: .WORD 0 ;USED IN TEST 7 TO STEP THROUGH UPPER MEMORY
1457 002302 000000 CMMERR:: .WORD 0 ;=0 NO ERROR IN COMMUNICATION AREA
1458 ; ;=1 ERROR WITHIN COMMUNICATION AREA
1459 ; ;=-1 ERROR BEYOND BOUNDS OF COMM AREA
1460 002304 000000 CMTBLG:: .WORD 0 ;# OF CONTIGUOUS WORDS IN ERROR IN COMM AREA
1461 002306 000000 CMARLG:: .WORD 0 ;LENGTH OF COMM AREA FOR TEST N
1462 002310 000000 FRUIS:: .WORD 0 ;POINTER TO FAULTY FRU ASCII FOR PRINTOUT
1463 002312 000000 LOGUNT:: .WORD 0 ;LOGICAL UNIT # OF CURRENT UUT
1464 002314 000000 SAEXP:: .WORD 0 ;LOADED WITH EXPECTED SA FOR ERROR CHECKING
1465 002316 000000 INISTP:: .WORD 0 ;CURRENT STEP OF INIT SEQUENCE
1466 002320 000000 STEPST:: .WORD 0 ;SUCCESS/FAIL STATUS FROM STEP SUBROUTINES
1467 002322 000000 WRDATA:: .WORD 0 ;LOADED WITH DATA FRO WRAP MODE TEST
1468 002324 000000 INNER:: .WORD 0 ;COUNTER FOR PDELAY ROUTINE
1469 002326 000000 OUTER:: .WORD 0 ;OTHER COUNTER FOR PDELAY
1470 002330 000000 TOUT:: .WORD 0 ;TIMEOUT INDICATOR FOR PDELAY
1471 002332 000000 TEMP:: .WORD 0 ;TEMPORARY STORAGE LOCATION
1472 002334 000000 FLAGS:: .WORD 0 ;WORK LOCATION FOR SUPERVISOR FLAGS
```

```
1474 ;*****
1475 ;*****
1476 ;
1477 ;TMSCP/DUP COMMAND PACKETS
1478 ;
1479 ;*****
1480 ;*****
1481 ;
1482 ;*****
1483 ;
1484 ;TMSCP SET CONTROLLER CHARACTERISTICS COMMAND PACKET
1485 ;
1486 ;*****
1487
1488 002336 000040 000000          .WORD 32.,0
1489 002342 000000 000000 SCTRLC: .WORD 0,0
1490 002346 000000 000000          .WORD 0,0
1491 002352 000004 000000          .WORD OP.SCC,0
1492 002356 000000 000000          .WORD 0,0
1493 002362 000000 000000          .WORD 0,0
1494 002366 000000 000000          .WORD 0,0
1495 002372 000000 000000          .WORD 0,0
1496 002376 000000 000000          .WORD 0,0
1497
1498
1499 ;STARS
1500 ;
1501 ;TMSCP ONLINE COMMAND PACKET
1502 ;
1503 ;*****
1504
1505 002402 000044 000000          .WORD 36.,0
1506 002406 000000 000000 ONLINE: .WORD 0,0
1507 002412 000000 000000          .WORD 0,0
1508 002416 000011 000000          .WORD OP.O'IL,0
1509 002422 000000 000000          .WORD 0,0
1510 002426 000000 000000          .WORD 0,0
1511 002432 000000 000000          .WORD 0,0
1512 002436 000000 000000          .WORD 0,0
1513 002442 000000 000000          .WORD 0,0
1514 002446 000000 000000          .WORD 0,0
1515
1516
1517 ;STARS
1518 ;
1519 ;DUP GET DUST STATUS COMMAND PACKET
1520 ;
1521 ;*****
1522
1523 002452 000020 000000          .WORD 16.,0
1524 002456 000000 000000 GDUST: .WORD 0,0
1525 002462 000000 000000          .WORD 0,0
1526 002466 000001 000000          .WORD OP.GDS,0
1527
1528
1529 ;STARS
1530 ;
```



```
1531 ;DUP EXECUTE LOCAL PROGRAM COMMAND PACKET
1532 ;
1533 ;:*****
1534
1535 002472 000022 000000 XLOCPR: .WORD 18.,0
1536 002476 000000 000000 .WORD 0,0
1537 002502 000000 000000 .WORD 0,0
1538 002506 000003 000000 .WORD OP.ELP,0
1539 002512 104 111 101 .ASCII /DIAGL2/
1540
1541
1542 ;STARS
1543 ;
1544 ;DUP RECEIVE DATA COMMAND PACKET
1545 ;
1546 ;:*****
1547
1548 002520 000024 000000 RCVDAT: .WORD 20.,0
1549 002524 000000 000000 .WORD 0,0
1550 002530 000000 000000 .WORD 0,0
1551 002534 000005 000000 .WORD OP.REC,0
1552 002540 000226 000000 .WORD 150.,0
1553 002544 060000 000000 .WORD RDBUF,0
1554
1555
1556 ;STARS
1557 ;
1558 ;DUP ABORT COMMAND PACKET
1559 ;
1560 ;:*****
1561
1562 002550 000014 000000 ABORT: .WORD 12.,0
1563 002554 000000 000000 .WORD 0,0
1564 002560 000000 000000 .WORD 0,0
1565 002564 000006 000000 .WORD OP.ABT,0
1566
```

```

1568 ;*****
1569 ;*****
1570 ;
1571 ;CLASS DRIVER BUFFERS
1572 ;
1573 ;*****
1574 ;*****
1575 ;
1576 002570 RESPBF:: .BLKW 2. ;TOP 4 LOCATIONS OF RESPONSE BUFFER
1577 002574 RSPBUF:: .BLKW 66. ;DRIVER RESPONSE BUFFER
1578 ;
1579 ;
1580 ;*****
1581 ;*****
1582 ;
1583 ;U/Q PORT DESCRIPTOR RINGS
1584 ;
1585 ;*****
1586 ;*****
1587 ;
1588 003000 DSCRNG:: .BLKW 2. ;DESCRIPTOR RING
1589 003004 RSPEND:: ;END OF RESPONSE BUFFER
1590 003004 RSPRNG:: .BLKW 4. ;RESPONSE DESCRIPTOR RING
1591 003014 CMDRNG:: .BLKW 4. ;COMMAND DESCRIPTOR RING
1592 003024 DSCEND:: ;END OF DESCRIPTOR RING
1593 ;
1594 ;
1595 ;*****
1596 ;*****
1597 ;
1598 ;CLASS AND PORT DRIVER VARIABLES
1599 ;
1600 ;*****
1601 ;*****
1602 ;
1603 003024 000000 CNTHI:: .WORD 0 ;VALUE OF THE HIGH TIMEOUT
1604 003026 000000 CNTFLG:: .WORD 0 ;CONTROLLER FLAGS
1605 003030 000000 PCKSIZ:: .WORD 0 ;PACKET SIZE IN BYTES
1606 003032 000000 CMDREF:: .WORD 0 ;COMMAND REFERENCE NUMBER
1607 003034 000000 CMDCNT:: .WORD 0 ;COMMAND COUNT
1608 003036 WRBUF:: .BLKW 4096. ;WRITE BUFFER
1609 023036 000000 CMDSAV:: .WORD 0 ;COMMAND DESCRIPTOR SAVE
1610 023040 000000 RSPSAV:: .WORD 0 ;RESPONSE DESCRIPTOR SAVE
1611 023042 000000 CURCMD:: .WORD 0 ;POINTER TO CURRENT COMMAND ASCII
1612 ;
    
```

```
1614 ;:*****
1615 ;:*****
1616 ;
1617 ;PROTECTION TABLE
1618 ;
1619 ;:*****
1620 ;:*****
1624
1625 023044 BGNPROT
      023044 L$PROT::
1626 023044 000000 .WORD 0
1627 023046 177777 .WORD -1
1628 023050 177777 .WORD -1
1629
1630 023052 ENDPROT
1631
```

```

1633 023052          STARS
1634 023052          ;:*****
                    STARS
                    ;:*****
1635                :
1636                ;LEVEL 2 MICRODIAGNOSTIC ERROR TABLE
1637                :
1638 023052          STARS
                    ;:*****
1639 023052          STARS
                    ;:*****

1640
1641 023052          ERRTBL
                    L$ERRTBL::
                    ERRTP::      .WORD  0
                    ERRNBR::     .WORD  0
                    ERRMSG::     .WORD  0
                    ERRLK::      .WORD  0

1642
1643 023062          L2ETBL::          ;START OF TMSCP COMMAND ERROR MESSAGES
1644
1645 023062 000001   .WORD  1.          ;INVALID COMMAND
1646 023064 000145   .WORD 101.
1647 023066 030763   .WORD L2ER1
1648 023070 033544   .WORD L2DUMP
1649
1650 023072 000001   .WORD  1.          ;COMMAND ABORTED
1651 023074 000146   .WORD 102.
1652 023076 031003   .WORD L2ER2
1653 023100 033544   .WORD L2DUMP
1654
1655 023102 000001   .WORD  1.          ;UNIT-OFFLINE
1656 023104 000147   .WORD 103.
1657 023106 031023   .WORD L2ER3
1658 023110 033544   .WORD L2DUMP
1659
1660 023112 000001   .WORD  1.          ;UNIT-AVAILABLE
1661 023114 000150   .WORD 104.
1662 023116 031040   .WORD L2ER4
1663 023120 033544   .WORD L2DUMP
1664
1665 023122 000001   .WORD  1.          ;INVALID STATUS
1666 023124 000151   .WORD 105.
1667 023126 031057   .WORD L2ER5
1668 023130 033544   .WORD L2DUMP
1669
1670 023132 000001   .WORD  1.          ;WRITE PROTECTED
1671 023134 000152   .WORD 106.
1672 023136 031107   .WORD L2ER6
1673 023140 033544   .WORD L2DUMP
1674
1675 023142 000001   .WORD  1.          ;COMPARE ERROR
1676 023144 000153   .WORD 107.
1677 023146 031127   .WORD L2ER7
1678 023150 033544   .WORD L2DUMP
1679
1680 023152 000001   .WORD  1.          ;DATA ERROR
    
```

1681	023154	000154	.WORD	108.	
1682	023156	031145	.WORD	L2ER8	
1683	023160	033544	.WORD	L2DUMP	
1684					
1685	023162	000001	.WORD	1.	;HOST BUFFER ACCESS ERROR
1686	023164	000155	.WORD	109.	
1687	023166	031160	.WORD	L2ER9	
1688	023170	033544	.WORD	L2DUMP	
1689					
1690	023172	000001	.WORD	1.	;CONTROLLER ERROR
1691	023174	000156	.WORD	110.	
1692	023176	031211	.WORD	L2ER10	
1693	023200	033544	.WORD	L2DUMP	
1694					
1695	023202	000001	.WORD	1.	;DRIVE ERROR
1696	023204	000157	.WORD	111.	
1697	023206	031232	.WORD	L2ER11	
1698	023210	033544	.WORD	L2DUMP	
1699					
1700	023212	000001	.WORD	1.	;FORMATTER ERROR
1701	023214	000160	.WORD	112.	
1702	023216	031246	.WORD	L2ER12	
1703	023220	033544	.WORD	L2DUMP	
1704					
1705	023222	000001	.WORD	1.	;BOT ENCOUNTERED
1706	023224	000161	.WORD	113.	
1707	023226	031266	.WORD	L2ER13	
1708	023230	033544	.WORD	L2DUMP	
1709					
1710	023232	000001	.WORD	1.	;TAPE MARK ENCOUNTERED
1711	023234	000162	.WORD	114.	
1712	023236	031306	.WORD	L2ER14	
1713	023240	033544	.WORD	L2DUMP	
1714					
1715	023242	000001	.WORD	1.	;INVALID STATUS
1716	023244	000163	.WORD	115.	
1717	023246	031334	.WORD	L2ER15	
1718	023250	033544	.WORD	L2DUMP	
1719					
1720	023252	000001	.WORD	1.	;RECORD DATA TRUNCATED
1721	023254	000164	.WORD	116.	
1722	023256	031364	.WORD	L2ER16	
1723	023260	033544	.WORD	L2DUMP	
1724					
1725	023262	000001	.WORD	1.	;POSITION LOST
1726	023264	000165	.WORD	117.	
1727	023266	031412	.WORD	L2ER17	
1728	023270	033544	.WORD	L2DUMP	
1729					
1730	023272	000001	.WORD	1.	;SERIOUS EXCEPTION
1731	023274	000166	.WORD	118.	
1732	023276	031430	.WORD	L2ER18	
1733	023300	033544	.WORD	L2DUMP	
1734					
1735	023302	000001	.WORD	1.	;LEOT DETECTED
1736	023304	000167	.WORD	119.	
1737	023306	031451	.WORD	L2ER19	

1738 023310 033544	.WORD	L2DUMP	
1739			
1740 023312	L2MSG:		;START OF MICRODIAGNOSTIC DETECTED ERROR MESSAGES
1741			
1742 023312 000001	DCERR: .WORD	1.	;DATA COMPARE ERROR
1743 023314 000170	.WORD	120.	
1744 023316 031467	.WORD	L2ER20	
1745 023320 033544	.WORD	L2DUMP	
1746			
1747 023322 000001	CNTERR: .WORD	1.	;CONTROLLER ERROR
1748 023324 000171	.WORD	121.	
1749 023326 031551	.WORD	L2ER21	
1750 023330 033544	.WORD	L2DUMP	
1751			
1752 023332 000001	INVSTA: .WORD	1.	;INVALID STATUS
1753 023334 000172	.WORD	122.	
1754 023336 031631	.WORD	L2ER22	
1755 023340 033544	.WORD	L2DUMP	
1756			
1757 023342 000001	BPERR: .WORD	1.	;BAD PATTERN NUMBER
1758 023344 000173	.WORD	123.	
1759 023346 031720	.WORD	L2ER23	
1760 023350 033544	.WORD	L2DUMP	
1761			
1762 023352 000001	RSPADD: .WORD	1.	;RESPONSE ADDRESS ERROR
1763 023354 000174	.WORD	124.	
1764 023356 032010	.WORD	L2ER24	
1765 023360 033544	.WORD	L2DUMP	
1766			
1767 023362 000001	HBFAADD: .WORD	1.	;HOST BUFFER ADDRESS ERROR
1768 023364 000175	.WORD	125.	
1769 023366 032076	.WORD	L2ER25	
1770 023370 033544	.WORD	L2DUMP	
1771			
1772 023372 000001	UNERLG: .WORD	1.	;UNKNOWN ERROR LOG RECEIVED
1773 023374 000176	.WORD	126.	
1774 023376 032167	.WORD	L2ER26	
1775 023400 033544	.WORD	L2DUMP	
1776			
1777 023402 000001	RSPTO: .WORD	1.	;RESPONSE TIMEOUT ERROR
1778 023404 000177	.WORD	127.	
1779 023406 032261	.WORD	L2ER27	
1780 023410 033544	.WORD	L2DUMP	

1782  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1805  
1806  
023412  
023412  
023412  
1807

124 113 065

```
.SBTTI GLOBAL TEXT SECTION  
;*****  
;*****  
; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,  
; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN  
; MORE THAN ONE TEST.  
;*****  
;*****  
;*****  
;*****  
; NAMES OF DEVICES SUPPORTED BY PROGRAM  
;*****  
L$DVTYP: DEVTYP <TK50>  
      .ASCIZ *TK50*  
      .EVEN
```

```

1812
1813
1814
1815
1816
1817
1818
1819 023420 045 101 111 LINE1:: .ASCIZ ?%AINIT SEQUENCE STEP #: %D1?
1820 023454 045 116 045 LINE2:: .ASCIZ ?%N%ASA REG: %06%A EXPCTD: %06%A ACTUAL SA: %06?
1821 023534 045 116 045 LINE3:: .ASCIZ ?%N%AIP REG ADDRESS: %06?
1822 023564 045 116 045 LINE4:: .ASCIZ ?%N%A****FAILING FRU: %T%A****%N%N?
1823 023626 045 101 122 LINE5:: .ASCIZ ?%ARELOCATION CONSTANT: %06%A VIRT. ADD: %06?
1824 023703 045 116 045 LINE6:: .ASCIZ ?%N%AEXPECTED: %06%A RECEIVED: %06?
1825 023746 045 101 120 LINE7:: .ASCIZ ?%APHYSICAL ADD: %06?
1826
1827 023772 045 116 045 WR1:: .ASCIZ ?%N%ASA REG: %06%A SA CONTENTS: %06?
1828 024035 045 116 045 WR2:: .ASCIZ ?%N%AFAILING COMMAND : %T%N?
1829 024071 045 116 045 WR3:: .ASCIZ ?%N%A RESPONSE PACKET%N?
1830 024122 045 116 045 WR4:: .ASCIZ ?%N%A %06%A %06?
1831 024147 045 116 045 WR5:: .ASCIZ ?%N%ACMD EXP: %06%A CMD REC: %06?
1832 024211 045 116 000 WR6:: .ASCIZ ?%N?
1833 024214 045 116 045 WR7:: .ASCIZ ?%N%AFAILING COMMAND : %T%N%AUNIT UNKNOWN?
1834 024265 045 116 045 WR8:: .ASCIZ ?%N%AFAILING COMMAND : %T%N%AANO MEDIA MOUNTED?
1835 024342 045 116 045 WR9:: .ASCIZ ?%N%AFAILING COMMAND : %T%N%AUNIT INOPERATIVE?
1836
1837
1838 024417 045 116 062 L21:: .ASCIZ ?%N2%A**** LEVEL 2 MICRODIAGNOSTIC DUMP ****?
1839 024473 045 116 062 L22:: .ASCIZ ?%N2%APROGRAM STATUS: %06?
1840 024531 045 116 045 L23:: .ASCIZ ?%N%ADRIE STATUS: %03?
1841 024571 045 116 045 L24:: .ASCIZ ?%N%AATEST NUMBER: %03?
1842 024631 045 116 045 L25:: .ASCIZ ?%N%ATRACK NUMBER: %03?
1843 024671 045 116 045 L26:: .ASCIZ ?%N%ATMSCP COMMAND: %03?
1844 024731 045 116 045 L27:: .ASCIZ ?%N%ATMSCP RESPONSE FLAG: %03?
1845 024771 045 116 045 L28:: .ASCIZ ?%N%ATMSCP UNIT FLAG: %06?
1846 025026 045 116 062 L29:: .ASCIZ ?%N2%ALOCKS WRITTEN CHANNEL 1: %D6?
1847 025073 045 116 045 L210:: .ASCIZ ?%N%ABLOCKS WRITTEN CHANNEL 2: %D6?
1848 025137 045 116 045 L211:: .ASCIZ ?%N%ABLOCKS READ CHANNEL 1: %D6?
1849 025203 045 116 045 L212:: .ASCIZ ?%N%ABLOCKS READ CHANNEL 2: %D6?
1850 025247 045 116 045 L213:: .ASCIZ ?%N%ASOFT WRITE CHANNEL 1: %D6?
1851 025313 045 116 045 L214:: .ASCIZ ?%N%ASOFT WRITE CHANNEL 2: %D6?
1852 025357 045 116 045 L215:: .ASCIZ ?%N%AECC CORRECTED CHANNEL 1: %D6?
1853 025423 045 116 045 L216:: .ASCIZ ?%N%AECC CORRECTED CHANNEL 2: %D6?
1854 025467 045 116 045 L219:: .ASCIZ ?%N%AREAD REPOSITIONS CHANNEL 1: %D6?
1855 025533 045 116 045 L220:: .ASCIZ ?%N%AREAD REPOSITIONS CHANNEL 2: %D6%N?
1856 025601 045 116 062 L221:: .ASCIZ ?%N2%AERROR LOG (TAPE TRANSFER ERROR)?
1857 025646 045 116 045 L222:: .ASCIZ ?%N%AEERROR FORMAT: %03?
1858 025710 045 116 045 L223:: .ASCIZ ?%N%AEERROR FLAGS: %03?
1859 025752 045 116 045 L224:: .ASCIZ ?%N%AEVENT CODE: %06?
1860 026011 045 116 045 L225:: .ASCIZ ?%N%AEERROR RETRY LEVEL: %03?
1861 026053 045 116 045 L226:: .ASCIZ ?%N%ANUMBER OF RETRIES: %03?
1862 026115 045 116 045 L227:: .ASCIZ ?%N%APOSITION (LOW ORDER): %06?
1863 026154 045 116 045 L228:: .ASCIZ ?%N%APOSITION (HIGH ORDER): %06?
1864 026213 045 116 045 L229:: .ASCIZ ?%N%ACONTROLLER STATUS: %03?
1865 026255 045 116 045 L230:: .ASCIZ ?%N%ADRIE ERROR CODE: %03?
1866 026317 045 116 045 L231:: .ASCIZ ?%N%ADRIE FLAGS: %03?
1867 026361 045 116 045 L232:: .ASCIZ ?%N%ATRACK NUMBER: %03?
1868 026423 045 116 045 L233:: .ASCIZ ?%N%APHYSICAL BLOCK NUMBER: %D6?
    
```



1869	026462	045	116	045	L234::	.ASCIZ	?N#ALOGICAL BLOCK NUMBER:	%03?		
1870	026524	045	116	045	L235::	.ASCIZ	?N#ATAPE COUNT 0 :	%03?		
1871	026566	045	116	045	L236::	.ASCIZ	?N#ATAPE COUNT 1 :	%03?		
1872	026630	045	116	045	L237::	.ASCIZ	?N#ATAPE COUNT 2 :	%03?		
1873	026672	045	116	045	L238::	.ASCIZ	?N#ADRIIVE STATE:	%06?		
1874	026731	045	116	045	L239::	.ASCIZ	?N#AREAD/WRITE STATE:	%06?		
1875	026770	045	116	045	L240::	.ASCIZ	?N#AOPERATION FLAGS:	%06?		
1876	027027	045	116	062	L241::	.ASCIZ	?N2#AERROR LOG (CONTROLLER ERROR)?			
1877	027071	045	116	045	L242::	.ASCIZ	?N#ACONTROLLER ERROR CODE:	%06?		
1878										
1879	027130	045	116	062	L244::	.ASCIZ	?N2#ABYTE READ: %03#A	%03#A	%03#A	%06?
1880	027224	045	116	045	L245::	.ASCIZ	?N#A	%03#A	%03#A	%06?
1881	027320	045	116	045	L246::	.ASCIZ	?N#A	%03#A	%03#A	%06?
1882	027414	045	116	045	L247::	.ASCIZ	?N#A	%03#A	%03#A	%06?
1883	027510	045	116	045	L248::	.ASCIZ	?N#A	%03#A	%03#A	%06?
1884	027604	045	116	045	L249::	.ASCIZ	?N#A	%03#A	%03#A	%06?
1885	027700	045	116	045	L250::	.ASCIZ	?N#A	%03#A	%03#A	%06?
1886	027774	045	116	045	L251::	.ASCIZ	?N#A	%03#A	%03#A	%06?
1887	030070	045	116	045	L252::	.ASCIZ	?N#A	%03#A	%03#A	%06?
1888	030164	045	116	045	L253::	.ASCIZ	?N#A	%03#A	%03#A	%06?
1889	030260	045	116	062	L254::	.ASCIZ	?N2#ATOTAL NUMBER OF MISCOMPARED BYTES:	%06?		
1890							.EVEN			
1891										
1892										
1893										
1894										
1895										
1896										
1897										
1898										
1899	030334	116	130	115	MSG5::	.ASCIZ	?NXM ON READ TKIP?			
1900	030355	124	113	111	MSG6::	.ASCIZ	?TKIP NOT 0 ON FIRST READ?			
1901	030406	116	130	115	MSG7::	.ASCIZ	?NXM ON READ TKSA?			
1902	030427	123	101	040	MSG8::	.ASCIZ	?SA REG IN ERROR ON FIRST READ?			
1903	030465	123	101	040	MSG9::	.ASCIZ	?SA CONTENTS IN ERROR?			
1904	030512	123	101	040	MSG10::	.ASCIZ	?SA WRONG IN DATA WRAP?			
1905	030540	105	130	120	MSG11::	.ASCIZ	?EXPECTED INTERRUPT DID NOT OCCUR?			
1906	030601	111	116	124	MSG12::	.ASCIZ	?INTRRPT OCCURRED WITH CPU PRIORITY = 7?			
1907	030650	123	101	040	MSG13::	.ASCIZ	?SA NOT 0 IN PURGE/POLL?			
1908	030677	120	125	122	MSG14::	.ASCIZ	?PURGE/POLL TEST FAILED?			
1909	030726	105	130	124	MSG15::	.ASCIZ	?EXTENDED ADDRESS TEST FAILED?			
1910										
1911	030763	111	116	126	L2ER1::	.ASCIZ	?INVALID COMMAND?			
1912	031003	103	117	115	L2ER2::	.ASCIZ	?COMMAND ABORTED?			
1913	031023	125	116	111	L2ER3::	.ASCIZ	?UNIT-OFFLINE?			
1914	031040	125	116	111	L2ER4::	.ASCIZ	?UNIT-AVAILABLE?			
1915	031057	111	116	126	L2ER5::	.ASCIZ	?INVALID STATUS RECEIVED?			
1916	031107	127	122	111	L2ER6::	.ASCIZ	?WRITE PROTECTED?			
1917	031127	103	117	115	L2ER7::	.ASCIZ	?COMPARE ERROR?			
1918	031145	104	101	124	L2ER8::	.ASCIZ	?DATA ERROR?			
1919	031160	110	117	123	L2ER9::	.ASCIZ	?HOST BUFFER ACCESS ERROR?			
1920	031211	103	117	116	L2ER10::	.ASCIZ	?CONTROLLER ERROR?			
1921	031232	104	122	111	L2ER11::	.ASCIZ	?DRIVE ERROR?			
1922	031246	106	117	122	L2ER12::	.ASCIZ	?FORMATTER ERROR?			
1923	031260	102	117	124	L2ER13::	.ASCIZ	?BOT ENCOUNTERED?			
1924	031306	124	101	120	L2ER14::	.ASCIZ	?TAPE MARK ENCOUNTERED?			
1925	031334	111	116	126	L2ER15::	.ASCIZ	?INVALID STATUS RECEIVED?			

```

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

1926	031364	122	105	103	L2ER16:: .ASCIZ	?RECORD DATA TRUNCATED?
1927	031412	120	117	123	L2ER17:: .ASCIZ	?POSITION LOST?
1928	031430	123	105	122	L2ER18:: .ASCIZ	?SERIOUS EXEPTION?
1929	031451	114	105	117	L2ER19:: .ASCIZ	?LEOT DETECTED?
1930	031467	115	111	103	L2ER20:: .ASCIZ	?MICRODIAGNOSTIC PROGRAM ERROR: DATA COMPARE ERROR?
1931	031551	115	111	103	L2ER21:: .ASCIZ	?MICRODIAGNOSTIC PROG'M ERROR: CONTROLLER ERROR?
1932	031631	115	111	103	L2ER22:: .ASCIZ	?MICRODIAGNOSTIC PROGRAM ERROR: INVALID STATUS RECEIVED?
1933	031720	115	111	103	L2ER23:: .ASCIZ	?MICRODIAGNOSTIC PROGRAM ERROR: BAD PATTERN NUMBER ERROR?
1934	032010	115	111	103	L2ER24:: .ASCIZ	?MICRODIAGNOSTIC PROGRAM ERROR: RESPONSE ADDRESS ERROR?
1935	032076	115	111	103	L2ER25:: .ASCIZ	?MICRODIAGNOSTIC PROGRAM ERROR: HOST BUFFER ADDRESS ERROR?
1936	032167	115	111	103	L2ER26:: .ASCIZ	?MICRODIAGNOSTIC PROGRAM ERROR: UNKNOWN ERROR LOG RECEIVED?
1937	032261	115	111	103	L2ER27:: .ASCIZ	?MICRODIAGNOSTIC PROGRAM ERROR: RESPONSE TIMEOUT ERROR?
1938						
1939	032347	124	111	115	WRER1:: .ASCIZ	?TIME OUT DURING PORT INIT?
1940	032401	120	117	122	WRER2:: .ASCIZ	?PORT INIT FAILED?
1941	032422	124	115	123	WRER3:: .ASCIZ	?TMSCP COMMAND FAILURE?
1942	032450	120	117	122	WRER4:: .ASCIZ	?PORT DETECTED ERROR?
1943	032474	122	105	123	WRER5:: .ASCIZ	?RESPONSE OUT OF SEQUENCE?
1944	032525	103	117	115	WRER6:: .ASCIZ	?COMMAND TIME OUT?
1945	032546	125	116	105	WRER7:: .ASCIZ	?UNEXPECTED STATUS FOR ONLINE RECEIVED?
1946	032614	104	125	120	WRER8:: .ASCIZ	?DUP COMMAND FAILURE?
1947					.EVEN	
1948						
1949						::*****
1950						:
1951						: MISCELLANEOUS ERROR MESSAGES
1952						:
1953						::*****
1954						
1955	032640	103	116	124	CTRL:: .ASCIZ	?CNTRLR/CABLE?
1956	032655	104	122	111	DRVE:: .ASCIZ	?DRIVE?
1957	032663	123	103	103	SCCMD:: .ASCIZ	?SCC ?
1958	032670	117	116	114	ONLCMD:: .ASCIZ	?ONL ?
1959	032675	122	105	127	REWCMD:: .ASCIZ	?REW ?
1960	032702	127	122	040	WRCMD:: .ASCIZ	?WR ?
1961	032707	122	104	040	RDCMD:: .ASCIZ	?RD ?
1962	032714	107	104	123	GDSCMD:: .ASCIZ	?GDS ?
1963	032721	105	114	120	ELPCMD:: .ASCIZ	?ELP ?
1964	032726	122	103	126	RCVCMDC:: .ASCIZ	?RCV ?
1965	032733	101	102	124	ABTCMD:: .ASCIZ	?ABT ?
1966					.EVEN	
1967						

1969  
 1973  
 1974  
 1975  
 1976  
 1977  
 1978  
 1979  
 1980  
 1981  
 1982  
 1983  
 1987  
 1988 032740  
 1989  
 1990 032740  
 1991 032740  
     032740 013746 002316  
     032744 012746 023420  
     032750 012746 000002  
     032754 010600  
     032756 104415  
     032760 062706 000006  
 1992  
 1993 032764  
 1994 032764  
     032764 016446 000012  
     032770 013746 002314  
     032774 016446 000002  
     033000 012746 023454  
     033004 012746 000004  
     033010 010600  
     033012 104415  
     033014 062706 000012  
 1995 033020 000452  
 1996  
 1997 033022  
 1998 033022  
     033022 010246  
     033024 012746 023746  
     033030 012746 000002  
     033034 010600  
     033036 104415  
     033040 062706 000006  
 1999 033044 000413  
 2000  
 2001 033046  
 2002 033046  
     033046 010246  
     033050 013746 172346  
     033054 012746 023626  
     033060 012746 000003  
     033064 010600  
     033066 104415  
     033070 062706 000010  
 2003  
 2004 033074

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

BGNMSG

PRINI::

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 TKSASV(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 #LINES,KPAR3,R2  
 MOV R2,-(SP)  
 MOV KPAR3,-(SP)  
 MOV #LINES,-(SP)  
 MOV #3,-(SP)  
 MOV SP,R0  
 TRAP C#PNTX  
 ADD #10,SP

PRIDAT::

2005	033074			PRINTX	#LINE6,R1,(R2)
	033074	011246		MOV	(R2),-(SP)
	033076	010146		MOV	R1,-(SP)
	033100	012746	023703	MOV	#LINE6,-(SP)
	033104	012746	000003	MOV	#3,-(SP)
	033110	010600		MOV	SP,R0
	033112	104415		TRAP	C#PNTX
	033114	062706	000010	ADD	#10,SP
2006	033120	000412		BR	PRIERR
2007					
2008	033122			PRIIP::	
2009	033122			PRINTX	#LINE3,TKIP(R4)
	033122	016446	000000	MOV	TKIP(R4),-(SP)
	033126	012746	023534	MOV	#LINE3,-(SP)
	033132	012746	000002	MOV	#2,-(SP)
	033136	010600		MOV	SP,R0
	033140	104415		TRAP	C#PNTX
	033142	062706	000006	ADD	#6,SP
2010					
2011	033146			PRIERR::	
2012	033146			PRINTB	#LINE4,FRUIS
	033146	013746	002310	MOV	FRUIS,-(SP)
	033152	012746	023564	MOV	#LINE4,-(SP)
	033156	012746	000002	MOV	#2,-(SP)
	033162	010600		MOV	SP,R0
	033164	104414		TRAP	C#PNTB
	033166	062706	000006	ADD	#6,SP
2013	033172	000137	033536	JMP	PRIEX
2014					
2015	033176			WRINTO::	
2016	033176			PRINTX	#LINE1,INISTP
	033176	013746	002316	MOV	INISTP,-(SP)
	033202	012746	023420	MOV	#LINE1,-(SP)
	033206	012746	000002	MOV	#2,-(SP)
	033212	010600		MOV	SP,R0
	033214	104415		TRAP	C#PNTX
	033216	062706	000006	ADD	#6,SP
2017					
2018	033222			WRPRTE::	
2019	033222			PRINTX	#WR1,TKSA(R4),TKSASV(R4)
	033222	016446	000012	MOV	TKSASV(R4),-(SP)
	033226	016446	000002	MOV	TKSA(R4),-(SP)
	033232	012746	023772	MOV	#WR1,-(SP)
	033236	012746	000003	MOV	#3,-(SP)
	033242	010600		MOV	SP,R0
	033244	104415		TRAP	C#PNTX
	033246	062706	000010	ADD	#10,SP
2020	033252	000137	033512	JMP	WREX
2021					
2022	033256			WRCMDE::	
2023	033256			PRINTX	R2,CURCMD
	033256	013746	023042	MOV	CURCMD,(SP)
	033262	010246		MOV	R2,-(SP)
	033264	012746	000002	MOV	#2,-(SP)
	033270	010600		MOV	SP,R0
	033272	104415		TRAP	C#PNTX
	033274	062706	000006	ADD	#6,SP

2024	033300			PRINTX	#WR3
	033300	012746	024071	MOV	#WR3,-(SP)
	033304	012746	000001	MOV	#1,-(SP)
	033310	010600		MOV	SP,R0
	033312	104415		TRAP	C\$PNTX
	033314	062706	000004	ADD	#4,SP
2025	033320	010301		MOV	R3,R1
2026	033322			PRINTX	#WR4,2(R1),(R1)
	033322	011146		MOV	(R1),-(SP)
	033324	016146	000002	MOV	2(R1),-(SP)
	033330	012746	024122	MOV	#WR4,(SP)
	033334	012746	000003	MOV	#3,-(SP)
	033340	010600		MOV	SP,R0
	033342	104415		TRAP	C\$PNTX
	033344	062706	000010	ADD	#10,SP
2027	033350	062701	000004	ADD	#4,R1
2028	033354	162763	000004	SUB	#4,MSGLEN(R3)
2029	033362	001357		BNE	W1
2030	033364			PRINTX	#WR6
	033364	012746	024211	MOV	#WR6,-(SP)
	033370	012746	000001	MOV	#1,-(SP)
	033374	010600		MOV	SP,R0
	033376	104415		TRAP	C\$PNTX
	033400	062706	000004	ADD	#4,SP
2031	033404	000137	033512	JMP	WREX
2032					
2033	033410			WRSEQE::	
2034	033410			PRINTX	#WR2,CURCMD
	033410	013746	023042	MOV	CURCMD,-(SP)
	033414	012746	024035	MOV	#WR2,-(SP)
	033420	012746	000002	MOV	#2,-(SP)
	033424	010600		MOV	SP,R0
	033426	104415		TRAP	C\$PNTX
	033430	062706	000006	ADD	#6,SP
2035	033434			PRINTX	#WR5,P.CRF(R3),P.CRF(R5)
	033434	016546	000000	MOV	P.CRF(R5),-(SP)
	033440	016346	000000	MOV	P.CRF(R3),-(SP)
	033444	012746	024147	MOV	#WR5,-(SP)
	033450	012746	000003	MOV	#3,-(SP)
	033454	010600		MOV	SP,R0
	033456	104415		TRAP	C\$PNTX
	033460	062706	000010	ADD	#10,SP
2036	033464	000412		BR	WREX
2037					
2038	033466			WRTOE::	
2039	033466			PRINTX	#WR2,CURCMD
	033466	013746	023042	MOV	CURCMD,-(SP)
	033472	012746	024035	MOV	#WR2,(SP)
	033476	012746	000002	MOV	#2,-(SP)
	033502	010600		MOV	SP,R0
	033504	104415		TRAP	C\$PNTX
	033506	062706	000006	ADD	#6,SP
2040					
2041	033512			WREX::	
2042	033512			PRINTB	#LINE4,FRUIS
	033512	013746	002310	MOV	FRUIS,-(SP)
	033516	012746	023564	MOV	#LINE4,(SP)

	033522	012746	000002	MOV	#2,-(SP)
	033526	010600		MOV	SP,R0
	033530	104414		TRAP	C#PNTB
	033532	062706	000006	ADD	#6,SP
2043					
2044	033536			PRIEX: EXIT	MSG
	033536	000167		.WORD	J\$JMP
	033540	000000		.WORD	L10003 2 .
2045					
2046	033542			ENDMSG	
	033542			L10003:	
	033542	104423		TRAP	C#MSG
2047					
2048	033544			BGNMSG	L2DUMP
	033544			L2DUMP::	
2049					
2050	033544			PRINTB	#LINE4,FRUIS
	033544	013746	002310	MOV	FRUIS,(SP)
	033550	012746	023564	MOV	#LINE4,-(SP)
	033554	012746	000002	MOV	#2,-(SP)
	033560	010600		MOV	SP,R0
	033562	104414		TRAP	C#PNTB
	033564	062706	000006	ADD	#6,SP
2051	033570			PRINTB	#L21
	033570	012746	024417	MOV	#L21,-(SP)
	033574	012746	000001	MOV	#1,-(SP)
	033600	010600		MOV	SP,R0
	033602	104414		TRAP	C#PNTB
	033604	062706	000004	ADD	#4,SP
2052	033610			PRINTB	#L22,L2STA(R1)
	033610	016146	000002	MOV	L2STA(R1),-(SP)
	033614	012746	024473	MOV	#L22,-(SP)
	033620	012746	000002	MOV	#2,-(SP)
	033624	010600		MOV	SP,R0
	033626	104414		TRAP	C#PNTB
	033630	062706	000006	ADD	#6,SP
2053	033634			PRINTB	#L23,L2DRV(R1)
	033634	016146	000004	MOV	L2DRV(R1),-(SP)
	033640	012746	024531	MOV	#L23,-(SP)
	033644	012746	000002	MOV	#2,-(SP)
	033650	010600		MOV	SP,R0
	033652	104414		TRAP	C#PNTB
	033654	062706	000006	ADD	#6,SP
2054	033660			PRINTB	#L24,<B,L2TST(R1)>
	033660	005046		CLR	-(SP)
	033662	156116	000006	BISB	L2TST(R1),(SP)
	033666	012746	024571	MOV	#L24,-(SP)
	033672	012746	000002	MOV	#2,-(SP)
	033676	010600		MOV	SP,R0
	033700	104414		TRAP	C#PNTB
	033702	062706	000006	ADD	#6,SP
2055	033706			PRINTB	#L25,<B,L2TRK(R1)>
	033706	005046		CLR	-(SP)
	033710	156116	000007	BISB	L2TRK(R1),(SP)
	033714	012746	024631	MOV	#L25,-(SP)
	033720	012746	000002	MOV	#2,-(SP)
	033724	010600		MOV	SP,R0

	033726	104414			TRAP	C:PNTB
	033730	062706	000006		ADD	#6,SP
2056	033734				PRINTB	#L26,<B,L2CMD(R1)>
	033734	005046			CLR	-(SP)
	033736	156116	000010		BISB	L2CMD(R1),(SP)
	033742	012746	024671		MOV	#L26,-(SP)
	033746	012746	000002		MOV	#2,-(SP)
	033752	010600			MOV	SP,RO
	033754	104414			TRAP	C:PNTB
	033756	062706	000006		ADD	#6,SP
2057	033762				PRINTB	#L27,<B,L2RSP(R1)>
	033762	005046			CLR	-(SP)
	033764	156116	000011		BISB	L2RSP(R1),(SP)
	033770	012746	024731		MOV	#L27,-(SP)
	033774	012746	000002		MOV	#2,-(SP)
	034000	010600			MOV	SP,RO
	034002	104414			TRAP	C:PNTB
	034004	062706	000006		ADD	#6,SP
2058	034010				PRINTB	#L28,L2UNT(R1)
	034010	016146	000012		MOV	L2UNT(R1),-(SP)
	034014	012746	024771		MOV	#L28,-(SP)
	034020	012746	000002		MOV	#2,-(SP)
	034024	010600			MOV	SP,RO
	034026	104414			TRAP	C:PNTB
	034030	062706	000006		ADD	#6,SP
2059	034034	126127	000134	000005	CMPB	L2ELFM(R1),#TXFER
2060	034042	001402			BEQ	1\$
2061	034044	000137	034720		JMP	10\$
2062	034050			1\$:	PRINTB	#L221
	034050	012746	025601		MOV	#L221,-(SP)
	034054	012746	000001		MOV	#1,-(SP)
	034060	010600			MOV	SP,RO
	034062	104414			TRAP	C:PNTB
	034064	062706	000004		ADD	#4,SP
2063	034070				PRINTB	#L222,<B,L2ELFM(R1)>
	034070	005046			CLR	-(SP)
	034072	156116	000134		BISB	L2ELFM(R1),(SP)
	034076	012746	025646		MOV	#L222,-(SP)
	034102	012746	000002		MOV	#2,-(SP)
	034106	010600			MOV	SP,RO
	034110	104414			TRAP	C:PNTB
	034112	062706	000006		ADD	#6,SP
2064	034116				PRINTB	#L223,<B,L2ELFL(R1)>
	034116	005046			CLR	-(SP)
	034120	156116	000135		BISB	L2ELFL(R1),(SP)
	034124	012746	025710		MOV	#L223,-(SP)
	034130	012746	000002		MOV	#2,-(SP)
	034134	010600			MOV	SP,RO
	034136	104414			TRAP	C:PNTB
	034140	062706	000006		ADD	#6,SP
2065	034144				PRINTB	#L224,L2ELEV(R1)
	034144	016146	000136		MOV	L2ELEV(R1),-(SP)
	034150	012746	025752		MOV	#L224,-(SP)
	034154	012746	000002		MOV	#2,-(SP)
	034160	010600			MOV	SP,RO
	034162	104414			TRAP	C:PNTB
	034164	062706	000006		ADD	#6,SP

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

2066	034170			PRINTB	#L225,<B,L2ELRL(R1)>
	034170	005046		CLR	-(SP)
	034172	156116	000144	BISB	L2ELRL(R1),(SP)
	034176	012746	026011	MOV	#L225,-(SP)
	034202	012746	000002	MOV	#2,-(SP)
	034206	010600		MOV	S,R0
	034210	104414		TRAP	C#PNTB
	034212	062706	000006	ADD	#6,SP
2067	034216			PRINTB	#L226,<B,L2ELRT(R1)>
	034216	005046		CLR	-(SP)
	034220	156116	000145	BISB	L2ELRT(R1),(SP)
	034224	012746	026053	MOV	#L226,-(SP)
	034230	012746	000002	MOV	#2,-(SP)
	034234	010600		MOV	SP,R0
	034236	104414		TRAP	C#PNTB
	034240	062706	000006	ADD	#6,SP
2068	034244			PRINTB	#L227,L2ELP1(R1)
	034244	016146	000146	MOV	L2ELP1(R1),-(SP)
	034250	012746	026115	MOV	#L227,-(SP)
	034254	012746	000002	MOV	#2,-(SP)
	034260	010600		MOV	SP,R0
	034262	104414		TRAP	C#PNTB
	034264	062706	000006	ADD	#6,SP
2069	034270			PRINTB	#L228,L2ELP2(R1)
	034270	016146	000150	MOV	L2ELP2(R1),-(SP)
	034274	012746	026154	MOV	#L228,-(SP)
	034300	012746	000002	MOV	#2,-(SP)
	034304	010600		MOV	SP,R0
	034306	104414		TRAP	C#PNTB
	034310	062706	000006	ADD	#6,SP
2070	034314			PRINTB	#L229,<B,L2ELST(R1)>
	034314	005046		CLR	-(SP)
	034316	156116	000154	BISB	L2ELST(R1),(SP)
	034322	012746	026213	MOV	#L229,-(SP)
	034326	012746	000002	MOV	#2,-(SP)
	034332	010600		MOV	SP,R0
	034334	104414		TRAP	C#PNTB
	034336	062706	000006	ADD	#6,SP
2071	034342			PRINTB	#L230,<B,L2ELDE(R1)>
	034342	005046		CLR	-(SP)
	034344	156116	000155	BISB	L2ELDE(R1),(SP)
	034350	012746	026255	MOV	#L230,-(SP)
	034354	012746	000002	MOV	#2,-(SP)
	034360	010600		MOV	SP,R0
	034362	104414		TRAP	C#PNTB
	034364	062706	000006	ADD	#6,SP
2072	034370			PRINTB	#L231,<B,L2ELDF(R1)>
	034370	005046		CLR	-(SP)
	034372	156116	000156	BISB	L2ELDF(R1),(SP)
	034376	012746	026317	MOV	#L231,-(SP)
	034402	012746	000002	MOV	#2,-(SP)
	034406	010600		MOV	SP,R0
	034410	104414		TRAP	C#PNTB
	034412	062706	000006	ADD	#6,SP
2073	034416			PRINTB	#L232,<B,L2ELTN(R1)>
	034416	005046		CLR	-(SP)
	034420	156116	000157	BISB	L2ELTN(R1),(SP)



	034424	012746	026361	MOV	@L232,-(SP)
	034430	012746	000002	MOV	@2,-(SP)
	034434	010600		MOV	SP,R0
	034436	104414		TRAP	C:PNTB
	034440	062706	000006	ADD	@6,SP
2074	034444			PRINTB	@L233,L2ELPB(R1)
	034444	016146	000160	MOV	L2ELPB(R1),(SP)
	034450	012746	026423	MOV	@L233,-(SP)
	034454	012746	000002	MOV	@2,(SP)
	034460	010600		MOV	SP,R0
	034462	104414		TRAP	C:PNTB
	034464	062706	000006	ADD	@6,SP
2075	034470			PRINTB	@L234,<B,L2ELLB(R1)>
	034470	005046		CLR	-(SP)
	034472	156116	000162	BISB	L2ELLB(R1),(SP)
	034476	012746	026462	MOV	@L234,-(SP)
	034502	012746	000002	MOV	@2,-(SP)
	034506	010600		MOV	SP,R0
	034510	104414		TRAP	C:PNTB
	034512	062706	000006	ADD	@6,SP
2076	034516			PRINTB	@L235,<B,L2ELT0(R1)>
	034516	005046		CLR	-(SP)
	034520	156116	000163	BISB	L2ELT0(R1),(SP)
	034524	012746	026524	MOV	@L235,-(SP)
	034530	012746	000002	MOV	@2,-(SP)
	034534	010600		MOV	SP,R0
	034536	104414		TRAP	C:PNTB
	034540	062706	000006	ADD	@6,SP
2077	034544			PRINTB	@L236,<B,L2ELT1(R1)>
	034544	005046		CLR	-(SP)
	034546	156116	000164	BISB	L2ELT1(R1),(SP)
	034552	012746	026566	MOV	@L236,-(SP)
	034556	012746	000002	MOV	@2,-(SP)
	034562	010600		MOV	SP,R0
	034564	104414		TRAP	C:PNTB
	034566	062706	000006	ADD	@6,SP
2078	034572			PRINTB	@L237,<B,L2ELT2(R1)>
	034572	005046		CLR	-(SP)
	034574	156116	000165	BISB	L2ELT2(R1),(SP)
	034600	012746	026630	MOV	@L237,-(SP)
	034604	012746	000002	MOV	@2,-(SP)
	034610	010600		MOV	SP,R0
	034612	104414		TRAP	C:PNTB
	034614	062706	000006	ADD	@6,SP
2079	034620			PRINTB	@L238,L2ELDS(R1)
	034620	016146	000166	MOV	L2ELDS(R1),(SP)
	034624	012746	026672	MOV	@L238,-(SP)
	034630	012746	000002	MOV	@2,-(SP)
	034634	010600		MOV	SP,R0
	034636	104414		TRAP	C:PNTB
	034640	062706	000006	ADD	@6,SP
2080	034644			PRINTB	@L239,L2ELRW(R1)
	034644	016146	000170	MOV	L2ELRW(R1),(SP)
	034650	012746	026731	MOV	@L239,-(SP)
	034654	012746	000002	MOV	@2,-(SP)
	034660	010600		MOV	SP,R0
	034662	104414		TRAP	C:PNTB

2081	034664	062706	000006			ADD	#6,SP	
	034670					PRINTB	#L240,L2ELOF(R1)	
	034670	016146	000172			MOV	L2ELOF(R1), (SP)	
	034674	012746	026770			MOV	#L240, (SP)	
	034700	012746	000002			MOV	#2,-(SP)	
	034704	010600				MOV	SP,R0	
	034706	104414				TRAP	C\$PNTB	
	034710	062706	000006			ADD	#6,SP	
2082	034714	000137	034774			JMP	25\$	
2083	034720	126127	000134	000000	10\$:	CMPB	L2ELFM(R1),#CNTER	; "CONTROLLER ERROR" ERROR LOG ?
2084	034726	001022				BNE	25\$	; NO, NO ERROR LOG TO PRINT, EXIT
2085	034730					PRINTB	#L241	; DUMP "CONTROLLER ERROR" ERROR LOG
	034730	012746	027027			MOV	#L241,-(SP)	
	034734	012746	000001			MOV	#1, (SP)	
	034740	010600				MOV	SP,R0	
	034742	104414				TRAP	C\$PNTB	
	034744	062706	000004			ADD	#4,SP	
2086	034750					PRINTB	#L242,L2ELEC(R1)	
	034750	016146	000140			MOV	L2ELEC(R1),-(SP)	
	034754	012746	027071			MOV	#L242,-(SP)	
	034760	012746	000002			MOV	#2,-(SP)	
	034764	010600				MOV	SP,R0	
	034766	104414				TRAP	C\$PNTB	
	034770	062706	000006			ADD	#6,SP	
2087	034774				25\$:	EXIT	MSG	
	034774	000167				.WORD	J\$JMP	
	034776	000000				.WORD	L10004 2--	
2088								
2089	035000					ENDMSG		
	035000				L10004:			
	035000	104423				TRAP	C\$MSG	

```

2091 .SBTTL STATISTICAL REPORT SECTION
2092 035002 STARS
      ;:*****
2093 035002 STARS
      ;:*****
2094 ;
2095 ;STATISTICAL REPORT SECTION
2096 ; THESE ARE THE "PRINTS" STATEMENTS USED TO DUMP THE LEVEL 2
2097 ; MICRODIAGNOSTIC TEST RESULTS
2098 ;
2099 035002 STARS
      ;:*****
2100 035002 STARS
      ;:*****
2101
2102 035002 BGNRPT
      035002 L$RPT::
2103 035002 PRINTS #L29,L2BWR1(R1)
      035002 016146 000014 MOV L2BWR1(R1),-(SP)
      035006 012746 025026 MOV #L29,-(SP)
      035012 012746 000002 MOV #2,-(SP)
      035016 010600 MOV SP,R0
      035020 104416 TRAP C$PNTS
      035022 062706 000006 ADD #6,SP
2104 035026 PRINTS #L210,L2BWR2(R1)
      035026 016146 000020 MOV L2BWR2(R1),-(SP)
      035032 012746 025073 MOV #L210,-(SP)
      035036 012746 000002 MOV #2,-(SP)
      035042 010600 MOV SP,R0
      035044 104416 TRAP C$PNTS
      035046 062706 000006 ADD #6,SP
2105 035052 PRINTS #L211,L2BRD1(R1)
      035052 016146 000024 MOV L2BRD1(R1),-(SP)
      035056 012746 025137 MOV #L211,-(SP)
      035062 012746 000002 MOV #2,-(SP)
      035066 010600 MOV SP,R0
      035070 104416 TRAP C$PNTS
      035072 062706 000006 ADD #6,SP
2106 035076 PRINTS #L212,L2BRD2(R1)
      035076 016146 000030 MOV L2BRD2(R1),-(SP)
      035102 012746 025203 MOV #L212,-(SP)
      035106 012746 000002 MOV #2,-(SP)
      035112 010600 MOV SP,R0
      035114 104416 TRAP C$PNTS
      035116 062706 000006 ADD #6,SP
2107 035122 PRINTS #L213,L2SWR1(R1)
      035122 016146 000034 MOV L2SWR1(R1),-(SP)
      035126 012746 025247 MOV #L213,-(SP)
      035132 012746 000002 MOV #2,-(SP)
      035136 010600 MCV SP,R0
      035140 104416 TRAP C$PNTS
      035142 062706 000006 ADD #6,SP
2108 035146 PRINTS #L214,L2SWR2(R1)
      035146 016146 000036 MOV L2SWR2(R1),-(SP)
      035152 012746 025313 MOV #L214,-(SP)
      035156 012746 000002 MOV #2,-(SP)
      035162 010600 MOV SP,R0
    
```

	035164	104416			TRAP	C:PNTS
	035166	062706	000006		ADD	#6,SP
2109	035172				PRINTS	#L215,L2ECC1(R1)
	035172	016146	000040		MOV	L2ECC1(R1),-(SP)
	035176	012746	025357		MOV	#L215,-(SP)
	035202	012746	000002		MOV	#2,-(SP)
	035206	010600			MOV	SP,R0
	035210	104416			TRAP	C:PNTS
	035212	062706	000006		ADD	#6,SP
2110	035216				PRINTS	#L216,L2ECC2(R1)
	035216	016146	000042		MOV	L2ECC2(R1),-(SP)
	035222	012746	025423		MOV	#L216,-(SP)
	035226	012746	000002		MOV	#2,-(SP)
	035232	010600			MOV	SP,R0
	035234	104416			TRAP	C:PNTS
	035236	062706	000006		ADD	#6,SP
2111	035242				PRINTS	#L219,L2REP1(R1)
	035242	016146	000050		MOV	L2REP1(R1),-(SP)
	035246	012746	025467		MOV	#L219,-(SP)
	035252	012746	000002		MOV	#2,-(SP)
	035256	010600			MOV	SP,R0
	035260	104416			TRAP	C:PNTS
	035262	062706	000006		ADD	#6,SP
2112	035266				PRINTS	#L220,L2REP2(R1)
	035266	016146	000052		MOV	L2REP2(R1),-(SP)
	035272	012746	025533		MOV	#L220,-(SP)
	035276	012746	000002		MOV	#2,-(SP)
	035302	010600			MOV	SP,R0
	035304	104416			TRAP	C:PNTS
	035306	062706	000006		ADD	#6,SP
2113	035312	026127	000124	000000	CMP	124(R1),#0
2114	035320	001002			BNE	1\$
2115	035322	000137	036052		JMP	10\$
2116	035326			1\$:	PRINTS	#L244,<B,66(R1)>,<B,54(R1)>,100(R1)
	035326	016146	000100		MOV	100(R1),-(SP)
	035332	005046			CLR	-(SP)
	035334	156116	000054		BISB	54(R1),(SP)
	035340	005046			CLR	-(SP)
	035342	156116	000066		BISB	66(R1),(SP)
	035346	012746	027130		MOV	#L244,-(SP)
	035352	012746	000004		MOV	#4,-(SP)
	035356	010600			MOV	SP,R0
	035360	104416			TRAP	C:PNTS
	035362	062706	000012		ADD	#12,SP
2117	035366				PRINTS	#L245,<B,67(R1)>,<B,55(R1)>,102(R1)
	035366	016146	000102		MOV	102(R1),-(SP)
	035372	005046			CLR	-(SP)
	035374	156116	000055		BISB	55(R1),(SP)
	035400	005046			CLR	-(SP)
	035402	156116	000067		BISB	67(R1),(SP)
	035406	012746	027224		MOV	#L245,-(SP)
	035412	012746	000004		MOV	#4,-(SP)
	035416	010600			MOV	SP,R0
	035420	104416			TRAP	C:PNTS
	035422	062706	000012		ADD	#12,SP
2118	035426				PRINTS	#L246,<B,70(R1)>,<B,56(R1)>,104(R1)
	035426	016146	000104		MOV	104(R1),-(SP)

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

	035432	005046		CLR	(SP)
	035434	156116	000056	BISB	56(R1),(SP)
	035440	005046		CLR	-(SP)
	035442	156116	000070	BISB	70(R1),(SP)
	035446	012746	027320	MOV	#L246, -(SP)
	035452	012746	000004	MOV	#4, -(SP)
	035456	010600		MOV	SP, R0
	035460	104416		TRAP	C#PNTS
	035462	062706	000012	ADD	#12, SP
2119	035466			PRINTS	#L247, <B, 71(R1)>, <B, 57(R1)>, 106(R1)
	035466	016146	000106	MOV	106(R1), -(SP)
	035472	005046		CLR	-(SP)
	035474	156116	000057	BISB	57(R1),(SP)
	035500	005046		CLR	-(SP)
	035502	156116	000071	BISB	71(R1),(SP)
	035506	012746	027414	MOV	#L247, -(SP)
	035512	012746	000004	MOV	#1, -(SP)
	035516	010600		MOV	SP, R0
	035520	104416		TRAP	C#PNTS
	035522	062706	000012	ADD	#12, SP
2120	035526			PRINTS	#L248, <B, 72(R1)>, <B, 60(R1)>, 110(R1)
	035526	016146	000110	MOV	110(R1), -(SP)
	035532	005046		CLR	-(SP)
	035534	156116	000060	BISB	60(R1),(SP)
	035540	005046		CLR	-(SP)
	035542	156116	000072	BISB	72(R1),(SP)
	035546	012746	027510	MOV	#L248, -(SP)
	035552	012746	000004	MOV	#4, -(SP)
	035556	010600		MOV	SP, R0
	035560	104416		TRAP	C#PNTS
	035562	062706	000012	ADD	#12, SP
2121	035566			PRINTS	#L249, <B, 73(R1)>, <B, 61(R1)>, 112(R1)
	035566	016146	000112	MOV	112(R1), -(SP)
	035572	005046		CLR	-(SP)
	035574	156116	000061	BISB	61(R1),(SP)
	035600	005046		CLR	-(SP)
	035602	156116	000073	BISB	73(R1),(SP)
	035606	012746	027604	MOV	#L249, -(SP)
	035612	012746	000004	MOV	#4, -(SP)
	035616	010600		MOV	SP, R0
	035620	104416		TRAP	C#PNTS
	035622	062706	000012	ADD	#12, SP
2122	035626			PRINTS	#L250, <B, 74(R1)>, <B, 62(R1)>, 114(R1)
	035626	016146	000114	MOV	114(R1), -(SP)
	035632	005046		CLR	-(SP)
	035634	156116	000062	BISB	62(R1),(SP)
	035640	005046		CLR	-(SP)
	035642	156116	000074	BISB	74(R1),(SP)
	035646	012746	027700	MOV	#L250, -(SP)
	035652	012746	000004	MOV	#4, -(SP)
	035656	010600		MOV	SP, R0
	035660	104416		TRAP	C#PNTS
	035662	062706	000012	ADD	#12, SP
2123	035666			PRINTS	#L251, <B, 75(R1)>, <B, 63(R1)>, 116(R1)
	035666	016146	000116	MOV	116(R1), -(SP)
	035672	005046		CLR	-(SP)
	035674	156116	000063	BISB	63(R1),(SP)

	035700	005046		CLR	-(SP)
	035702	156116	000075	BISB	75(R1),(SP)
	035706	012746	027774	MOV	#L251,-(SP)
	035712	012746	000004	MOV	#4,-(SP)
	035716	010600		MOV	SP,R0
	035720	104416		TRAP	C#PNTS
	035722	062706	000012	ADD	#12,SP
2124	035726			PRINTS	#L252,<B,76(R1)>,<B,64(R1)>,120(R1)
	035726	016146	000120	MOV	120(R1),-(SP)
	035732	005046		CLR	-(SP)
	035734	156116	000064	BISB	64(R1),(SP)
	035740	005046		CLR	-(SP)
	035742	156116	000076	BISB	76(R1),(SP)
	035746	012746	030070	MOV	#L252,-(SP)
	035752	012746	000004	MOV	#4,-(SP)
	035756	010600		MOV	SP,R0
	035760	104416		TRAP	C#PNTS
	035762	062706	000012	ADD	#12,SP
2125	035766			PRINTS	#L253,<B,77(R1)>,<B,65(R1)>,122(R1)
	035766	016146	000122	MOV	122(R1),-(SP)
	035772	005046		CLR	-(SP)
	035774	156116	000065	BISB	65(R1),(SP)
	036000	005046		CLR	-(SP)
	036002	156116	000077	BISB	77(R1),(SP)
	036006	012746	030164	MOV	#L253,-(SP)
	036012	012746	000004	MOV	#4,-(SP)
	036016	010600		MOV	SP,R0
	036020	104416		TRAP	C#PNTS
	036022	062706	000012	ADD	#12,SP
2126	036026			PRINTS	#L254,124(R1)
	036026	016146	000124	MOV	124(R1),-(SP)
	036032	012746	030260	MOV	#L254,-(SP)
	036036	012746	000002	MOV	#2,-(SP)
	036042	010600		MOV	SP,R0
	036044	104416		TRAP	C#PNTS
	036046	062706	000006	ADD	#6,SP
2127	036052			10\$: ENDRPT	
	036052			L10005:	
	036052	104425		TRAP	C#RPT

```

2129          .SBTTL  GLOBAL SUBROUTINES SECTION
2133
2134          ;*****
2135          ;*****
2136          ;
2137          ;GLOBAL SUBROUTINES SECTION
2138          ;   THIS SECTION CONTAINS ALL SUBROUTINES AND
2139          ;   INTERRUPT SERVICE ROUTINES THAT ARE AC-
2140          ;   CESSSED FROM ANYWHERE IN THE PROGRAM.
2141          ;
2142          ;*****
2143          ;*****
2144
2145
2146          ;*****
2147          ;*****
2148          ;
2149          ;TRAP4
2150          ;   THE ADDRESS OF THIS ROUTINE IS LOADED
2151          ;   INTO VECTOR 4 WHENEVER THE PROGRAM IS
2152          ;   ATTEMPTING TO ACCESS A PIECE OF HARDWARE
2153          ;   FOR THE FIRST TIME. IT IS INTENDED TO
2154          ;   CATCH NON-EXISTENT MEMORY TIMEOUTS IN
2155          ;   THE EVENT THE HARDWARE IS NOT REALLY PRE-
2156          ;   SENT OR IS MALFUNCTIONING. IT SIMPLY
2157          ;   SETS A FLAG, INDICATING THE TRAP OCCURRED.
2158          ;
2159          ;*****
2160          ;*****
2164
2165          036054          BGNSRV  TRAP4
                036054          TRAP4::
2166
2167          036054          005237  002276          INC      TRP4FG          ;SET THE FLAG - TRAP OCCURRED
2168
2169          036060          ENDSRV
                036060          L10006:
                036060          000002          RTI
2170
    
```

```

2175
2176
2177 ;*****
2178 ;*****
2179 ;
2180 ;INTRCV
2181 ; THIS IS THE TK50 INTERRUPT HANDLER USED BY THE PPO-
2182 ; GRAM WHEN INTERRUPTS HAVE BEEN ENABLED. IF THE
2183 ; BRFLAG IS CLEAR, THE ROUTINE SETS A FLAG INDICATING
2184 ; THE EXPECTED INTERRUPT OCCURRED. IF BRFLAG IS SET,
2185 ; IT INDICATES THAT PROCESOR PRIORITY WAS SET TO A
2186 ; LEVEL THAT SHOULD HAVE INHIBITED THE INTERRUPT, SO
2187 ; THE ROUTINE SETS AN ERROR INDICATOR.
2188 ;
2189 ;*****
2190 ;*****
2191
2195 036062          BGNSRV  INTRCV
      036062          INTRCV::
2196
2197 ;          BIT      #BRFLAG,LUNFLG(R4)      ;IF NOT PRIORITY LEVEL TESTING
2198 ;          BEQ      5$                        ; THEN SKIP AROUND
2199 ;          MOV      #DRPFLG,LUNFLG(R4)      ; ELSE SET FAILED BIT
2200 ;          BR       EXTINT                    ;RETURN
2201
2202 036062  052764  000002  000014  5$:  BIS      #INTFLG,LUNFLG(R4)      ;SET THE FLAG
2203
2204 036070          EXTINT:
2205 036070          ENDSRV
      036070          L10007:
      036070  000002          RTI
2206
    
```



```
2211  
2212 ;:*****  
2213 ;:*****  
2214 ;  
2215 ;ILLINT  
2216 ; THIS HANDLER ROUTINE'S ADDRESS IS LOADED INTO THE  
2217 ; CURRENT UUT'S VECTOR FOR ALL TESTS THAT DO NOT EN-  
2218 ; ABLE DEVICE INTERRUPTS.  
2219 ;  
2220 ;:*****  
2221 ;:*****  
2225  
2226 036072 BGNSRV ILLINT  
036072 ILLINT::  
2227  
2228 036072 052764 000001 000014 BIS #DRPFLG,LUNFLG(R4)  
2229  
2230  
2231 036100 ENDSRV  
036100 L10010:  
036100 000002 RTI
```

```

2236
2237
2238 ;:*****
2239 ;:*****
2240 ;
2241 ;KTTTEST
2242 ;
2243 ; THIS SUBROUTINE IS USED BY THE INIT CODE TO
2244 ; DETERMINE IF THE MEMORY MANAGEMENT UNIT IS
2245 ; PRESENT. IF SO, IT RETURNS A FLAG IN THE
2246 ; SET STATE. OTHERWISE THE FLAG IS CLEAR IN
2247 ; WHICH CASE TEST SEVEN IS BYPASSED.
2248 ;:*****
2249 ;:*****
2252
2253 KTTTEST::
2254 036102 SETVEC #VEC4,#TRAP4,#PRI07 ;SET UP FOR POSSIBLE NXM
      036102 012746 000340 MOV #PRI07,-(SP)
      036106 012746 036054 MOV #TRAP4,-(SP)
      036112 012746 000004 MOV #VEC4,-(SP)
      036116 012746 000003 MOV #3,-(SP)
      036122 104437 TRAP C$SVEC
      036124 062706 000010 ADD #10,SP
2255 036130 005737 177572 TST MMUSRO ;ARE YOU THERE, MMU?
2256 036134 DELAY 1 ;GIVE NXM TIMEOUT A CHANCE
      036134 012727 000001 MOV #1,(PC)+
      036140 000000 .WORD 0
      036142 013727 002116 MOV L$DLY,(PC)+
      036146 000000 .WORD 0
      036150 005367 177772 DEC -6(PC)
      036154 001375 BNE .-4
      036156 005367 177756 DEC -22(PC)
      036162 001367 BNE . 20

2257
2258 036164 005737 002276 TST TRP4FG ;IF NXM OCCURRED
2259 036170 001026 BNE NOKT ; THEN NO MMU IS PRESENT
2260 036172 005237 002274 INC KTFLAG ; ELSE SAY WE FOUND 18 BIT SO FAR
2261
2262 036176 005737 172516 TST MMUSR3 ;NOW LOOK FOR 22 BIT MAPPING
2263 036202 DELAY 1 ;GIVE NXM A CHANCE
      036202 012727 000001 MOV #1,(PC)+
      036206 000000 .WORD 0
      036210 013727 002116 MOV L$DLY,(PC)+
      036214 000000 .WORD 0
      036216 005367 177772 DEC -6(PC)
      036222 001375 BNE .-4
      036224 005367 177756 DEC -22(PC)
      036230 001367 BNE .-20

2264
2265 036232 005737 002276 TST TRP4FG ;IF NXM OCCURRED
2266 036236 001005 BNE KTEXT ; THEN 18 BIT IS ALL WE'VE GOT
2267 036240 005237 002274 INC KTFLAG ; ELSE SAY WE'VE GOT 22 BIT
2268 036244 000402 BR KTEXT ; AND BRANCH AROUND NEXT
2269
2270 036246 005037 002274 NOKT: CLR KTFLAG ;NO MMU - CLEAR FLAG
2271
2272 036252 KTEXT: CLRVEC #VEC4 ;RESTORE VECTOR
      036252 012700 000004 MOV #VEC4,R0
    
```

036256 104436  
2273 036260 005037 002276  
2274 036264 000207  
2275  
2276

TRAP C3VEC  
CLR TRP4FG  
RTS PC

;MORE HOUSEKEEPING

```
2281  
2282  
2283 ;*****  
2284 ;*****  
2285 ;  
2286 ;RSTVEC  
2287 ; THIS ROUTINE IS CALLED FROM VARIOUS PLACES  
2288 ; IN THE PROGRAM TO SET THE UUT'S INTERRUPT  
2289 ; VECTOR WITH THE ADDRESS OF A HANDLER ROUTINE  
2290 ; WHICH WILL CATCH ILLEGAL DEVICE INTERRUPTS,  
2291 ; SPECIFICALLY "ILLINT". INTERRUPT PRIORITY  
2292 ; IS SET TO 0.  
2293 ;  
2294 ;*****  
2295 ;*****  
2296 ;  
2300 036266 RSTVEC::  
2301 036266 SETVEC TKVEC(R4),#ILLINT,#PRI00  
036266 C.2746 000000 MOV #PRI00,-(SP)  
036272 012746 036072 MOV #ILLINT,-(SP)  
036276 016446 000004 MOV TKVEC(R4),-(SP)  
036302 012746 000003 MOV #3,-(SP)  
036306 104437 TRAP C#SVEC  
036310 062706 000010 ADD #10,SP  
2302  
2303 036314 000207 RTS PC  
2304
```

```

2309
2310
2311 ;*****
2312 ;*****
2313 ;
2314 ;VECTOR
2315 ; THIS ROUTINE IS CALLED FROM VARIOUS PLACES
2316 ; IN THE PROGRAM TO SET THE UUT'S VECTOR WITH
2317 ; THE ADDRESS OF A HANDLER ROUTINE WHEN DEVICE
2318 ; INTERRUPTS HAVE BEEN ENABLED. THE ROUTINE HAS
2319 ; TWO MODES OF OPERATION: WHEN BRFLAG IS CLEAR,
2320 ; PROCESSOR PRIORITY IS SET TO ZERO, ALLOWING
2321 ; DEVICE INTERRUPTS. IF BRFLAG IS SET, PRIORITY
2322 ; IS SET TO 7. IF AN INTERRUPT OCCURS IN THIS
2323 ; CASE, AN ERROR IS RETURNED BY THE HANDLER
2324 ; ROUTINE, "INTRCV".
2325 ;
2326 ;*****
2327 ;*****
2328
2332 036316 VECTOR::
2333 036316 032764 000004 000014 BIT #BRFLAG,LUNFLG(R4) ;IF FLAG IS SET
2334 036324 001014 BNE 5# ; THEN SKIP TO SECOND HALF
2335 036326 SETVEC TKVEC(R4),#INTRCV,#PRI00 ;ELSE LOW PRIORITY
      036326 012746 000000 MOV #PRI00,-(SP)
      036332 012746 036062 MOV #INTRCV,-(SP)
      036336 016446 000004 MOV TKVEC(R4),-(SP)
      036342 012746 000003 MOV #3,-(SP)
      036346 104437 TRAP C#SVEC
      036350 062706 000010 ADD #10,SP
2336 036354 000413 BR EXTVEC ;RETURN
2337
2338 5#: SETVEC TKVEC(R4),#INTRCV,#PRI07 ;HIGH PRIORITY
      036356 012746 000340 MOV #PRI07,-(SP)
      036362 012746 036062 MOV #INTRCV,-(SP)
      036366 016446 000004 MOV TKVEC(R4),-(SP)
      036372 012746 000003 MOV #3,-(SP)
      036376 104437 TRAP C#SVEC
      036400 062706 000010 ADD #10,SP
2339
2340 036404 000207 EXTVEC: RTS PC
2341
2342
    
```

2347  
 2348  
 2349  
 2350  
 2351  
 2352  
 2353  
 2354  
 2355  
 2356  
 2357  
 2358  
 2359  
 2360  
 2361  
 2362  
 2363  
 2364  
 2368  
 2369  
 2370  
 2371  
 2372  
 2373  
 2374  
 2375  
 2376  
 2377  
 2378  
 2379

```

:*****
:*****
:
: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      ; ELSE SET TIMEOUT
PDLYEX: RTS      PC
    
```

```

036406 005037 002330
036412 005337 002324
036416 001373
036420 005337 002326
036424 001002
036426 005237 002330
036432 000207
    
```

```

2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405 036434
2406 036434 005037 002320
2407 036440 012774 000000 000000
2408 036446 012727 000001
      036452 000000
      036454 013727 002116
      036460 000000
      036462 005367 177772
      036466 001375
      036470 005367 177756
      036474 001367
2409
2410 036476 017464 000002 000012
2411 036504 022764 005700 000012
2412
2413 036512 001413
2414 036514 022764 004700 000012
2415 036522 001013
2416 036524 022737 007776 002120
2417 036532 002003
2418 036534 012737 007776 002120
2419
2420 036542 013774 002250 000002 1$:
2421 036550 000402
2422
2423 036552 005237 002320
2424
2425 036556 000207
2426

```

```

;*****
;*****
;
;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$DLY,(PC)+
      .WORD        0
      DEC          -6(PC)
      BNE          -.4
      DEC          -22(PC)
      BNE          -.20
      MOV          @TKSA(R4),TKSASV(R4) ;GET THE SA REG CONTENTS
      CMP          #B.S'D.QB!B.DI!B.OD!B.MP,TKSASV(R4)
      BEQ          1$              ;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
      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

```

```

2431      ;:*****
2432      ;:*****
2433      ;
2434      ;BAKPAT
2435      ;   THIS SUBROUTINE WILL FILL THE COMMUNICATION WITH AN
2436      ;   ALL 1'S DATA PATTERN.  THE LENGTH OF THE AREA IN USE
2437      ;   BY THE CURRENT TEST IS CONTAINED IN "CMARLG".
2438      ;
2439      ;:*****
2440      ;:*****
2441
2442
2443
2444
2445 036560      3AKPAT::
2446 036560      012702 060000      MOV     #COMMBF,R2      ;STARTING ADDRESS OF COMM AREA
2447              ;-20 WORDS
2448 036564      012703 000024      MOV     #20.,R3      ;BUFFER LENGTH IN FRONT OF AREA
2449 036570      006303              ASL     R3           ;MULTIPLIED BY 2
2450 036572      063703 002306      ADD     CMARLG,R3    ;ADD COMM AREA LENGTH USED
2451 036576      012722 177777      1$:    MOV     #-1,(R2)+ ;WRITE THE DATA
2452 036602      005303              DEC     R3           ;IF NOT DONE YET
2453 036604      001374              BNE    1$           ; THEN DO IT AGAIN
2454
2455 036606      000207              RTS     PC
2456
    
```



```

2461
2462
2463      ;:*****^*****
2464      ;:*****
2465
2466      ;
2467      ;CHKCOM
2468      ;   THIS ROUTINE IS CALLED BY TESTS DOING THE PURGE/POLL
2469      ;   CHECK. IT IS USED TO VERIFY THAT THE PORT LEFT THE
2470      ;   COMMUNICATIONS AREA CLEARED. ADDITIONALLY, IT CHECKS
2471      ;   THE 20 WORDS PRECEDING AND SUCCEEDING THE COMM AREA
2472      ;   TO MAKE SURE THE PORT DIDN'T GO OUTSIDE THE COMM AREA.
2473
2474      ;:*****
2475      ;:*****
2476
2477      ;
2478
2479      ;CHKCOM::
2480      ;   MOV     #-1,R1           ;TEST DATA
2481      ;   MOV     #COMMBF,R2      ;STARTING ADDRESS
2482      ;   MOV     #18.,R3        ;FIRST COUNT
2483      ;1$:   CMP     R1,(R2)+     ;IF NOT ALL 1'S
2484      ;       BNE     15$        ; THEN GO REPORT ERROR
2485      ;       DEC     R3         ;IF NOT ALL DONE
2486      ;       BNE     1$        ; THEN GO CHECK ANOTHER
2487
2488      ;   CLR     R1             ;TEST DATA FOR PRINTOUT
2489      ;   MOV     CMARLG,R3      ;SET UP COUNTER FOR COMM AREA
2490      ;5$:   TST     (R2)+       ;IF NOT 0
2491      ;       BNE     15$        ; THEN GO REPORT ERROR
2492      ;       DEC     R3         ;IF NOT ALL DONE
2493      ;       BNE     5$        ; THEN GO CHECK ANOTHER
2494
2495      ;   MOV     #-1,R1           ;TEST DATA FOR PRINTOUT
2496      ;   MOV     #20.,R3        ;SET UP COUNTER FOR POST COMM AREA
2497      ;10$:  CMP     R1,(R2)+     ;IF NOT ALL 1'S
2498      ;       BNE     15$        ; THEN GO REPORT ERROR
2499      ;       DEC     R3         ;IF NOT ALL DONE
2500      ;       BNE     10$       ; THEN GO CHECK ANOTHER
2501      ;       BR     CKCMEX      ; ELSE RETURN
2502
2503      ;15$:  SUB     #2,R2         ;ADJUST ADDRESS FOR PRINTOUT
2504      ;       CMP     #8.,L$TEST  ;IF IN TEST 8
2505      ;       BEQ     20$        ; THEN DO ALTERNATE PRINTOUT
2506      ;       ERRDF  1,MSG14,PRIPAD ;"PURGE/POLL TEST FAILED"
2507      ;       TRAP  C$ERDF
2508      ;       .WORD  1
2509      ;       .WORD  MSG14
2510      ;       .WORD  PRIPAD
2511      ;       BR     25$        ;COMMON EXIT
2512
2513      ;20$:  ERRDF  2,MSG15,PRIVAD ;"EXTENDED ADDRESS TEST FAILED"
2514      ;       TRAP  C$ERDF
2515      ;       .WORD  2
2516      ;       .WORD  MSG15
2517      ;       .WORD  PRIVAD
2518
2519      ;25$:  DODU   LOGUNT
2520      ;       MOV     LOGUNT,R0
    
```

2512	036736	104451	TRAP	C:DODU
2513	036740	000207	CKCMEX: RTS	PC
2514				

```

2519
2520
2521 ;:*****
2522 ;:*****
2523 ;
2524 ;INTMMU
2525 ;
2526 ; THIS SUBROUTINE IS CALLED FROM TEST 8 TO INITIALIZE
2527 ; MEMORY MANAGEMENT REGISTERS. ALL PAR'S EXCEPT ONE
2528 ; ARE SET UP TO MAP VIRTUAL ADDRESSES INTO THE LOWEST
2529 ; 32K OF PHYSICAL MEMORY. KPAR7 IS SET UP TO MAP TO
2530 ; THE I/O PAGE. THE PAR REGISTER THAT CORRESPONDS TO
2531 ; THE VIRTUAL ADDRESS OF THE COMMUNICATION AREA IS SET
2532 ; UP TO POINT TO THE SECOND 32K OF PHYSICAL MEMORY.
2533 ; ALL PDR'S ARE INITIALIZED TO THE SAME VALUE; NAMELY,
2534 ; UPWARD EXPANDABLE, READ/WRITE ACCESS ENABLED, AND THE
2535 ; FULL 8KBYTE PAGE IS ACCESSIBLE.
2536 ;:*****
2537 ;:*****
2541
2542 036742 INTMMU::
2543 036742 012703 172300 MOV #KPDRO,R3 ;START OF PDR ADDRESS RANGE
2544 036746 012702 172340 MOV #KPAR0,R2 ;START OF PAR ADDRESS RANGE
2545 036752 005001 CLR R1 ;STARTING RELOCATION VALUE
2546
2547 036754 010122 1$: MOV R1,(R2)+ ;LOAD RELOCATION VALUE
2548 036756 012723 077406 MOV #77406,(R3)+ ;LOAD PDR
2549 036762 062701 000200 ADD #200,R1 ;ADJUST RELOCATION VALUE
2550 036766 022701 002000 CMP #2000,R1 ;IF NOT AT THE END
2551 036772 001370 BNE 1$ ; THEN DO ANOTHER ONE
2552
2553 036774 010137 172346 MOV R1,KPAR3 ; ELSE SET THIS REG TO NEXT 32K
2554 037000 012737 007600 172356 MOV #7600,KPAR7 ;18 BIT I/O PAGE
2555 037006 032737 000002 002274 BIT #BIT1,KTFLAG ;IF 22-BIT BUS NOT AVAILBLE
2556 037014 001406 BEQ 2$ ; THEN GO TURN MMU ON
2557 037016 012737 177600 172356 MOV #177600,KPAR7 ; ELSE SET 22 BIT I/O PAGE
2558 037024 012737 000020 172516 MOV #MM22ON,MMUSR3 ; AND ENABLE 22 BIT MAPPING
2559
2560 037032 012737 000001 177572 2$: MOV #MMON,MMUSRO ;TURN ON THE WHOLE THING
2561 037040 000207 RTS PC
2562
2563
2564 037042 PRTINT::
2565 037042 010174 000000 MOV R1,@TKIP(R4) ;INITIALIZE THE DRIVE
2566 037046 012703 037264 MOV #INTTBL,R3 ;PUT THE TABLE ADDRESS INTO R3
2567 037052 012701 004000 MOV #S1,R1 ;SET UP TO BEGIN AT STEP 1
2568 037056 005037 002316 CLR INISTP ;CLEAR THE STEP TRACKER
2569 037062 012737 000030 003024 LOOP: MOV #24.,CNTHI ;SET UP THE TIME OUT COUNTER
2570 037070 005002 CLR R2 ;CLEAR R2
2571 037072 005202 ILOOP: INC R2 ;INCREMENT HI TIME OUT VALUE ?
2572 037074 001016 BNE 2$ ;IF NOT, BRANCH
2573 037076 005337 003024 DEC CNTHI ;ELSE, DECREMENT LO TIMEOUT
2574 037102 001013 BNE 2$ ;BRANCH IF NO TIME OUT
2575 037104 017464 000002 000012 MOV @TKSA(R4),TKSASV(R4) ;SAVE THE SA FOR THE ERROR PRINTOUT
2576 037112 ERRDF 51.,WRER1,WRINTO ;PRINT PORT INIT FAILURE
037112 104455 TRAP C$ERDF
037114 000063 .WORD 51
    
```

```

037116 032347
037120 033176
2577 037122 002312
037122 013700 002312
037126 104451
2578 037130 000454
2579 037132 037401 000002 2$:
2580 037136 001755
2581 037140 032774 100000 000002
2582 037146 001413
2583 037150 017464 000002 000012
2584 037156
037156 104455
037160 000064
037162 032401
037164 033222
2585 037166
037166 013700 002312
037172 104451
2586 037174 000432
2587 037176 005237 002316 3$:
2588 037202 012374 000002
2589 037206 006301
2590 037210 100324
2591 037212 012702 003004
2592 037216 012703 002574
2593 037222 010322 5$:
2594 037224 012722 100000
2595 037230 062703 000104
2596 037234 022703 003004
2597 037240 001370
2598 037242 012737 003004 023040
2599 037250 012737 003014 023036
2600 037256 005037 003032
2601 037262 000207 100$:
2602
2603
2604 037264 104400
2605 037266 003004
2606 037270 000000
2607 037272 000001

.WORD WRER1
.WORD WRINTO
DODU LOGUNT ;DROP THE UNIT
MOV LOGUNT,R0
TRAP C$DODU
BR 100$ ;EXIT ROUTINE
BIT @TKSA(R4),R1 ;TEST FOR STEP BIT FROM DRIVE
BEQ ILOOP ;LOOP UNTIL SOMETHING SETS
BIT #ERR,@TKSA(R4) ;CHECK FOR ERROR
BEQ 3$ ;NO ERROR, KEEP GOING
MOV @TKSA(R4),TKSASV(R4) ;SAVE THE SA CONTENTS
ERRDF 52.,WRER2,WRPRTE ;PRINT ERROR
TRAP C$ERDF
.WORD 52
.WORD WRER2
.WORD WRPRTE
DODU LOGUNT ;DROP THE UNIT
MOV LOGUNT,R0
TRAP C$DODU
BR 100$ ;EXIT ROUTINE
INC INISTP ;INCREMENT THE STEP TRACKER
MOV (R3)+,@TKSA(R4) ;WRITE WORD FROM TABLE TO CONTROLLER
ASL R1 ;SHIFT TO NEXT STEP
BPL LOOP ;IF NOT AT LAST STEP LOOP
MOV #RSPRNG,R2 ;PUT THE RESPONSE DESCRIPTOR ADDR IN R2
MOV #RSPBUF,R3 ;PUT THE RESPONSE BUFFER ADDRESS IN R3
MOV R3,(R2)+ ;PUT THE BUFF ADDR IN THE DESCRIPTOR
MOV #OWN,(R2)+ ;SET THE DESCRIPTOR TO THE CONTROLLER
ADJ #RSPSTP,R3 ;STEP TO THE NEXT BUFFER SLOT
CMP #RSPEND,R3 ;ARE WE AT THE END OF THE BUFFER ?
BNE 5$ ;NO, KEEP GOING
MOV #RSPRNG,RSPSAV ;SET UP TO USE FIRST RESPONSE BUFFER
MOV #CMDRNG,CMDSAV ;SET UP TO USE FIRST COMMAND BUFFER
CLR CMDREF ;SET THE COMMAND REFERENCE # TO 0
RTS PC ;RETURN

;INIT DATA TABLE
INTTBL: .WORD 104400
        .WORD RSPRNG
        .WORD 0
        .WORD GO
    
```

2609	037274				CLSDRV::		
2610	037274	005237	003032		5\$: INC	CMDREF	;ADD 1 TO THE COMMAND REFERENCE NUMBER
2611	037300	004737	037632		JSR	PC,PRTDRV	;GO SEND THE COMMAND
2612	037304	032764	000001	000014	BIT	#DRPFLG,LUNFLG(R4)	;IS THE DRIVE AVAILABLE
2613	037312	001146			BNE	100\$	;GET OUT IF NOT AVAILABLE
2614	037314	004737	037774		JSR	PC,CDRECV	;GO CHECK FOR ANY NEW RESPONSES
2615	037320	032764	000001	000014	BIT	#DRPFLG,LUNFLG(R4)	;IS THE DRIVE AVAILABLE
2616	037326	001140			BNE	100\$	;GET OUT IF NOT AVAILABLE
2617	037330	012702	024035		MOV	#WR2,R2	;R2 GETS ERROR MESSAGE ADDRESS
2618	037334	005763	000012		TST	P.STS(R3)	;WAS STATUS "NORMAL"?
2619	037340	001463			BEQ	15\$	;YES - BRANCH
2620	037342	032764	000010	000014	7\$: BIT	#DUPFLG,LUNFLG(R4)	;IS IT A DUP COMMAND ?
2621	037350	001047			BNE	12\$	;YES, PRINT DUP ERROR
2622	037352	022705	002406		CMP	#ONLINE,R5	;IS IT AN ONLINE COMMAND
2623	037356	001024			BNE	10\$	;NO, GO PRINT ERROR
2624	037360	122763	000003	000012	CMPB	#3,P.STS(R3)	;WAS IT A UNIT UNKNOWN ERROR ?
2625	037366	001003			BNE	8\$	;NO, CONTINUE
2626	037370	012702	024214		MOV	#WR7,R2	;R2 GETS ERROR MESSAGE ADDRESS
2627	037374	000415			BR	10\$	;PRINT ERROR MESSAGE
2628	037376	122763	000043	000012	8\$: CMPB	#43,P.STS(R3)	;WAS IT A NO MEDIA MOUNTED ERROR ?
2629	037404	001003			BNE	9\$	;NO, CONTINUE
2630	037406	012702	024265		MOV	#WR8,R2	;R2 GETS ERROR MESSAGE ADDRESS
2631	037412	000406			BR	10\$	;PRINT ERROR MESSAGE
2632	037414	122763	000103	000012	9\$: CMPB	#103,P.STS(R3)	;WAS IT A UNIT INOPERATIVE ERROR ?
2633	037422	001012			BNE	11\$	;NO, MUST BE INVALID STATUS
2634	037424	012702	024265		MOV	#WR8,R2	;R2 GETS ERROR MESSAGE ADDRESS
2635	037430				10\$: ERRDF	57.,WRER3,WRCMDE	;GET READY TO PRINT FAILURE
	037430	104455			TRAP	C\$ERDF	
	037432	000071			.WORD	57	
	037434	032422			.WORD	WRER3	
	037436	033256			.WORD	WRCMDE	
2636	037440				DODU	LOGUNT	;DROP THE UNIT
	037440	013700	002312		MOV	LOGUNT,R0	
	037444	104451			TRAP	C\$DODU	
2637	037446	000470			BR	100\$	;GET OUT ON ERROR
2638	037450				11\$: ERRDF	58.,WRER7,WRCMDE	;GET READY TO PRINT FAILURE
	037450	104455			TRAP	C\$ERDF	
	037452	000072			.WORD	58	
	037454	032546			.WORD	WRER7	
	037456	033256			.WORD	WRCMDE	
2639	037460				DODU	LOGUNT	;DROP THE UNIT
	037460	013700	002312		MOV	LOGUNT,R0	
	037464	104451			TRAP	C\$DODU	
2640	037466	000460			BR	100\$	;GET OUT ON ERROR
2641	037470				12\$: ERRDF	59.,WRER8,WRCMDE	;GET READY TO PRINT FAILURE
	037470	104455			TRAP	C\$ERDF	
	037472	000073			.WORD	59	
	037474	032614			.WORD	WRER8	
	037476	033256			.WORD	WRCMDE	
2642	037500				DODU	LOGUNT	;DROP THE UNIT
	037500	013700	002312		MOV	LOGUNT,R0	
	037504	104451			TRAP	C\$DODU	
2643	037506	000450			BR	100\$	;GET OUT ON ERROR
2644	037510	022705	002456		15\$: CMP	#GDUST,R5	;WAS IT A GET DUST STATUS COMMAND ?
2645	037514	001035			BNE	20\$	;NO, CONTINUE
2646	037516	126327	000014	000113	CMPB	P.EXT1(R3),#113	;1ST BYTE OF PROGRAM EXTENSION = "K" ?
2647	037524	001341			BNE	10\$	;NO, ERROR

2648	037526	126327	000015	000065	CMPB	P.EXT2(R3),#65	;2ND BYTE OF PROGRAM EXTENSION = '5' ?	
2649	037534	001335			BNE	10\$	;NO, ERROR	
2650	037536	126327	000016	000060	CMPB	P.EXT3(R3),#60	;3RD BYTE OF PROGRAM EXTENSION = '0' ?	
2651	037544	001331			BNE	10\$	;NO, ERROR	
2652	037546	126327	000017	000007	CMPB	DUSTFL(R3),#7	;FLAGS = 7 ?	
2653	037554	001325			BNE	10\$	;NO, ERROR	
2654	037556	026327	000020	000000	CMP	P.IND1(R3),#0	;1ST WORD OF PROGRESS INDICATOR = 0 ?	
2655	037564	001321			BNE	10\$	;NO, ERROR	
2656	037566	026327	000022	000000	CMP	P.IND2(R3),#0	;2ND WORD OF PROGRESS INDICATOR = 0 ?	
2657	037574	001315			BNE	10\$	;NO, ERROR	
2658	037576	026327	000024	000012	CMP	TIMOUT(R3),#12	;TIMEOUT VALUE = 12 ?	
2659	037604	001311			BNE	10\$	;NO, ERROR	
2660	037606	000405			BR	30\$	;NO ERRORS, RETURN	
2661	037610	022705	002524		20\$:	CMP	#RCVDAT,R5	;WAS IT A RECEIVE DATA COMMAND?
2662	037614	001002			BNE	30\$	;NO, CONTINUE	
2663	037616	004737	040224		JSR	PC,L2DATA	;YES, GO READ DIAGL2 PROGRAM STATUS	
2664	037622	005337	002272		30\$:	DEC	ITRCNT	;SUBTRACT 1 FROM TIMES TO DO
2665	037626	001222			BNE	5\$	;GO SEND IT AGAIN	
2666	037630	000207			100\$:	RTS	PC	;RETURN
2667								
2668	037632							
2669	037632	013701	023036			PRTDRV::		
2670	037636	013765	003032	000000	MOV	CMDSAV,R1	;SET UP COMMAND RING POINTER	
2671	037644	112765	000002	177776	MOV	CMDREF,P.CRF(R5)	;PUT COMMAND REFERENCE # INTO PACKET	
2672	037652	032764	000010	000014	MOVB	#2,CRD(R5)	;PUT THE CREDIT LIMIT INTO THE PACKET	
2673	037660	001404			BIT	#DUPFLG,LUNFLG(R4)	;IS IT A DUP COMMAND?	
2674	037662	112765	000002	177777	BEQ	1\$	;NO, CONNECTION TYPE = 1	
2675	037670	000403			MOVB	#2,CONID(R5)	;YES,CONNECTION TYPE = 2	
2676	037672	112765	000001	177777	BR	2\$	;CONTINUE	
2677	037700	010511			1\$:	MOVB	#1,CONID(R5)	;PUT THE CONNECTION TYPE INTO THE PACKET
2678	037702	012761	100000	000002	2\$:	MOV	R5,(R1)	;PUT THE PACKET ADDRESS INTO THE DESCRIPTOR
2679	037710	005774	000000		MOV	#OWN,HIADDR(R1)	;SET THE OWNERSHIP BIT OF THE DESCRIPTOR	
2680	037714	005774	000002		TST	@TKIP(R4)	;READ THE IP REGISTER	
2681	037720	100013			TST	@TKSA(R4)	;READ THE SA REGISTER	
2682	037722	017464	000002	000012	BPL	10\$	;BRANCH IF NO ERRORS	
2683	037730				MOV	@TKSA(R4),TKSASV(R4)	;SAVE THE SA FOR THE ERROR PRINTOUT	
	037730	104455			ERRDF	53.,WRER4,WRPRT	;PRINT PORT DETECTED ERROR	
	037732	000065			TRAP	C\$ERDF		
	037734	032450			.WORD	53		
	037736	033222			.WORD	WRER4		
					.WORD	WRPRT		
2684	037740				DODU	LOGUNT	;DROP THE UNIT	
	037740	013700	002312		MOV	LOGUNT,R0		
	037744	104451			TRAP	C\$DODU		
2685	037746	000411			BR	100\$	;GET OUT	
2686	037750	062701	000004		10\$:	ADD	#RNGSTP,R1	;ADJUST RESPONSE POINTER FOR NEXT TIME
2687	037754	022701	003024		CMP	#DSCEND,R1	;ARE WE AT THE END ?	
2688	037760	001002			BNE	15\$	;NO, GET OUT	
2689	037762	012701	003014		MOV	#CMDRNG,R1	;SET R1 TO TOP BUFFER	
2690	037766	010137	023036		15\$:	MOV	R1,CMDSAV	;SAVE THE COMMAND RING LOCATION
2691	037772	000207			100\$:	RTS	PC	;RETURN

```

2693 037774          CORECV::
2694 037774 004737 040102 1$: JSR    PC,PDRECV          ;CALL PORT DRIVER RECEIVE
2695 040000 032764 000071 000014 BIT    #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
2696 040006 001034          100$ BNE    100$                ;GET OUT IF NOT AVAILABLE
2697 040010 011103          MOV    (R1),R3            ;SET UP RESPONSE BUFFER POINTER
2698 040012 026365 000000 000000 CMP    P.CRF(R3),P.CRF(R5) ;IS THIS THE COMMAND THAT IS EXPECTED ?
2699 040020 001015          BNE    10$                ;GET OUT IF WRONG RESPONSE
2700 040022 012761 100000 000002 MOV    #OWN,HIADDR(R1)    ;GIVE THE CONTROLLER THE RING BACK
2701 040030 062701 000004          ADD    #RNGSTP,R1        ;ADJUST RESPONSE POINTER FOR NEXT TIME
2702 040034 0.2701 003014          CMP    #CMDRNG,R1        ;ARE WE AT THE END ?
2703 040040 001002          BNE    15$                ;NO, GET OUT
2704 040042 012701 003004          MOV    #RSPRNG,R1        ;SET R1 TO TOP BUFFER
2705 040046 010137 023040 15$: MOV    R1,RSPSAV          ;SAVE THE POINTER FOR NEXT TIME
2706 040052 000412          BR     100$
2707 040054 012737 032640 002310 10$: MOV    #CTRL,FRUIS        ;PRINT CONTROLLER ERROR
2708 040062          ERRDF 56.,WRER5,WRSEQE ;PRINT COMMAND OUT OF SEQUENCE ERROR
      040062 104455          TRAP C$ERDF
      040064 000070          .WORD 56
      040066 032474          .WORD WRER5
      040070 033410          .WORD WRSEQE
2709 040072          DODU  LOGUNT          ;DROP THE UNIT
      040072 013700 002312          MOV    LOGUNT,RO
      040076 104451          TRAP C$DODU
2710 040100 000207          100$: RTS    PC          ;RETURN
2711
2712
2713
2714 040102          PDRECV::
2715 040102 013701 023040          MOV    RSPSAV,R1        ;PUT THE RESPONSE RING SAVE IN R1
2716 040106 012737 000225 003024 1$: MOV    #225,CNTHI        ;SET UP THE TIME OUT COUNTER
2717 040114 005002          CLR    R2                ;CLEAR R2
2718 040116 005202          5$: INC    R2            ;INCREMENT HI TIME OUT VALUE ?
2719 040120 001013          BNE    10$                ;IF NOT, BRANCH
2720 040122 005337 003024          DEC    CNTHI            ;ELSE, INCREMENT HI TIMEOUT
2721 040126 001010          BNE    10$                ;KEEP GOING ,NO TIME OUT YET
2722 040130          ERRDF 54.,WRER6,WRTOE ;PRINT COMMAND TIMEOUT MESSAGE
      040130 104455          TRAP C$ERDF
      040132 000066          .WORD 54
      040134 032525          .WORD WRER6
      040136 033466          .WORD WRTOE
2723 040140          DODU  LOGUNT          ;GO DROP THE UNIT
      040140 013700 002312          MOV    LOGUNT,RO
      040144 104451          TRAP C$DODU
2724 040146 000425          BR     100$                ;GET OUT ON ERROR
2725 040150 017464 000002 000012 10$: MOV    #TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
2726 040156 005764 000012          TST    TKSASV(R4)        ;CHECK FOR SA ERROR
2727 040162 100013          BPL    20$                ;BRANCH IF NO ERROR
2728 040164 012737 032640 002310 MOV    #CTRL,FRUIS        ;LOAD FAILING FRU
2729 040172          ERRDF 55.,EMSG9,WRPRTE ;PRINT SA CONTENTS IN ERROR MESSAGE
      040172 104455          TRAP C$ERDF
      040174 000067          .WORD 55
      040176 030465          .WORD EMSG9
      040200 033222          .WORD WRPRTE
2730 040202          DODU  LOGUNT          ;GO DROP THE UNIT
      040202 013700 002312          MOV    LOGUNT,RO
      040206 104451          TRAP C$DODU
2731 040210 000404          BR     100$                ;GET OUT ON ERROR
    
```

```

2732 040212 032761 100000 000002 20$: BIT #OWN,HIADDR(R1) ;IS THE SLOT SET TO US ?
2733 040220 001336 BNE 5$ ;KEEP GOING TILL TIMEOUT OR SUCCESS
2734 040222 000207 100$: RTS PC ;RETURN
2735
2736
2737
2738
2739 040224 L2DATA::
2740 040224 010146 MOV R1,(SP) ;SAVE REGISTERS
2741 040226 010246 MOV R2,-(SP)
2742 040230 010346 MOV R3,-(SP)
2743 040232 012701 060000 MOV #RDBUF,R1 ;GET START ADDRESS OF RECEIVE DATA
2744 040236 121127 000001 CMPB (R1),#1 ;ANY NEW INFORMATION ?
2745 040242 001162 BNE EXIT ;NO, RETURN
2746 040244 026127 000002 000000 CMP L2STA(R1),#0 ;YES, LOCAL PROGRAM FINISHED ?
2747 040252 001552 BEQ GOABO ;YES, SET ABORT FLAG
2748 040254 016103 000002 MOV L2STA(R1),R3 ;GET LEVEL 2 PROGRAM STATUS
2749 040260 042703 177740 BIC #177740,R3 ;MASK OFF UPPER 11 BITS OF STATUS
2750 040264 022703 000037 CMP #37,R3 ;L2 MESSAGE IF 1ST 5 BITS SET
2751 040270 001113 BNE MSCPER ;TMSCP COMMAND ERROR OTHERWISE
2752 040272 016103 000002 MOV L2STA(R1),R3 ;GET LEVEL 2 PROGRAM STATUS
2753 040276 012737 032640 002310 MOV #CTRL,FRUIS ;NO, SET FAILING FRU TO CONTROLLER
2754 040304 042703 000037 BIC #000037,R3 ;MASK OFF UNUSED BITS OF STATUS
2755 040310 006203 ASR R3
2756 040312 006203 ASR R3
2757 040314 006203 ASR R3
2758 040316 006203 ASR R3
2759 040320 006203 ASR R3
2760
2761 040322 022703 000001 CMP #1.,R3 ;IS IT A DATA COMPARE ERROR ?
2762 040326 001006 BNE 5$ ;NO TRY AGAIN
2763 040330 012702 023312 MOV #DCERR,R2 ;SET UP TO PRINT ERROR
2764 040334 012737 032655 002310 MOV #DRVE,FRUIS ;SET FRU CALLOUT TO DRIVE
2765 040342 000454 BR 40$ ;GO PRINT IT
2766
2767 040344 022703 000002 5$: CMP #2.,R3 ;IS IT A CONTROLLER ERROR ?
2768 040350 001003 BNE 10$ ;NO TRY AGAIN
2769 040352 012702 023322 MOV #CNTERR,R2 ;SET UP TO PRINT ERROR
2770 040356 000446 BR 40$ ;GO PRINT IT
2771
2772 040360 022703 000012 10$: CMP #10.,R3 ;IS IT AN INVAILID STATUS ERROR ?
2773 040364 001003 BNE 15$ ;NO TRY AGAIN
2774 040366 012702 023332 MOV #INVSTA,R2 ;SET UP TO PRINT ERROR
2775 040372 000440 BR 40$ ;GO PRINT IT
2776
2777 040374 022703 000024 15$: CMP #20.,R3 ;IS IT A BAD PATTERN NUMBER ERROR ?
2778 040400 001003 BNE 20$ ;NO TRY AGAIN
2779 040402 012702 023342 MOV #BPNERR,R2 ;SET UP TO PRINT ERROR
2780 040406 000432 BR 40$ ;GO PRINT IT
2781
2782 040410 022703 000025 20$: CMP #21.,R3 ;IS IT A RESPONSE ADDRESS ERROR ?
2783 040414 001003 BNE 25$ ;NO TRY AGAIN
2784 040416 012702 023352 MOV #RSPADD,R2 ;SET UP TO PRINT ERROR
2785 040422 000424 BR 40$ ;GO PRINT IT
2786
2787 040424 022703 000026 25$: CMP #22.,R3 ;IS IT A HOST BUFFER ADDRESS ERROR ?
2788 040430 001003 BNE 30$ ;NO TRY AGAIN
    
```



```

2789 040432 012702 023362      MOV    #HBFADD,R2      ;SET UP TO PRINT ERROR
2790 040436 000416              BR     40$            ;GO PRINT IT
2791
2792 040440 022703 000027      30$:  CMP    #23.,R3      ;IS IT A UNKNOWN ERROR LOG RECIEVED ?
2793 040444 0010C3              BNE   35$            ;NO TRY AGAIN
2794 040446 012702 023372      MOV    #UNERLG,R2     ;SET UP TO PRINT ERROR
2795 040452 000410              BR     40$            ;GO PRINT IT
2796
2797 040454 022703 000030      35$:  CMP    #24.,R3      ;IS IT A RESPONSE TIME OUT ERROR ?
2798 040460 001003              BNE   36$            ;NO TRY AGAIN
2799 040462 012702 023402      MOV    #RSPT0,R2     ;SET UP TO PRINT ERROR
2800 040466 000402              BR     40$            ;GO PRINT IT
2801
2802 040470 012702 023332      36$:  MOV    #INVSTA,R2   ;SET UP TO PRINT ERROR
2803
2804 040474 012237 023052      40$:  MOV    (R2)+,ERRTYP  ;LOAD ERROR TYPE
2805 040500 012237 023054      MOV    (R2)+,ERRNBR  ;LOAD ERROR NUMBER
2806 040504 012237 023056      MOV    (R2)+,ERRMSG  ;LOAD ERROR MESSAGE ADDRESS
2807 040510 012237 023060      MOV    (R2)+,ERRBLK  ;LOAD ERROR SUBROUTINE ADDRESS
2808 040514              ERROR              ;CALL "ERROR" MACRO
2809 040514 104460              TRAP   C$ERROR
2810 040516 000430              BR     GOABO
2810 040520 022703 000012      MSCPER: CMP    #12,R3    ;ABORT MICRODIAGNOSTIC
2811 040524 001406              BEQ   10$            ;IS IT A CONTROLLER ERROR ?
2812 040526 022703 000014      CMP    #14,R3        ;YES, SET FAILING FRU TO CONTROLLER
2813 040532 001403              BEQ   10$            ;IS IT A FORMATTER ERROR ?
2814 040534 012737 032655 002310 5$:  MOV    #DRVE,FRUIS   ;YES, LEAVE FAILING FRU AS CONTRLLOER
2815 040542 005303              10$:  DEC    R3            ;NO, SET FAILING FRU TO DRIVE
2816 040544 070327 000010      MUL    #10,R3        ;R3 IS OFFSET INTO TMSCP ERROR TABLE
2817 040550 012702 023062      MOV    #L2ETBL,R2   ;GET TMSCP ERROR TABLE START ADDRESS
2818 040554 060302              ADD   R3,R2          ;R2 POINTS TO ENTRY IN TMSCP ERROR TABLE.
2819 040556 012237 023052      MOV    (R2)+,ERRTYP  ;LOAD ERROR TYPE
2820 040562 012237 023054      MOV    (R2)+,ERRNBR  ;LOAD ERROR NUMBER
2821 040566 012237 023056      MOV    (R2)+,ERRMSG  ;LOAD ERROR MESSAGE ADDRESS
2822 040572 012237 023060      MOV    (R2)+,ERRBLK  ;LOAD ERROR SUBROUTINE ADDRESS
2823 040576              ERROR              ;CALL "ERROR" MACRO
2824 040576 104460              TRAP   C$ERROR
2824 040600 052764 000020 000014 GOABO: BIS    #ABTFLG,LUNFLG(R4) ;SET ABORT FLAG
2825 040606              DORPT
2825 040606 104424              TRAP   C$DRPT        ;DUMP NORMAL COMPLETION STATISTICS
2826 040610 012603              EXIT:  MOV    (SP)+,R3 ;RESTORE REGISTERS
2827 040612 012602              MOV    (SP)+,R2
2828 040614 012601              MOV    (SP)+,R1
2829 040616 000207              RTS    PC
2830
2831 040620              ENDMOD
2842              .TITLE MISCELLANEOUS SECTIONS
2843              .SBTTL REPORT CODING SECTION
2871
2872 040620              BGNMOD
2873
    
```

```

2875          .SBTTL  INITIALIZE SECTION
2876
2877          ;**
2878          ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
2879          ; AT THE BEGINNING OF EACH PASS.
2880          ;--
2881
2882 040620          BGNINIT
2883 040620          L$INIT::
2884
2885 040620          READEF  #EF.START          ;IF THIS IS A FRESH START
2886 040620 012700 000040          MOV    #EF.START,R0
2887 040624 104447          TRAP   C$REFG
2888 040626          BCOMPLETE  START          ; THEN GO TO START
2889 040626 103421          BCS    START
2890
2891 040630          READEF  #EF.RESTART        ;IF THIS IS A RESTART
2892 040630 012700 000037          MOV    #EF.RESTART,R0
2893 040634 104447          TRAP   C$REFG
2894 040636          BCOMPLETE  START          ; THEN GO TO START
2895 040636 103415          BCS    START
2896
2897 040640          READEF  #EF.PWR           ;IF POWER-FAIL OCCURRED
2898 040640 012700 000034          MOV    #EF.PWR,R0
2899 040644 104447          TRAP   C$REFG
2900 040646          BCOMPLETE  START          ; THEN START FROM THE BEGINNING
2901 040646 103411          BCS    START
2902
2903 040650          READEF  #EF.NEW           ;IF THIS IS A NEW PASS
2904 040650 012700 000035          MOV    #EF.NEW,R0
2905 040654 104447          TRAP   C$REFG
2906 040656          BCOMPLETE  NUPASS        ; THEN SKIP START UP CODE
2907 040656 103422          BCS    NUPASS
2908
2909 040660          READEF  #EF.CONTINUE       ;IF THIS IS A CONTINUE
2910 040660 012700 000036          MOV    #EF.CONTINUE,R0
2911 040664 104447          TRAP   C$REFG
2912 040666          BCOMPLETE  END           ; THEN SKIP ALL INIT CODE
2913 040666 103465          BCS    END
2914
2915 040670          BR      NEXT              ;JUST HERE FOR NEXT UUT
2916
2917          START:
2918 040672          MOV    #0,PASCNT          ;INITIALIZE PASS COUNT
2919 040672 012737 000000 002270          CLR    KTFLAG
2920 040700 005037 002274          MOV    #LUNBLK,R4
2921 040704 012704 002212          ;R4 WILL ALWAYS POINT TO LUNBLK
2922 040710 022737 001400 002120          CMP    #1400,L$HIME
2923 040716 103002          ;IF <= 28KWORDS OF MEMORY PRESENT
2924 040720 004737 036102          BHS    NUPASS
2925 040720 004737 036102          ; THEN SKIP NEXT
2926 040720 004737 036102          JSR    PC,KTTEST
2927 040720 004737 036102          ; ELSE SEE IF MMJ IS PRESENT
2928
2929          NUPASS: BRESET
2930 040724          TRAP   C$RESET          ;CLEAR THE WORLD
2931 040724 104433          TRAP   C$RESET
2932 040726 005237 002270          INC    PASCNT
2933 040732 012737 177777 002312          ;UPDATE THE PASS COUNT
2934 040732 012737 177777 002312          MOV    #1,LOGUNT
2935 040732 012737 177777 002312          ;INITIALIZE LOGICAL UNIT COUNT
2936
2937          NEXT:  INC    LOGUNT
2938 040740 005237 002312          ;POINT TO NEXT UUT
    
```

```

2915 040744 023737 002312 002012      CMP      LOGUNT,L$UNIT      ;IF WE'VE PASSED MAXIMUM UUT'S
2916 040752 0C1433                      BEQ      END                ; THEN LEAVE INIT
2917
2918 040754                      GPWARD  LOGUNT,RO          ;GET P-TABLE FOR THIS UNIT
      040754 013700 002312      MOV      LOGUNT,RO
      040760 104442      TRAP    C$GPWRD
2919 040762                      BNCOMPLETE      NEXT      ;TRY AGAIN
      040762 103366      BCC     NEXT
2920
2921 040764 011064 000000      MOV      (R0),TKIP(R4)      ;PUT IP REG ADDRESS IN LUNBLK
2922 040770 012064 000002      MOV      (R0)+,TKSA(R4)    ; AND ANOTHER COPY IN LUNBLK
2923 040774 062764 000002 000002      ADD     #2,TKSA(R4)        ;MAKE IT THE SA REG ADDRESS
2924 041002 012064 000004      MOV      (R0)+,TKVEC(R4)   ;GET THE VECTOR INTO THE LUNBLK
2925 041006 011064 000006      MOV      (R0),MSCPUN(R4)   ;PUT THE T/MSCP UNIT # IN LUNBLK
2926 041012 004737 036266      JSR     PC,RSTVEC         ;SET UUT VECTOR FOR ILLEGAL INTRPTS.
2927 041016                      PRINTF  #IMSG,LOGUNT        ;"TESTING UNIT N"
      041016 013746 002312      MOV      LOGUNT,-(SP)
      041022 012746 041046      MOV      #IMSG,-(SP)
      041026 012746 000002      MOV     #2,-(SP)
      041032 010600      MOV     SP,RO
      041034 104417      TRAP    C$PNTF
      041036 062706 000006      ADD     #6,SP
2928
2929 041042                      END:
2930 041042                      EXIT   INIT
      041042 104432      TRAP    C$EXIT
      041044 000032      .WORD  L10011-.
2931
2943 041046      045      116      045  IMSG:  .ASCIZ  ?#N#ATESTING UNIT #D1#N?
2944                      .EVEN
2945
2946 041076                      ENDINIT
      041076                      L10011:
      041076 104411      TRAP    C$INIT
    
```

```

2948      .SBTTL  CLEANUP CODING SECTION
2949
2950      ;**
2951      ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
2952      ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
2953      ; -
2954
2955      BGNCLN
           L$CLEAN::
2956
2963      041100  032764  000000G  002212      BIT    @T9FLAG,LUNBLK(R4)      ;IF NOT HERE FROM TEST 9
2964      041106  001400                      BEQ    ENDCLE                      ; THEN SKIP THE REST
2965
2966      ;EVENTUALLY MORE CODE WILL BE PLACED HERE TO GUARANTEE THAT AN ABORT
2967      ;COMMAND IS ISSUED TO THE UUT TO STOP EXECUTION OF THE LOCAL PROGRAM.
2968
2969      041110  005064  000014      ENDCLE: CLR    LUNFLG(R4)      ;CLEAR OUT THE LUN FLAGS
2970
2971      ;NOTE: THIS LINE OF CODE MAY HAVE TO BE REMOVED TO HANDLE ^C FOLLOWED
2972      ;BY A PROCEED COMMAND CORRECTLY.
2973      041114                      CLRVEC  TKVEC(R4)                      ;PUT "TRAP CATCHER" INTO VECTOR
           041114  016400  000004      MOV    TKVEC(R4),R0
           041120  104436      TRAP   C$CVEC
2974
2975      041122                      EXIT   CLN
           041122  104432      TRAP   C$EXIT
           041124  000002      .WORD  L10012-.
2976
2988
2989      .EVEN
2990
2991      041126      ENDCLN
           041126      L10012:
           041126  104412      TRAP   C$CLEAN
    
```

```
2993          .SBTTL  DROP UNIT SECTION
2994
2995          ;**
2996          ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
2997          ; TO NO LONGER BE TESTED.
2998          ;--
2999
3000 041130          BGNDU
      041130          L$DU::
3001
3007
3008 041130 012764 000001 000014          MOV     $DRPFLG,LUNFLG(R4)          ;LETS PROGRAM KNOW IT'S DEAD
3009
3010 041136          EXIT     DU
      041136 000167          .WORD  J$JMP
      041140 000000          .WORD  L10013-2-.
3011
3023
3024          .EVEN
3025
3026 041142          ENDDU
      041142          L10013:
      041142 104453          TRAP   C$DU
```

```
3028 .SBTTL ADD UNIT SECTION
3029
3030 ;**
3031 ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
3032 ; TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
3033 ; TO THE TEST CYCLE.
3034 ; -
3035
3036 041144          EGNAU
      041144          L$AU::
3037
3043
3044 041144          EXIT      AU
      041144 000167      .WORD  J$JMP
      011146 000000      .WORD  L10014-2 .
3045
3057
3058          .EVEN
3059
3060 041150          ENDAU
      041150          L10014:
      041150 104452      TRAP    C$AU
3061
3062 041152          ENDMOD
3063
```

```

3066
3067
3071      000000
3072
3073
3074
3075
3076
3077
3081
3082
3095
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3114 041152
      041152
3115 041152 000240
3116 041154 012737 000001 002272
3117 041162 005737 002270
3118 041166 001404
3119 041170 012737 000010 002272
3120 041176 000240
3121 041200
      041200
      041200 104402
3122 041202 005037 002276
3123
3124 041206
      041206 012746 000340
      041212 012746 036054
      041216 012746 000004
      041222 012746 000003
      041226 104437
      041230 062706 000010
3125 041234 000240
3126 041236 005074 000000
3127 041242 000240
3128 041244
      041244 012727 000001
      041250 000000
      041252 013727 002116
      041256 000000
      041260 005367 177772
      041264 001375
      041266 005367 177756

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

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

      .SBTTL TEST 1: EXISTENCE VERIFICATION TEST

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

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

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

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

```

041272 001367      BNE      .-20
3129
3130 041274 005737 002276      TST      TRP4FG      ;IF NO TRAP OCCURRED
3131 041300 001416              BEQ      5$          ; THEN CONTINUE TEST
3132 041302 000240              NOP
3133 041304 012737 032640 002310  MOV      #CTRL,FRUIS ;IDENTIFY FAILING FRU FOP PRINTOUT
3134 041312              ERRDF      5,EMSG5,PRIERR ;"NXM ON READ TKIP"
      041312 104455          TRAP      C$ERDF
      041314 000005          .WORD    5
      041316 030334          .WORD    EMSG5
      041320 033146          .WORD    PRIERR
3135 041322              CKLOOP
      041322 104406          TRAP      C$CLP1      ;LOOP ON ERROR?
3136 041324              DODU      LOGUNT      ;DROP UNIT
      041324 013700 002312  MOV      LOGUNT,R0
      041330 104451          TRAP      C$DODU
3137 041332              ESCAPE     SUB          ;CAN'T CONTINUE
      041332 104410          TRAP      C$ESCAPE
      041334 000002          .WORD    L10016 .
3138
3139 041336              5$:      ENDSUB
      041336              L10016:
      041336 104403          TRAP      C$ESUB
3140 041340 000240              NOP
3141 041342              CLRVEC    #VEC4      ;RESTORE VECTOR 4
      041342 012700 000004  MOV      #VEC4,R0
      041346 104436          TRAP      C$CVEC
3142 041350 032764 000001 000014  BIT      #DRPFLG,LUNFLG(R4) ;IF UNIT WAS NOT DROPPED
3143 041356 001402              BEQ      T1.2      ; THEN CONTINUE TESTING
3144 041360              ESCAPE     TST          ; ELSE LEAVE TEST
      041360 104410          TRAP      C$ESCAPE
      041362 000264          .WORD    L10015-.
3145
3146 041364              BGNSUB
      041364              T1.2:
      041364 104402          TRAP      C$BSUB
3147 041366 005037 002276 10$:      CLR      TRP4FG      ;CLEAR NXM ERROR FLAG
3148
3149 041372              SETVEC    #VEC4,#TRAP4,#PRI07 ;SET VECTOR 4 FOR NXM TRAPS
      041372 012746 000340  MOV      #PRI07,-(SP)
      041376 012746 036054  MOV      #TRAP4,(SP)
      041402 012746 000004  MOV      #VEC4,-(SP)
      041406 012746 000003  MOV      #3,-(SP)
      041412 104437          TRAP      C$SVEC
      041414 062706 000010  ADD      #10,SP
3150 041420 000240              NOP
3151 041422 005774 000002          TST      @TKSA(R4)      ;READ THE SA REGISTER
3152 041426 000240              NOP
3153 041430              DELAY     25.          ;WAIT TO ALLOW NXM TRAP
      041430 012727 000031  MOV      #25.,(PC)+
      041434 000000          .WORD    0
      041436 013727 002116  MOV      L$DLY,(PC)+
      041442 000000          .WORD    0
      041444 005367 177772          DEC      -6(PC)
      041450 001375          BNE      .-4
      041452 005367 177756          DEC      -22(PC)
      041456 001367          BNE      . 20
  
```



```

3154
3155 041460 005737 002276          TST      TRP4FG          ;IF NXM DID NOT OCCUR
3156 041464 001416          BEQ      15$           ; THEN CONTINUE TEST
3157 041466 000240          NOP
3158 041470 012737 032640 002310  MOV      #CTRL,FRUIS    ;IDENTIFY FAILING FRU FOR PRINTOUT
3159 041476          ERRDF    7,MSG7,PRIERR ;"NXM ON FIRST READ OF SA"
      041476 104455          TRAP    C$ERDF
      041500 000007          .WORD   7
      041502 030406          .WORD   MSG7
      041504 033146          .WORD   PRIERR
3160 041506          CKLOOP                    ;LOOP ON ERROR?
      041506 104406          TRAP    C$CLP1
3161 041510          DODU    LOGUNT          ;DROP UNIT IF NOT
      041510 013700 002312  MOV      LOGUNT,RO
      041514 104451          TRAP    C$DODU
3162 041516          ESCAPE  SUB            ;LEAVE TEST
      041516 104410          TRAP    C$ESCAPE
      041520 000062          .WORD   L10017 .
3163
3164 041522 017464 000002 000012 15$:  MOV      @TKSA(R4),TKSASV(R4) ;GET A COPY OF SA IN MEMORY
3165 041530 032764 004000 000012  BIT      #B.S1,TKSASV(R4) ;IF STEP 1 BIT IS SET
3166 041536 001021          BNE     16$           ; THEN TEST 1 IS COMPLETE
3167 041540 000240          NOP
3168 041542 012737 004000 002314  MOV      #B.S1,SAEXP    ;LOAD "EXPECTED FOR PRINTOUT
3169 041550 012737 032640 002310  MOV      #CTRL,FRUIS    ;IDENTIFY FAILING FRU FOR PRINTOUT
3170 041556          ERRDF    8.,MSG8,PRISA ;"SA REGIN ERROR ON FIRST READ"
      041556 104455          TRAP    C$ERDF
      041560 000010          .WORD   8
      041562 030427          .WORD   MSG8
      041564 032764          .WORD   PRISA
3171 041566          CKLOOP                    ;LOOP ON ERROR?
      041566 104406          TRAP    C$CLP1
3172 041570          DODU    LOGUNT          ;DROP UNIT IF NOT
      041570 013700 002312  MOV      LOGUNT,RO
      041574 104451          TRAP    C$DODU
3173 041576          ESCAPE  SUB            ;LEAVE TEST
      041576 104410          TRAP    C$ESCAPE
      041600 000002          .WORD   L10017-.
3174 041602          ENDSUB
      041602          L10017:
      041602 104403          TRAP    C$ESUB
3175
3176 041604 005037 002314          CLR     SAEXP          ;CLEAR ERROR INDICATOR
3177 041610          CLRVEC #VEC4          ;RESTORE VECTOR 4
      041610 012700 000004  MOV      #VEC4,RO
      041614 104436          TRAP    C$CVEC
3178 041616 032764 000001 000014  BIT      #DRPFLG,LUNFLG(R4) ;IF UNIT DROPPED
3179 041624 001006          BNE     25$           ; THEN LEAVE NOW
3180 041626 005337 002272          DEC     ITRCNT        ;IF ITERATIONS EQUAL 0
3181 041632 000240          NOP
3182 041634 001402          BEQ     25$           ; THEN LEAVE TEST
3183 041636 000137 041200          JMP     T1.1          ; ELSE GO BACK FOR MORE
3184
3185 041642          EXIT  TST
      041642 104432          TRAP    C$EXIT
      041644 000002          .WORD   L10015-.
3186
    
```

3187				
3188			.EVEN	
3189				
3190	041646		ENDTST	
	041646	L10015:		
	041646	104401	TRAP	C\$ETST

```

3193          .SBTTL TEST 2: SA REGISTER WRAP TEST
3197
3198          ;*****
3199          ;*****
3200          ;
3201          ;TEST 2 - SA REGISTER WRAP TEST
3202          ;   THIS TEST WILL INITIALIZE THE UUT BY WRITING TO ITS
3203          ;   IP REGISTER. IT WILL FORCE THE UUT INTO DIAGNOSTIC
3204          ;   WRAP MODE, AND WRITE FIRST A FLOATING 0 DATA PATTERN,
3205          ;   FOLLOWED BY A FLOATING 1 DATA PATTERN TO THE SA REG.
3206          ;   EACH WRITE WILL BE FOLLOWED BY A READ AND COMPARE
3207          ;   OPERATION.
3208          ;
3209          ;*****
3210          ;*****
3214 041650    BGNTST
          041650    T2::
3215
3216 041650    032764 000001 000014    BIT      #DRPFLG,LUNFLG(R4)    ;IF UUT NOT DROPPED
3217 041656    001423                    BEQ      GO2                    ; THEN DO TEST
3218 041660    104421                    RFLAGS  FLAGS                    ;ELSE GO GET SUPRVISOR FLAGS
          041662    010037 002334    TRAP    C#RFLA
3219 041666    032737 001000 002334    MOV     RO,FLAGS
3220 041674    001412                    BIT      #PNT,FLAGS            ;SEE IF WE'RE PRINTING TEST NUMBERS
3221 041676    013746 002114                    BEQ      1#                    ;NO, DON'T PRINT BYPASSED
          041702    012746 047626    PRINTF  #BYPASS,L#TEST        ; ELSE PRINT THE TEST
          041706    012746 000002    MOV     L#TEST,-(SP)
          041712    010600                    MOV     #BYPASS,-(SP)
          041714    104417                    MOV     #2,-(SP)
          041716    062706 000006    MOV     SP,RO
          041722    104432                    TRAP   C#PNTF
          041724    000504                    ADD    #6,SP
3222 041722    1#:                    EXIT   TST                        ;BYPASSED MESSAGE AND GET OUT
          041722    104432                    TRAP   C#EXIT
          041724    000504                    .WORD  L10020-.
3223
3224 041726    012737 000001 002316  G02:    MOV     #1,INISTP                ;STEP 1 FOR ERROR PRINTOUT
3225 041734    012737 000001 002272    MOV     #1,ITRCNT                ;SET UP FOR ONE TEST ITERATION
3226 041742    022737 000001 002270    CMP     #1,PASCNT                ;IF FIRST PASS
3227 041750    001403                    BEQ     2#                        ; THEN START TEST
3228 041752    012737 000002 002272    MOV     #2,ITRCNT                ; ELSE DO 2 ITERATIONS
3229
3230 041760    012737 140000 002314  2#:    MOV     #BIT15!B.WR,SAEXP        ;SET UP STEP 1 FOR DIAG. WRAP MODE
3231 041766    013737 002314 002250    MOV     SAEXP,STPTBL            ;PUT IT IN STEP 1 OF TABLE
3232 041774    004737 036434                    JSR    PC,STEP1                ;GO DO IT
3233
3234 042000    005737 002320                    TST    STEPST                    ;IF STATUS OKAY
3235 042004    001415                    BEQ     5#                        ; THEN CONTINUE TEST
3236
3237 042006    012737 032640 002310    MOV     #CTRL,FRUIS              ;FAILING FRU FOR PRINTOUT
3238 042014    104455                    ERDF   9.,EMSG9,PRIINI          ;"SA CONTENTS IN ERROR"
          042014    104455                    TRAP   C#ERDF
          042016    000011                    .WORD  9
          042020    030465                    .WORD  EMSG9
          042022    032740                    .WORD  PRIINI
3239 042024    CKLOOP
          042024    104406                    TRAP   C#CLP1                    ;LOOP ON ERROR?
    
```

```

3240 042026          DODU      LOGUNT          LOGUNT
      042026 013700 002312      MOV      LOGUNT,RO
      042032 104451          TRAP      C$DODU
3241 042034          ESCAPE     TST              ;LEAVE TST
      042034 104410          TRAP      C$ESCAPE
      042036 000372          .WORD     L10020-.
3242
3243 042040 012737 000100 002326 5$:  MOV      #100,OUTER      ;SET UP FOR DELAY ROUTINE
3244 042046 012737 037200 002324 6$:  MOV      #16000.,INNER ;SET UP INNER
3245 042054 017464 000002 000012      MOV      @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
3246 042062 023764 002314 000012      CMP      SAEXP,TKSASV(R4) ;IF SA IS WHAT WE EXPECT
3247 042070 001422          BEQ      10$          ; THEN MOVE ALONG
3248 042072 004737 036406          JSR      PC,PDELAY    ; ELSE GIVE UUT SOME TIME
3249 042076 005737 002330          TST      TOUT        ;IF NO TIMEOUT YET
3250 042102 001761          BEQ      6$          ; THEN GO TAKE ANOTHER LOOK
3251
3252 042104 012737 032640 002310      MOV      #CTRL,FRUIS ;FAILING FRU FOR PRINTOUT
3253 042112          ERRDF     10.,EMSG9,PRIINI ;"SA CONTENTS IN ERROR"
      042112 104455          TRAP      C$ERDF
      042114 000012          .WORD     10
      042116 030465          .WORD     EMSG9
      042120 032740          .WORD     PRIINI
3254 042122          CKLOOP
      042122 104406          TRAP      C$CLP1
3255 042124          DODU      LOGUNT
      042124 013700 002312      MOV      LOGUNT,RO
      042130 104451          TRAP      C$DODU
3256 042132          ESCAPE     TST
      042132 104410          TRAP      C$ESCAPE
      042134 000274          .WORD     L10020 .
3257
3258 042136 012737 177776 002322 10$: MOV      #177776,WRDATA ;INITIALIZE WRAP DATA
3259 042144 013774 002322 000002 11$: MOV      WRDATA,@TKSA(R4) ;SEND DATA TO UUT
3260 042152 013737 002322 002314      MOV      WRDATA,SAEXP ;SAVE A COPY FOR COMPARE
3261 042160 012737 000100 002326      MOV      #100,OUTER ;SET UP FOR DELAY ROUTINE
3262
3263 042166 013737 037200 002324 15$: MOV      16000.,INNER ;INNER TOO
3264 042174 017464 000002 000012      MOV      @TKSA(R4),TKSASV(R4) ;READ SA
3265 042202 023764 002314 000012      CMP      SAEXP,TKSASV(R4) ;IF DATA MATCHES
3266 042210 001422          BEQ      20$          ; THEN CHANGE DATA
3267 042212 004737 036406          JSR      PC,PDELAY    ; ELSE GIVE UUT SOME TIME
3268 042216 005737 002330          TST      TOUT        ;IF NO TIMEOUT YET
3269 042222 001761          BEQ      15$          ; THEN GO TAKE ANOTHER LOOK
3270
3271 042224 012737 032640 002310      MOV      #CTRL,FRUIS ;FAILING FRU FOR PRINTOUT
3272 042232          ERRDF     11.,EMSG10,PRIINI ;"SA WRONG IN DATA WRAP"
      042232 104455          TRAP      C$ERDF
      042234 000013          .WORD     11
      042236 030512          .WORD     EMSG10
      042240 032740          .WORD     PRIINI
3273 042242          CKLOOP
      042242 104406          TRAP      C$CLP1
3274 042244          DODU      LOGUNT
      042244 013700 002312      MOV      LOGUNT,RO
      042250 104451          TRAP      C$DODU
3275 042252          ESCAPE     TST              ;GET OUT IF NOT LOOPING
      042252 104410          TRAP      C$ESCAPE
  
```

```

042254 000154 .WORD L10020 .
3276
3277 042256 006137 002322 20$: ROL WRDATA ;SHIFT TEST PATTERN
3278 042262 103730 BCS 11$ ;WE RE NOT DONE YET
3279
3280 042264 012737 000001 002322 MOV #1,WRDATA ;SET UP FOR FLOATING 1 PATTERN
3281 042272 013774 002322 000002 24$: MOV WRDATA,@TKSA(R4) ;SEND DATA TO UUT
3282 042300 013737 002322 002314 MOV WRDATA,SAEXP ;KEEP A COPY FOR COMPARE
3283 042306 012737 000100 002326 MOV #100,OUTER ;SET UP FOR DELAY ROUTINE
3284
3285 042314 012737 016000 002324 25$: MOV #16000,INNER ;DELAY ROUTINE TOO
3286 042322 017464 000002 000012 MOV @TKSA(R4),TKSASV(R4) ;READ THE SA
3287 042330 023764 002314 000012 CMP SAEXP,TKSASV(R4) ;IF IT MATCHES
3288 042336 001422 BEQ 30$ ; THEN SEE IF WE'RE DONE
3289 042340 004737 C36406 JSR PC,PDELAY ; ELSE GIVE UUT SOME MORE TIME
3290 042344 005737 002330 TST TOUT ;IF NO TIMEOUT YET
3291 042350 001761 BEQ 25$ ; THEN TAKE ANOTHER LOOK
3292
3293 042352 012737 032640 002310 MOV #CTRL,FRUIS ;FAILING FRU FOR PRINTOUT
3294 042360 ERRDF 12.,EMSG10,PRIINI ;"SA WRONG IN DATA WRAP"
042360 104455 TRAP C$ERDF
042362 000014 .WORD 12
042364 030512 .WORD EMSG10
042366 032740 .WORD PRIINI
3295 042370 CKLOOP
042370 104406 TRAP C$CLP1
3296 042372 DODU LOGUNT
042372 013700 002312 MOV LOGUNT,RO
042376 104451 TRAP C$DODU
3297 042400 ESCAPE TST ;LEAVE TEST IF NOT LOOPING
042400 104410 TRAP C$ESCAPE
042402 000026 .WORD L10020-.
3298
3299 042404 006137 002322 30$: ROL WRDATA ;SHIFT DATA PATTERN
3300 042410 103330 LCC 24$ ;WE'RE NOT DONE YET
3301 042412 005337 002272 DEC ITRCNT ;IF ITERATIONS = 0
3302 042416 001402 BEQ T2EXT ; THEN LEAVE TEST
3303 042420 000137 041760 JMP 2$ ; ELSE DO ANOTHER ONE
3304
3305 042424 T2EXT: EXIT TST ;GET OUTTA HERE
042424 104432 TRAP C$EXIT
042426 000002 .WORD L10020-.
3306
3307 042430 ENDTST
042430 L10020: TRAP C$ETST
042430 104401
    
```

```

3310 .SBTTL TEST 3: INITIALIZATION TEST
3314
3315
3316 ;*****
3317 ;*****
3318 ;
3319 ;TEST 3 - INITIALIZATION TEST
3320 ; THIS TEST COMMENCES THE UQ-PORT INITIALIZATION SEQUENCE
3321 ; WITH INTERRUPTS DISABLED. IT VERIFIES THAT ALL STEP
3322 ; TRANSITIONS OCCUR WITHIN THE ALLOTTED TIME, AND THAT ALL
3323 ; MOST SUPPLIED INFORMATION IS ECHOED BY THE UUT. THE
3324 ; PROGRAM FURTHER VERIFIES THAT NO INTERRUPTS OCCUR AS A
3325 ; RESULT OF THE STEP TRANSITIONS.
3326 ;
3327 ;*****
3328 ;*****
3332
3333 042432 BGNTST
3333 042432 T3::
3334
3335 042432 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;IF UUT NOT DROPPED
3336 042440 001423 BEQ G03 ; THEN DO TEST
3337 042442 RFLAGS FLAGS ;ELSE GO GET SUPRVISOR FLAGS
3337 042442 104421 TRAP C#RFLA
3338 042450 032737 001000 002334 MOV R0,FLAGS
3339 042456 001412 BIT #PNT,FLAGS ;SEE IF WE'RE PRINTING TEST NUMBERS
3340 042460 PRINTF #BYPASS,L#TEST ;NO, DON'T PRINT BYPASSED
3340 042460 013746 002114 MOV L#TEST,-(SP) ; ELS PRINT THE TEST
3340 042464 012746 047626 MOV #BYPASS,-(SP)
3340 042470 012746 000002 MOV #2,-(SP)
3340 042474 010600 MOV SP,R0
3340 042476 104417 TRAP C#PNTF
3341 042500 062706 000006 ADD #6,SP
3341 042504 1# EXIT TST ;BYPASSED MESSAGE AND GET OUT
3341 042504 104432 TRAP C#EXIT
3341 042506 000410 .WORD L10021-.
3342
3343 042510 012737 032640 002310 G03: MOV #CTRL,FRUIS ;FAILING FRU IN CASE OF ERROR
3344 042516 012737 000001 002272 MOV #1,ITRCNT ;SET UP FOR ONE TEST ITERATION
3345 042524 022737 000001 002270 CMP #1,PASCNT ;IF FIRST PASS
3346 042532 001403 BEQ 2# ; THEN START TEST
3347 042534 012737 000012 002272 MOV #10.,ITRCNT ; ELSE DO 10 ITERATIONS
3348
3349 042542 012705 000000 2#: MOV #0,R5 ;SET UP R5 AS INDEX TO STEP TABLES
3350 042546 012737 000001 002316 MOV #1,INISTP ;STEP 1 FOR ERROR PRINTOUT
3351 042554 016437 000004 002250 MOV TKVEC(R4),STPTBL ;PUT VECTOR IN STEP 1
3352 042562 006237 002250 ASR STPTBL ;DIVIDE BY TWO
3353 042566 006237 002250 ASR STPTBL ;DIVIDE BY FOUR
3354 042572 013737 002250 002264 MOV STPTBL,CMPTBL*4 ;PUT VECTOR IN STEP 3 COMPARE
3355 042600 052737 104400 002250 BIS #104400,STPTBL ;REST OF STEP ONE
3356 042606 READBUS ;IF QBUS
3356 042606 104407 TRAP C#RDBU
3357 042610 BCOMPLETE 3# ; THEN BR FOR QBUS SET UP
3357 042610 103404 BCS 3#
3358 042612 012737 004700 002260 MOV #4700,CMPTBL ; ELSE SET UP FOR UNIBUS
3359 042620 000403 BR 4#
    
```

```

3360 042622 012737 005700 002260 3$:  MOV    #B.S1!B.QB!B.DI!B.OO!B.MP,CMPTBL
3361                                     ;STEP 1 COMPARE VALUE
3362 042630 012737 060050 002252 4$:  MOV    #COMMAR,STPTBL+2      ;STEP 2 - COMM AREA ADDRESS
3363 042636 012737 010211 002262      MOV    #010211,CMPTBL+2    ;STEP 2 COMPARE
3364 042644 012737 000000 002254      MOV    #0,STPTBL+4        ;STEP 3 - HIGH ADDRESS
3365 042652 112737 000040 002265      MOV    #40,CMPTBL+5      ;REST OF STEP 3 COMPARE
3366 042660 012737 000000 002256      MOV    #0,STPTBL+6        ;STEP 4
3367 042666 012737 040000 002266      MOV    #040000,CMPTBL+6  ;STEP 4 COMPARE
3368
3369 042674 004737 036434      JSR    PC,STEP1          -GO DO IT
3370 042700 005737 002320      TST    STEPST           ;IF STATUS OKAY
3371 042704 001412      BEQ    5$              ; THEN CONTINUE TEST
3372
3373 042706      ERRDF  9.,EMSG9,PRIINI   ;"SA CONTENTS IN ERROR"
      TRAP  C$ERDF
      .WORD 9
      .WORD EMSG9
      .WORD PRIINI
3374 042716      CKLOOP
      TRAP  C$CLP1        ;LOOP ON ERROR?
      DODU  LOGUNT        ;DROP UUT
3375 042720      MOV    LOGUNT,RO
      TRAP  C$DODU
3376 042726      ESCAPE TST           ;LEAVE TST
      TRAP  C$ESCAPE
      .WORD L10021-.
3377
3378 042732 0C5237 002316      5$:  INC    INISTP           ;ADJUST STEP COUNTcR
3379 042736 062705 000002      ADD    #2,,5           ;ADJUST TABLE INDEX
3380 042742 012737 000100 002326 6$:  MOV    #100,OUTER      ;SET UP FOR DELAY ROUTINE
3381 042750 016537 002260 002314      MOV    CMPTBL(R5),SAEXP ;SET UP FOR COMPARE
3382 042756 012737 037200 002324 7$:  MOV    #16000.,INNER   ;SET UP INNER
3383 042764 017464 000002 000012      MOV    #TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
3384 042772 022705 000006      CMP    #6,R5           ;ARE WE IN STEP 4?
3385 042776 001005      BNE    8$              ;BRANCH IF NOT
3386 043000 033764 002314 000012      BIT    SAEXP,TKSASV(R4) ;JUST LOOK FOR STEP 4 BIT
3387 043006 001024      BNE    10$            ;IT'S SET SO LET'S GO
3388 043010 000404      BR     9$              ;STAY IN LOOP OTHERWISE
3389 043012 023764 002314 000012 8$:  CMP    SAEXP,TKSASV(R4) ;IF SA IS WHAT WE EXPECT
3390 043020 001417      BEQ    10$            ; THEN MOVE ALONG
3391 043022 004737 036406      9$:  JSR    PC,PDELAY       ; ELSE GIVE UUT SOME TIME
3392 043026 005737 002330      TST    TOUT           ;IF NO TIMEOUT YET
3393 043032 001751      BEQ    7$              ; THEN GO TAKE ANOTHER LOOK
3394
3395 043034      ERRDF  13.,EMSG9,PRIINI ;"SA CONTENTS IN ERROR"
      TRAP  C$ERDF
      .WORD 13
      .WORD EMSG9
      .WORD PRIINI
3396 043044      CKLOOP
      TRAP  C$CLP1
      DODU  LOGUNT
3397 043046      MOV    LOGUNT,RO
      TRAP  C$DODU
3398 043054      ESCAPE TST
      TRAP  C$ESCAPE
      .WORD L10021-.
    
```

```
3399
3400 043060 016574 002250 000002 10$:  MOV  STPTBL(R5),@TKSA(R4) ;WRITE NEXT STEP TO UUT
3401 043066 022705 000006          CMP  #6,R5 ;IF NOT IN STEP 4
3402 043072 001317          BNE  5$ ;GO BACK TO MAIN LOOP
3403
3404 043074 032764 000001 000014          BIT  @DRPFLG,LUNFLG(R4) ;HAS UUT BEEN DROPPED
3405 043102 001003          BNE  T3EXT ;LEAVE NOW IF SO
3406 043104 005337 002272          DEC  ITRCNT ;IF MORE ITERATIONS LEFT
3407 043110 001214          BNE  2$ ; THEN GO DO IT AGAIN
3408
3409          T3EXT:  EXIT  TST
          043112 104432          TRAP C$EXIT
          043114 000002          .WORD L10021 .
3410
3411          ENDTST
          043116          L10021:
          043116 104401          TRAP  C$ETST
```



3414  
 3418  
 3419  
 3420  
 3421  
 3422  
 3423  
 3424  
 3425  
 3426  
 3427  
 3428  
 3432  
 3433

.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.
;
;*****
;*****
    
```

3434 043120  
 043120

BGNTST  
 T4::

3435

3436 043120 032764 000001 000014

3437 043126 001423

3438 043130

043130 104421

043132 010037 002334

3439 043136 032737 001000 002334

3440 043144 001412

3441 043146

043146 013746 002114

043152 012746 047626

043156 012746 000002

043162 010600

043164 104417

043166 062706 000006

3442 043172

043172 104432

043174 000516

3443

3444 043176 042764 000004 000014

3445 043204 042764 000002 000014

3446 043212 012737 032640 002310

3447 043220 012737 000001 002272

3448 043226 022737 000001 002270

3449 043234 001403

3450 043236 012737 000012 002272

3451

3452 043244 004737 036316

3453 043250 012705 000000

3454 043254 012737 000001 002316

3455 043262 016437 000004 002250

3456 043270 006237 002250

3457 043274 006237 002250

3458 043300 013737 002250 002264

3459 043306 052737 104600 002250

3460 043314

043314 104407

3461 043316

043316 103404

3462 043320 012737 004700 002260

3463 043326 000403

```

;*****
;*****
;
;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.
;
;*****
;*****
    BGNTST
    T4::
        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
        1$:   EXIT    TST                ;BYPASSED MESSAGE AND GET OUT
        TRAP   C#EXIT
        .WORD  L10022-.

        G04:  BIC    #BRFLAG,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$:   JSR    PC,VECTOR           ;SET UP VECTOR WITH INTERRUPT HANDLER
        MOV    #0,R5                    ;SET UP R5 AS INDEX TO STEP TABLES
        MOV    #1,INISTP                ;STEP 1 FOR ERROR PRINTOUT
        MOV    TKVEC(R4),STPTBL         ;PUT VECTOR IN STEP 1
        ASR    STPTBL                    ;DIVIDE BY TWO
        ASR    STPTBL                    ;DIVIDE BY FOUR
        MOV    STPTBL,CMPTBL+4          ;PUT VECTOR IN STEP 3 COMPARE
        BIS    #104600,STPTBL          ;REST OF STEP ONE
        READBUS
        TRAP   C#RDBU                    ;IF QBUS
        BCOMPLETE    3$                 ; THEN BR FOR QBUS SET UP
        BCS    3$
        MOV    #4700,CMPTBL            ; ELSE SET UP FOR UNIBUS
        BR    4$
    
```

```

3464 043330 012737 005700 002260 3$:  MOV    #8.S1!B.Q8!B.DI!B.OD!B.MP,CMPTBL
3465                                     ;STEP 1 COMPARE VALUE
3466 043336 012737 060050 002252 4$:  MOV    @COMMAR,STPTBL+2      ;STEP 2 - COMM AREA ADDRESS
3467 043344 012737 010211 002262      MOV    @010211,CMPTBL+2    ;STEP 2 COMPARE
3468 043352 012737 000000 002254      MOV    @0,STPTBL+4        ;STEP 3 HIGH ADDRESS
3469 043360 052737 000200 002264      BIS    @8.IE,CMPTBL+4     ;SET THE INTERRUPT ENABLE BIT
3470 043366 112737 000040 002265      MOV    @40,CMPTBL+5      ;REST OF STEP 3 COMPARE
3471 043374 012737 000000 002256      MOV    @0,STPTBL+6      ;STEP 4
3472 043402 012737 040000 002266      MOV    @040000,CMPTBL+6  ;STEP 4 COMPARE
3473
3474 043410 004737 036434      JSR    PC,STEP1          ;GO DO IT
3475 043414 005737 002320      TST    STEPST           ;IF STATUS OKAY
3476 043420 001412      BEQ    5$               ; THEN CONTINUE TEST
3477
3478 043422      ERRDF  14.,MSG9,PRIINI    ;"SA CONTENTS IN ERROR"
      043422 104455      TRAP  C$ERDF
      043424 000016      .WORD 14
      043426 030465      .WORD MSG9
      043430 032740      .WORD PRIINI
3479 043432      CKLOOP
      043432 104406      TRAP  C$CLF1           ;LOOP ON ERROR?
3480 043434      DODU  LOGUNT           ;DROP UUT
      043434 013700 002312      MOV    LOGUNT,R0
      043440 104451      TRAP  C$DODU
3481 043442      ESCAPE TST             ;LEAVE TST
      043442 104410      TRAP  C$ESCAPE
      043444 000246      .WORD L10022-.
3482
3483 043446 012737 000100 002326 5$:  MOV    #100,OUTER        ;SET UP FOR DELAY ROUTINE
3484 043454 016537 002260 002314      MOV    CMPTBL(R5),SAEXP  ;SET UP FOR COMPARE
3485 043462 012737 037200 002324 7$:  MOV    #16000.,INNER    ;SET UP INNER
3486 043470 032764 000002 000014      BIT    #INTFLG,LUNFLG(R4) ;IF INTERRUPT OCCURRED
3487 043476 001017      BNE   10$              ; THEN SEE IF SA IS CORRECT
3488 043500 004737 036406      JSR    PC,PDELAY        ; ELSE GIVE UUT SOME TIME
3489 043504 005737 002330      TST    TOUT            ;IF NO TIMEOUT YET
3490 043510 001764      BEQ    7$              ; THEN GO TAKE ANOTHER LOOK
3491
3492 043512      ERRDF  15.,MSG11,PRIERR  ;"EXPECTED INTERRUPT DID NOT OCCUR"
      043512 104455      TRAP  C$ERDF
      043514 000017      .WORD 15
      043516 030540      .WORD MSG11
      043520 033146      .WORD PRIERR
3493 043522      CKLOOP
      043522 104406      TRAP  C$CLF1
3494 043524      DODU  LOGUNT
      043524 013700 002312      MOV    LOGUNT,R0
      043530 104451      TRAP  C$DODU
3495 043532      ESCAPE TST
      043532 104410      TRAP  C$ESCAPE
      043534 000156      .WORD L10022-.
3496
3497 043536 042764 000002 000014 10$: BIC    #INTFLG,LUNFLG(R4) ;CLEAR THE INTERRUPT FLAG
3498 043544 005237 002316      INC    INISTP          ;ADJUST THE STEP COUNTER
3499 043550 062705 000002      ADD    #2,R5           ;ADJUST TABLE INDEX
3500 043554 016537 002260 002314      MOV    CMPTBL(R5),SAEXP  ;GET THE COMPARISON VALUE
3501 043562 017464 000002 000012      MOV    @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
3502 043570 022705 000006      CMP    #6,R5           ;ARE WE IN STEP 4?
    
```

3503	043574	001005			BNE	15\$					
3504	043576	033764	002314	000012	BIT	SAEXP,TKSASV(R4)					
3505	043604	001017			BNE	20\$					
3506	043606	000404			BR	16\$					
3507	043610	023764	002314	000012	15\$:	CMP	SAEXP,TKSASV(R4)				
3508	043616	001412			BEQ	20\$					
3509											
3510	043620				10\$:	ERRDF	16.,EMSG9,PRIINI				
	043620	104455			TRAP	C\$ERDF					
	043622	000020			.WORD	16					
	043624	030465			.WORD	EMSG9					
	043626	032740			.WORD	PRIINI					
3511	043630				CKLOOP						
	043630	104406			TRAP	C\$CLP1					
3512	043632				DODU	LOGUNT					
	043632	013700	002312		MOV	LOGUNT,R0					
	043636	104451			TRAP	C\$DODU					
3513	043640				ESCAPE	TST					
	043640	104410			TRAP	C\$ESCAPE					
	043642	000050			.WORD	L10022-					
3514											
3515	043644	016574	002250	000002	20\$:	MOV	STPTBL(R5),@TKSA(R4)				
3516	043652	022705	000006		CMP	#6,R5					
3517	043656	001273			BNE	5\$					
3518											
3519	043660	032764	000001	000014	BIT	#DRPFLG,LUNFLG(R4)					
3520	043666	001005			BNE	T4EXT					
3521	043670	005337	002272		DEC	ITRCNT					
3522	043674	001402			BEQ	T4EXT					
3523	043676	000137	043244		JMP	2\$					
3524											
3525	043702	004737	036266		T4EXT:	JSR	PC,RSTVEC				
3526	043706				EXIT	TST					
	043706	104432			TRAP	C\$EXIT					
	043710	000002			.WORD	L10022					
3527											
3528	043712				ENDTST						
	043712				L10022:						
	043712	104401			TRAP	C\$ETST					

;BRANCH IF NOT  
 ;JUST LOOK FOR STEP 4 BIT  
 ;IT'S SET SO LET'S GO  
 ;ERROR  
 ;IF SA IS WHAT WE EXPECT  
 ; THEN MOVE ALONG

;"SA CONTENTS IN ERROR'

;WRITE NEXT STEP TO UUT  
 ;IF NOT IN STEP 4  
 ;GO BACK TO MAIN LOOP  
 ;HAS UUT BEEN DROPPED  
 ;LEAVE NOW IF SO  
 ;IF NO MORE ITERATIONS LEFT  
 ; THEN EXIT  
 ; ELSE DO IT AGAIN

;CATCH ILLEGAL INTERRUPTS

```

3531 .SBITL TEST 5: BR LEVEL TEST
3535
3536 ;*****
3537 ;*****
3538 ;
3539 ;TEST 5 - BR LEVEL TEST
3540 ; THIS TEST INSURES THAT THE TK50 CAN NOT INTERRUPT
3541 ; WHEN THE CPU PRIORITY IS SET TO 7. THE TEST GOES
3542 ; ONLY THROUGH THE FIRST STEP OF THE INIT SEQUENCE
3543 ; SINCE THE CONTROLLER WILL "HANG" WAITING FOR THE
3544 ; INTERRUPT ACKNOWLEDGE.
3545 ;
3546 ;*****
3547 ;*****
3551
3552 043714 BGNTST
      043714 TS::
3553
3554 043714 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;IF OUT NOT DROPPED
3555 043722 001423 BEQ G05 ; THEN DO TEST
3556 043724 RFLAGS FLAGS ;ELSE GO GET SUPRVISOR FLAGS
      043724 104421 TRAP C$RFLA
      043726 010037 002334 MOV R0,FLAGS
3557 043732 032737 001000 002334 BIT #PNT,FLAGS ;SEE IF WE'RE PRINTING TEST NUMBERS
3558 043740 001412 BEQ 1$ ;NO, DON'T PRINT BYPASSED
3559 043742 PRINTF #BYPASS,L$TEST ; ELSE PRINT THE TEST
      043712 013746 002114 MOV L$TEST,(SP)
      043746 012746 047626 MOV #BYPASS,-(SP)
      043752 012746 000002 MOV #2,-(SP)
      043756 010600 MOV SP,R0
      043760 104417 TRAP C$PNTF
      043762 062706 000006 ADD #6,SP
3560 043766 1$: EXIT TST ;BYPASSED MESSAGE AND GET OUT
      043766 104432 TRAP C$EXIT
      043770 000370 .WORD L10023-.
3561
3562 043772 052764 000004 000014 G05: BIS #BRFLAG,LUNFLG(R4) ;DO TEST WITH HIGH PRIORITY
3563 044000 012737 032640 002310 MOV #CTRL,FRUIS ;FAILING FRU IN CASE OF ERROR
3564 044006 012737 000001 002272 MOV #1,ITRCNT ;SET UP FOR ONE TEST ITERATION
3565 044014 022737 000001 002270 CMP #1,PASCNT ;IF FIRST PASS
3566 044022 001403 BEQ 2$ ; THEN START TEST
3567 044024 012737 000002 002272 MOV #2,ITRCNT ; ELSE DO 10 ITERATIONS
3568
3569 044032 2$: SETPRI #PRI07 ;CPU PRIORITY = 7
      044032 012700 000340 MOV #PRI07,R0
      044036 104441 TRAP C$SPRI
3570 044040 004737 036316 JSR PC,VECTOR ;SET UP VECTOR WITH INTERRUPT HANDLER
3571 044044 012705 000000 MOV #0,R5 ;SET UP R5 AS INDEX TO STEP TABLES
3572 044050 012737 000001 002316 MOV #1,INISTP ;STEP 1 FOR ERROR PRINTOUT
3573 044056 016437 000004 002250 MOV TKVEC(R4),STPTBL ;PUT VECTOR IN STEP 1
3574 044064 006237 002250 ASR STPTBL ;DIVIDE BY TWO
3575 044070 006237 002250 ASR STPTBL ;DIVIDE BY FOUR
3576 044074 052737 104600 002250 BIS #104600,STPTBL ;REST OF STEP ONE
3577 044102 016437 000004 002260 MOV TKVEC(R4),CMPTBL ;STEP 1 COMPARE VALUE
3578
3579 044110 004737 036434 JSR PC,STEP1 ;GO DO IT
3580 044114 005737 002320 TST STEPST ;IF STATUS OKAY
    
```

```

3581 044120 001412          BEQ      5$          ; THEN CONTINUE TEST
3582
3583 044122          ERRDF    14.,EMSG9,PRIINI ; "SA CONTENTS IN ERROR"
      044122 104455      TRAP    C$ERDF
      044124 000016      .WORD    14
      044126 030465      .WORD    EMSG9
      044130 032740      .WORD    PRIINI
3584 044132          CKLOOP           ; LOOP ON ERROR?
      044132 104406      TRAP    C$CLP1
3585 044134          DODU     LOGUNT          ; DROP UUT
      044134 013700 002312  MOV     LOGUNT,RO
      044140 104451      TRAP    C$DODU
3586 044142          ESCAPE   TST           ; LEAVE TST
      044142 104410      TRAP    C$ESCAPE
      044144 000214      .WORD    L10023-.
3587
3588 044146 012737 000100 002326 5$:  MOV     #100,OUTER          ; SET UP FOR DELAY ROUTINE
3589 044154 016537 002260 002314  MOV     CMPTBL(R5),SAEXP      ; SET UP FOR COMPARE
3590 044162 012737 037200 002324 7$:  MOV     #16000.,INNER       ; SET UP INNER
3591 044170 004737 036406 9$:  JSR     PC,PDELAY          ; ELSE GIVE UUT SOME TIME
3592 044174 005737 002330      TST     TOUT              ; IF NO TIMEOUT YET
3593 044200 001770      BEQ     7$                ; THEN GO TAKE ANOTHER LOOK
3594
3595 044202 017464 000002 000012  MOV     @TKSA(R4),TKSASV(R4) ; GET SA CONTENTS
3596 044210 023764 002314 000012  CMP     SAEXP,TKSASV(R4)    ; IF CONTENTS OKAY
3597 044216 001412      BEQ     10$              ; THEN CHECK FOR INTERRUPT
3598
3599 044220          ERRDF    17.,EMSG9,PRIINI ; "SA CONTENTS IN ERROR"
      044220 104455      TRAP    C$ERDF
      044222 000021      .WORD    17
      044224 030465      .WORD    EMSG9
      044226 032740      .WORD    PRIINI
3600 044230          CKLOOP           ; LOOP ON ERROR?
      044230 104406      TRAP    C$CLP1
3601 044232          DODU     LOGUNT          ; DROP UUT
      044232 013700 002312  MOV     LOGUNT,RO
      044236 104451      TRAP    C$DODU
3602 044240          ESCAPE   TST           ; LEAVE TST
      044240 104410      TRAP    C$ESCAPE
      044242 000116      .WORD    L10023-.
3603
3604 044244 032764 000002 000014 10$: BIT     #INTFLG,LUNFLG(R4)    ; IF NO INTERRUPT OCCURRED
3605 044252 001415      BEQ     20$              ; THEN CARRY ON WITH TEST
3606 044254 042764 000002 000014  BIC     #INTFLG,LUNFLG(R4)  ; CLEAR FLAG IN CASE WE'RE LOOPING
3607 044262          ERRDF    18.,EMSG12,PRIINI ; "INTRRPT WITH CPU PRIORITY =7"
      044262 104455      TRAP    C$ERDF
      044264 000022      .WORD    18
      044266 030601      .WORD    EMSG12
      044270 032740      .WORD    PRIINI
3608 044272          CKLOOP           ; LOOP ON ERROR?
      044272 104406      TRAP    C$CLP1
3609 044274          DODU     LOGUNT          ; DROP UUT
      044274 013700 002312  MOV     LOGUNT,RO
      044300 104451      TRAP    C$DODU
3610 044302          ESCAPE   TST           ; LEAVE TST
      044302 104410      TRAP    C$ESCAPE
      044304 000054      .WORD    L10023-.
    
```

```
3611
3612 044306          20$:  SETPRI  @PRI00          ;CPU PRIORITY = 0
      044306 012700 000000      MOV    @PRI00,R0
      044312 104441          TRAP   C$SPRI
3613 044314 000240          NOP
3614 044316 000240          NOP
3615 044320 042764 000002 000014 BIC    @INTFLG,LUNFLG(R4) ;DELAY FOR PENDING INTERRUPT
3616                                     ;CLEAR THE FLAG NOW
3617 044326 032764 000001 000014 BIT    @DRPFLG,LUNFLG(R4) ;HAS UUT BEEN DROPPED
3618 044334 001005          BNE    T$EXT          ;LEAVE NOW IF SO
3619 044336 005337 002272          DEC    ITRCNT          ;IF NO MORE ITERATIONS LEFT
3620 044342 001402          BEQ    T$EXT          ; THEN EXIT
3621 044344 000137 044032          JMP    2$             ; ELSE DO IT AGAIN
3622
3623 044350 004737 036266          T$EXT: JSR    PC,RSTVEC ;CATCH ILLEGAL INTERRUPTS
3624 044354          EXIT    TST
      044354 104432          TRAP   C$EXIT
      044356 000002          .WORD  L10023-.
3625
3626 044360          ENDTST
      044360          L10023: TPA^  C$ETST
      044360 104401
```

```

3629 .SBTTL TEST 6: PURGE AND POLL TEST
3633 ;*****
3634 ;*****
3635 ;
3636 ;
3637 ;TEST 6 - PURGE AND POLL TEST
3638 ; THIS TEST WILL AGAIN RUN THROUGH THE INIT SEQUENCE, THIS
3639 ; TIME SETTING THE "PURGE AND POLL" BIT IN STEP 3. THIS
3640 ; SHOULD CAUSE THE PORT TO DMA VARIOUS DATA PATTERNS TO
3641 ; AND FROM THE COMMUNICATIONS AREA AND FINALLY LEAVE IT
3642 ; CLEARED BEFORE TRANSITIONING TO STEP 4. THE PROGRAM WILL
3643 ; HAVE FILLED THIS AREA WITH A BACKGROUND PATTERN OF ALL
3644 ; 1'S DATA PRIOR TO STARTING THE INIT. WHEN STEP 4 IS
3645 ; REACHED, THE PROGRAM WILL VERIFY THAT THE COMM AREA IS
3646 ; ALL 0'S, AND THAT THE 20 WORDS PRECEDING AND SUCCEEDING
3647 ; THE COMM AREA ARE UNTOUCHED. RING DEPTH USED IN THIS
3648 ; TEST IS THE MINIMUM.
3649 ;
3650 ;*****
3651 ;*****
3655
3656 044362 BGNTST
      044362 T6::
3657
3658 044362 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;IF OUT NOT DROPPED
3659 044370 001423 BEQ G06 ; THEN DO TEST
3660 044372 RFLAGS FLAGS ;ELSE GO GET SUPRVISOR FLAGS
      044372 104421 TRAP C$RFLA
      044374 010037 002334 MOV R0,FLAGS
3661 044400 032737 001000 002334 BIT #PNT,FLAGS ;SEE IF WE'RE PRINTING TEST NUMBERS
3662 044406 001412 BEQ 1$ ;NO, DON'T PRINT BYPASSED
3663 044410 PRINTF #BYPASS,L$TEST ; ELSE PRINT THE TEST
      044410 013746 002114 MOV L$TEST,-(SP)
      044414 012746 047626 MOV #BYPASS,-(SP)
      044420 012746 000002 MOV #2,-(SP)
      044424 010600 MOV SP,R0
      044426 104417 TRAP C$PNTF
      044430 062706 000006 ADD #6,SP
3664 044434 1$: EXIT TST ;BYPASSED MESSAGE AND GET OUT
      044434 104432 TRAP C$EXIT
      044436 000542 .WORD L10024-.
3665
3666 044440 012737 032640 002310 G06: MOV #CTRL,FRUIS ;FAILING FRU IN CASE OF ERROR
3667 044446 012737 000001 002272 MOV #1,ITRCNT ;SET UP FOR ONE TEST ITERATION
3668 044454 022737 000001 002270 CMP #1,PASCNT ;IF FIRST PASS
3669 044462 001403 BEQ 2$ ; THEN START TEST
3670 044464 012737 000012 002272 MOV #10.,ITRCNT ; ELSE DO 10 ITERATIONS
3671
3672 044472 012705 000000 2$: MOV #0,R5 ;SET UP R5 AS INDEX TO STEP TABLES
3673 044476 012737 000001 002316 MOV #1,INISTP ;STEP 1 FOR ERROR PRINTOUT
3674 044504 016437 000004 002250 MOV TKVEC(R4),STPTBL ;PUT VECTOR IN STEP 1
3675 044512 006237 002250 ASR STPTBL ;DIVIDE BY TWO
3676 044516 006237 002250 ASR STPTBL ;DIVIDE BY FOUR
3677 044522 013737 002250 002264 MOV STPTBL,CMPTBL+4 ;PUT VECTOR IN STEP 3 COMPARE
3678 044530 052737 104400 002250 BIS #104400,STPTBL ;REST OF STEP ONE
3679 044536 READBUS ;IF QBUS
      044536 104407 TRAP C$RDBU
    
```

```

3680 044540          BCOMPLETE      3$          : THEN BR FOR QBUS 1 UP
      044540 103404          BCS      3$          :
3681 044542 012737 004700 002260          MOV      #4700,CMPTBL          : ELSE SET UP FOR UNIBUS
3682 044550 000403          BR      4$          :
3683 044552 012737 005700 002260 3$:          MOV      #B.S1!B.QB!B.DI!B.OD!B.MP,CMPTBL
3684          :STEP 1 COMPARE VALUE
3685 044560 012737 060050 002252 4$:          MOV      #COMMAR,STPTBL+2          :STEP 2 - COMM AREA ADDRESS
3686 044566 012737 010211 002262          MOV      #010211,CMPTBL+2          :STEP 2 COMPARE
3687 044574 012737 100000 002254          MOV      #B.PP,STPTBL+4          :STEP 3 - HIGH ADDRESS AND PRGE/POLL
3688 044602 112737 000040 002265          MOVB     #40,CMPTBL+5          :REST OF STEP 3 COMPARE
3689 014610 012737 000000 002256          MOV      #0,STPTBL+6          :STEP 4
3690 044616 012737 040000 002266          MOV      #040000,CMPTBL+6          :STEP 4 COMPARE
3691
3692 044624 012737 000012 002306          MOV      #10.,CMARLG          :LENGTH OF COMM AREA FOR THIS TEST
3693 044632 004737 036560          JSR      PC,BAKPAT          :FILL COMM AREA WITH ALL 1'S DATA
3694
3695 044636 004737 036434          JSR      PC,STEP1          :GO DO IT
3696 044642 005737 002320          TST      STEPST          :IF STATUS OKAY
3697 044646 001412          BEQ      5$          : THEN CONTINUE TEST
3698
3699 044650          ERRDF    19.,MSG9,PRIINI          : "SA CONTENTS IN ERROR"
      044650 104455          TRAP     C$ERDF
      044652 000023          .WORD   19
      044654 030465          .WORD   MSG9
      044656 032740          .WORD   PRIINI
3700 044660          CKLOOP          :LOOP ON ERROR?
      044660 104406          TRAP     C$CLP1
3701 044662          DODU     LOGUNT          :DROP UUT
      044662 013700 002312          MOV      LOGUNT,RO
      044666 104451          TRAP     C$DODU
3702 044670          ESCAPE   TST          :LEAVE TST
      044670 104410          TRAP     C$ESCAPE
      044672 000306          .WORD   L10024-.
3703
3704 044674 005237 002316          5$:      INC      INISTP          :ADJUST ST'P COUNTER
3705 044700 062705 000002          ADD      #2,R5          :ADJUST TABLE INDEX
3706 044704 012737 000100 002326 6$:      MOV      #100,OUTER          :SET UP FOR DELAY ROUTINE
3707 044712 016537 002260 002314          MOV      CMPTBL(R5),SAEXP          :SET UP FOR COMPARE
3708 044720 012737 037200 002324 7$:      MOV      #16000.,INNER          :SET UP INNER
3709 044726 017464 000002 000012          MOV      @TKSA(R4),TKSASV(R4)          :GET SA CONTENTS
3710 044734 022705 000006          CMP      #6,R5          :ARE WE IN STEP 4?
3711 044740 001005          BNE      8$          :BRANCH IF NOT
3712 044742 033764 002314 000012          BIT      SAEXP,TKSASV(R4)          :JUST LOOK FOR STEP 4 BIT
3713 044750 001024          BNE      10$          :IT'S SET SO LET'S GO
3714 044752 000404          BR      9$          :STAY IN LOOP OTHERWISE
3715 044754 023764 002314 000012 8$:      CMP      SAEXP,TKSASV(R4)          :IF SA IS WHAT WE EXPECT
3716 044762 001417          BEQ      10$          : THEN MOVE ALONG
3717 044764 004737 036406 9$:      JSR      PC,PDELAY          : ELSE GIVE UUT SOME TIME
3718 044770 005737 002330          TST      TOUT          :IF NO TIMEOUT YET
3719 044774 001751          BEQ      7$          : THEN GO TAKE ANOTHER LOOK
3720
3721 044776          ERRDF    20.,MSG9,PRIINI          : "SA CONTENTS IN ERROR"
      044776 104455          TRAP     C$ERDF
      045000 000024          .WORD   20
      045002 030465          .WORD   MSG9
      045004 032740          .WORD   PRIINI
3722 045006          CKLOOP
    
```



```

045006 104406          TRAP  C#CLP1
3723 045010          DODU  LOGUNT
      045010 013700 002312  MOV  LOGUNT,R0
      045014 104451          TRAP  C#DODU
3724 045016          ESCAPE TST
      045016 104410          TRAP  C#ESCAPE
      045020 000160          .WORD L10024 .
3725
3726 045022 016574 002250 000002 10$:  MOV  STPTBL(R5),@TKSA(R4) ;WRITE NEXT STEP TO UUT
3727 045030 022705 000004          CMP  #4,R5 ;IF STEP 3
3728 045034 001404          BEQ  15$ ; THEN DO PURGE/POLL STUFF
3729 045036 022705 000006          CMP  #6,R5 ;IF NOT IN STEP 4
3730 045042 001314          BNE  5$ ; THEN GO BACK TO MAIN LOOP
3731 045044 000440          BR   20$ ; ELSE GO CHECK RESULTS
3732
3733 045046          15$:  DELAY 1 ;GIVE PORT SOME TIME
      045046 012727 000001          MOV  #1,(PC)+
      045052 000000          .WORD 0
      045054 013727 002116          MOV  L$DLY,(PC)+
      045060 000000          .WORD 0
      045062 005367 177772          DEC  -6(PC)
      045066 001375          BNE  -.4
      045070 005367 177756          DEC  -22(PC)
      045074 001367          BNE  -.20
3734 045076 017464 000002 000012          MOV  @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
3735 045104 001412          BEQ  16$ ;BRANCH IF OKAY
3736
3737 045106          ERRDF 21.,EMSG13,PRIINI ;SA NOT 0 IN PURGE/POLL
      045106 104455          TRAP  C#ERDF
      045110 000025          .WORD 21
      045112 030650          .WORD EMSG13
      045114 032740          .WORD PRIINI
3738 045116          CKLOOP
      045116 104406          TRAP  C#CLP1
3739 045120          DODU  LOGUNT
      045120 013700 002312  MOV  LOGUNT,R0
      045124 104451          TRAP  C#DODU
3740 045126          ESCAPE TST
      045126 104410          TRAP  C#ESCAPE
      045130 000050          .WORD L10024-.
3741
3742 045132 012774 000000 000002 16$:  MOV  #0,@TKSA(R4) ;WRITE 0'S TO SA
3743 045140 005774 000000          TST  @TKIP(R4) ;AND READ IP
3744 045144 000653          BR   5$ ;GO WAIT FOR NEXT TRANSITION
3745
3746 045146 004737 036610          20$:  JSR  PC,CHKCOM ;GO CHECK COMM AREA
3747 045152 032764 000001 000014          BIT  #DRPFLG,LUNFLG(R4) ;HAS UUT BEEN DROPPED
3748 045160 001005          BNE  T6EXT ;LEAVE NOW IF SO
3749 045162 005337 002272          DEC  ITRCNT ;IF NO MORE ITERATIONS LEFT
3750 045166 001402          BEQ  T6EXT ; THEN LEAVE TEST
3751 045170 000137 044472          JMP  2$ ; ELSE DO IT AGAIN
3752
3753 045174          T6EXT: EXIT TST
      045174 104432          TRAP  C#EXIT
      045176 000002          .WORD L10024-.
3754
3755 045200          ENDTST
    
```

045200  
045200 104401

L10024:  
TRAP C\$ETST

```

3758 .SBTTL TEST 7: MAXIMUM RING BUFFER TEST
3759
3760 045202 STARS
;*****
3761 045202 STARS
;*****
3762 ;
3763 ;TEST 7 - MAXIMUM RING BUFFER TEST
3764 ; THIS TEST IS SIMILAR TO TEST 5, BUT IT WILL UTILIZE
3765 ; THE MAXIMUM ALLOWABLE RING DEPTH AS SPECIFIED IN
3766 ; UQSSP. THIS VALUE IS EQUAL TO 128 COMMAND AND 128
3767 ; RESPONSE SLOTS OF 32 BITS PER SLOT.
3768 ;
3769 045202 STARS
;*****
3770 045202 STARS
;*****
3771
3772 045202 BGNTST
045202 T7::
3773
3774 045202 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;IF UUT NOT DROPPED
3775 045210 001423 BEQ G07 ; THEN DO TEST
3776 045212 RFLAGS FLAGS ;ELSE GO GET SUPRVISOR FLAGS
045212 104421 TRAP C#RFLA
045214 010037 002334 MOV R0,FLAGS
3777 045220 032737 001000 002334 BIT #PNT,FLAGS ;SEE IF WE'RE PRINTING TEST NUMBERS
3778 045226 001412 BEQ 1$ ;NO, DON'T PRINT BYPASSED
3779 045230 PRINTF #BYPASS,L$TEST ; ELSE PRINT THE TEST
045230 013746 002114 MOV L$TEST,-(SP)
045234 012746 047626 MOV #BYPASS,-(SP)
045240 012746 000002 MOV #2,-(SP)
045244 010600 MOV SP,R0
045246 104417 TRAP C#PNTF
045250 062706 000006 ADD #6,SP
3780 045254 1$: EXIT TST ;BYPASSED MESSAGE AND GET OUT
045254 104432 TRAP C#EXIT
045256 000542 .WORD L10025-.
3781
3782 045260 012737 032640 002310 G07: MOV #CTRL,FRUIS ;FAILING FRU IN CASE OF ERROR
3783 045266 012737 000001 002272 MOV #1,ITRCNT ;SET UP FOR ONE TEST ITERATION
3784 045274 022737 000001 002270 CMP #1,PASCNT ;IF FIRST PASS
3785 045302 001403 BEQ 2$ ; THEN START TEST
3786 045304 012737 000012 002272 MOV #10.,ITRCNT ; ELSE DO 10 ITERATIONS
3787
3788 045312 012705 000000 2$: MOV #0,R5 ;SET UP R5 AS INDEX TO STEP TABLES
3789 045316 012737 000001 002316 MOV #1,INISTP ;STEP 1 FOR ERROR PRINTOUT
3790 045324 016437 000004 002250 MOV TKVEC(R4),STPTBL ;PUT VECTOR IN STEP 1
3791 045332 006237 002250 ASR STPTBL ;DIVIDE BY TWO
3792 045336 006237 002250 ASR STPTBL ;DIVIDE BY FOUR
3793 045342 013737 002250 002264 MOV STPTBL,CMPTBL+4 ;PUT VECTOR IN STEP 3 COMPARE
3794 045350 052737 137400 002250 BIS #137400,STPTBL ;REST OF STEP ONE
3795 045356 READBUS ;IF QBUS
045356 1044J7 TRAP C#RDBU
3796 045360 BCOMPLETE 3$ ; THEN BR FOR QBUS SET UP
045360 103404 BCS 3$
3797 045362 012737 004700 002260 MOV #4700,CMPTBL ; ELSE SET UP FOR UNIBUS
    
```

```

3798 045370 000403          BR      4$
3799 045372 012737 005700 002260 3 :    MOV      @B.S1!B.QB!B.DI!B.OO!B.MP,CMPTBL
3800                                     ;STEP 1 COMPARE VALUE
3801 045400 012737 060050 002252 4$:    MOV      @COMMAR,STPTBL+2          ;STEP 2 COMM AREA ADDRESS
3802 045406 012737 010277 002262          MOV      @010277,CMPTBL+2        ;STEP 2 COMPARE
3803 045414 012737 100000 002254          MOV      @B.PP,STPTBL+4        ;STEP 3 - HIGH ADDRESS AND PRGE/POLL
3804 045422 112737 000040 002265          MOV      @40,CMPTBL+5         ;REST OF STEP 3 COMPARE
3805 045430 012737 000000 002256          MOV      @0,STPTBL+6          ;STEP 4
3806 045436 012737 040000 002266          MOV      @040000,CMPTBL+6     ;STEP 4 COMPARE
3807
3808 045444 012737 001002 002306          MOV      @514.,CMARLG         ;LENGTH OF COMM AREA FOR THIS TEST
3809 045452 004737 036560          JSR      PC,BAKPAT           ;FILL COMM AREA WITH ALL 1'S DATA
3810
3811 045456 004737 036434          JSR      PC,STEP1           ;GO DO IT
3812 045462 005737 002320          TST      STEPST             ;IF STATUS OKAY
3813 045466 001412          BEQ      5$                 ; THEN CONTINUE TEST
3814
3815 045470          ERRDF   22.,EMSG9,PRIINI     ;"SA CONTENTS IN ERROR"
      045470 104455          TRAP    C$ERDF
      045472 000026          .WORD   22
      045474 030465          .WORD   EMSG9
      045476 032740          .WORD   PRIINI
3816 045500          CKLOOP
      045500 104406          TRAP    C$CLP1             ;LOOP ON ERROR?
3817 045502          DODU    LOGUNT             ;DROP UUT
      045502 013700 002312          MOV     LOGUNT,RO
      045506 104451          TRAP    C$DODU
3818 045510          ESCAPE  TST               ;LEAVE TST
      045510 104410          TRAP    C$ESCAPE
      045512 000306          .WORD   L10025-.
3819
3820 045514 005237 002316          5$:    INC     INISTP             ;ADJUST STEP COUNTER
3821 045520 062705 000002          ADD     @2,R5              ;ADJUST TABLE INDEX
3822 045524 012737 000100 002326 6$:    MOV     @100,OUTER         ;SET UP FOR DELAY ROUTINE
3823 045532 016537 002260 002314          MOV     CMPTBL(R5),SAEXP   ;SET UP FOR COMPARE
3824 045540 012737 037200 002324 7$:    MOV     @16000.,INNER     ;SET UP INNER
3825 045546 017464 000002 000012          MOV     @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
3826 045554 022705 000006          CMP     @6,R5              ;ARE WE IN STEP 4?
3827 045560 001005          BNE     8$                 ;BRANCH IF NOT
3828 045562 033764 002314 000012          BIT     SAEXP,TKSASV(R4)   ;JUST LOOK FOR STEP 4 BIT
3829 045570 001024          BNE     10$                ;IT'S SET SO LET'S GO
3830 045572 000404          BR      9$                 ;STAY IN LOOP OTHERWISE
3831 045574 023764 002314 000012 8$:    CMP     SAEXP,TKSASV(R4)   ;IF SA IS WHAT WE EXPECT
3832 045602 001417          BEQ     10$                ; THEN MOVE ALONG
3833 045604 004737 036406          9$:    JSR     PC,PDELAY         ; ELSE GIVE UUT SOME TIME
3834 045610 005737 002330          TST     TOUT              ;IF NO TIMEOUT YET
3835 045614 001751          BEQ     7$                 ; THEN GO TAKE ANOTHER LOOK
3836
3837 045616          ERRDF   23.,EMSG9,PRIINI     ;"SA CONTENTS IN ERROR"
      045616 104455          TRAP    C$ERDF
      045620 000027          .WORD   23
      045622 030465          .WORD   EMSG9
      045624 032740          .WORD   PRIINI
3838 045626          CKLOOP
      045626 104406          TRAP    C$CLP1             ;LOOP ON ERROR?
3839 045630          DODU    LOGUNT             ;DROP UUT
      045630 013700 002312          MOV     LOGUNT,RO
    
```

```

3840 045634 104451      TRAP      C%DODU
      045636          ESCAPE     TST
      045636 104410      TRAP      C$ESCAPE
      045640 000160      .WORD    L10025-.

3841
3842 045642 016574 002250 000002 10$:  MOV      STPTBL(R5),@TKSA(R4)  ;WRITE NEXT STEP TO UUT
3843 045650 022705 000004          CMP      #4,R5              ;IF STEP 3
3844 045654 001404          BEQ      15$                ; THEN DO PURGE/POLL STUFF
3845 045656 022705 000006          CMP      #6,R5              ;IF NOT IN STEP 4
3846 045662 001314          BNE      5$                 ; THEN GO BACK TO MAIN LOOP
3847 045664 000440          BR       20$                ; ELSE GO CHECK RESULTS
3848
3849 045666          15$:  DELAY     1                    ;GIVE PORT SOME TIME
      045666 012727 000001      MOV      #1,(PC)+
      045672 000000          .WORD    0
      045674 013727 002116      MOV      L$DLY,(PC)+
      045700 000000          .WORD    0
      045702 005367 177772      DEC      -6(PC)
      045706 001375          BNE      -.4
      045710 005367 177756      DEC      -22(PC)
      045714 001367          BNE      -.20
3850 045716 017464 000002 000012      MOV      @TKSA(R4),TKSASV(R4) ;GET SA CONTENTS
3851 045724 001412          BEQ      16$                ;BRANCH IF OKAY
3852
3853 045726          ERRDF    24.,EMSG13,PRIINI  ;SA NOT 0 IN PURGE/POLL
      045726 104455      TRAP      C$ERDF
      045730 000030          .WORD    24
      045732 030650          .WORD    EMSG13
      045734 032740          .WORD    PRIINI
3854 045736          CKLOOP
      045736 104406      TRAP      C$CLP1
3855 045740          DODU     LOGUNT
      045740 013700 002312      MOV      LOGUNT,R0
      045744 104451      TRAP      C%DODU
3856 045746          ESCAPE     TST
      045746 104410      TRAP      C$ESCAPE
      045750 000050          .WORD    L10025-.

3857
3858 045752 012774 000000 000002 16$:  MOV      #0,@TKSA(R4)        ;WRITE 0'S TO SA
3859 045760 005774 000000          TST      @TKIP(R4)         ;AND READ IP
3860 045764 000653          BR       5$                 ;GO WAIT FOR NEXT TRANSITION
3861
3862 045766 004737 036610          20$:  JSR      PC,CHKCOM         ;GO CHECK COMM AREA
3863 045772 032764 000001 000014      BIT      #DRPFLG,LUNFLG(R4) ;HAS UUT BEEN DROPPED
3864 046000 001005          BNE      T7EXT             ;LEAVE NOW IF SO
3865 046002 005337 002272          DEC      ITRCNT           ;IF NO MORE ITERATIONS LEFT
3866 046006 001402          BEQ      T7EXT             ; THEN LEAVE TEST
3867 046010 000137 045312          JMP      2$                 ; ELSE DO IT AGAIN
3868
3869 046014          T7EXT:  EXIT     TST
      046014 104432      TRAP      C$EXIT
      046016 000002          .WORD    L10025-.

3870
3871 046020          ENDTST
      046020          L10025:
      046020 104401      TRAP      C$ETST
    
```

3874  
 3875  
 3876 046022  
  
 3877 046022  
  
 3878  
 3879  
 3880  
 3881  
 3882  
 3883  
 3884  
 3885  
 3886 046022  
  
 3887 046022  
  
 3888  
 3889 046022  
 046022  
 3890  
 3891 046022 032764 000001 000014  
 3892 046030 001423  
 3893 046032  
 046032 104421  
 046034 010037 002334  
 3894 046040 032737 001000 002334  
 3895 046046 001412  
 3896 046050  
 046050 013746 002114  
 046054 012746 047626  
 046060 012746 000002  
 046064 010600  
 046066 104417  
 046070 062706 000006  
 3897 046074  
 046074 104432  
 046076 000646  
  
 3898  
 3899 046100 005737 002274  
 3900 046104 001002  
 3901 046106  
 046106 104432  
 046110 000634  
 3902 046112 012737 032640 002310  
 3903 046120 012737 000001 002272  
 3904 046126 022737 000001 002270  
 3905 046134 001403  
 3906 046136 012737 000012 002272  
 3907  
 3908 046144 004737 036742  
 3909 046150 012705 000000  
 3910 046154 012737 000001 002316  
 3911 046162 016437 000004 002250  
 3912 046170 006237 002250  
 3913 046174 006237 002250

.SBTTL TEST 8: EXTENDED ADDRESS TEST

```

STARS
;*****
STARS
;*****
;
;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.
;
STARS
;*****
STARS
;*****
    
```

```

T8:: BGNTS(
BIT #DRPFLG,LUNFLG(R4) ;IF UUT NOT DROPPED
BEQ G08 ; 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$PNTF
ADD #6,SP
1$: EXIT TST ;BYPASSED MESSAGE AND GET OUT
TRAP C$EXIT
.WORD L10026-.

G08: TST KTFLEG ;IF MEMORY MANAGEMENT AVAILABLE
BNE 1$ ; THEN DO TEST
EXIT TST ; ELSE GET OUT
TRAP C$EXIT
.WORD L10026-.

1$: MOV #CTRL,FRUIS ;FAILING FRU IN CASE OF ERROR
MOV #1,IIRCNT ;SET UP FOR ONE TEST ITERATION
CMP #1,PASCNT ;IF FIRST PASS
BEQ 2$ ; THEN START TEST
MOV #10.,IIRCNT ; ELSE DO 10 ITERATIONS

2$: JSR PC,INTMMU ;INITIALIZE MMU REGISTERS
3$: 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
    
```

3914	046200	052737	002250	002264	MOV	STPTBL,CMPTBL+4	;PUT VECTOR IN STEP 3 COMPARE
3915	046206	052737	111000	002250	BIS	#111000,STPTBL	;REST OF STEP ONE
3916	046214				READBUS		;IF QBUS
	046214	104407			TRAP	C#RDBU	
3917	046216				BCOMPLETE	11#	; THEN BR FOR QBUS SET UP
	046216	103404			BCS	11#	
3918	046220	012737	004700	002260	MOV	#4700,CMPTBL	; ELSE SET UP FOR UNIBUS
3919	046226	000403			BR	12#	
3920	046230	012737	005700	002260	11#:	MOV	#8.S1!8.QB!8.DI!8.OD!8.MP,CMPTBL
3921							;STEP 1 COMPARE VALUE
3922	046236	012737	060050	002252	12#:	MOV	#COMMAR,STPTBL+2
3923	046244	042737	160000	002252		BIC	#BIT15!BIT14!BIT13,STPTBL+2
3924							;CLEAR THE ACTIVE PAGE FIELD
3925	046252	012737	010222	002262	MOV	#010222,CMPTBL+2	;STEP 2 COMPARE
3926	046260	013737	172346	002332	MOV	KPAR3,TEMP	;GET RELOCATION VALUE
3927	046266	113737	002333	002254	MOVB	TEMP+1,STPTBL+4	;JUST THE HGH BYTE
3928	046274	006237	002254		ASR	STPTBL+4	;MAKE IT THE EXTENDED
3929	046300	006237	002254		ASR	STPTBL+4	; ADDRESS OF THE COMM AREA
3930	046304	052737	100000	002254	BIS	#8.PP,STPTBL+4	;NOW SET PURGE/POLL BIT
3931	046312	112737	000640	002265	MOVB	#40,CMPTBL+5	;REST OF STEP 3 COMPARE
3932	046320	012737	000000	002256	MOV	#0,STPTBL+6	;STEP 4
3933	046326	012737	040000	002266	MOV	#040000,CMPTBL+6	;STEP 4 COMPARE
3934							
3935	046334	012737	000022	002306	MOV	#18.,CMARLG	;LENGTH OF COMM AREA FOR THIS TEST
3936	046342	004737	036560		JSR	PC,BAKPAT	;FILL COMM AREA WITH ALL 1'S DATA
3937							
3938	046346	004737	036434		JSR	PC,STEP1	;GO DO IT
3939	046352	005737	002320		TST	STEPST	;IF STATUS OKAY
3940	046356	001412			BEQ	5#	; THEN CONTINUE TEST
3941							
3942	046360				ERRDF	25.,EMSG9,PRIINI	; "SA CONTENTS IN ERROR"
	046360	104455			TRAP	C#ERDF	
	046362	000031			.WORD	25	
	046364	030465			.WORD	EMSG9	
	046366	032740			.WORD	PRIINI	
3943	046370				CKLOOP		;LOOP ON ERROR?
	046370	104406			TRAP	C#CLP1	
3944	046372				DGDU	LOGUNT	;DROP UUT
	046372	013700	002312		MOV	LOGUNT,R0	
	046376	104451			TRAP	C#DODU	
3945	046400				ESCAPE	TST	;LEAVE TST
	046400	104410			TRAP	C#ESCAPE	
	046402	000342			.WORD	L10026	
3946							
3947	046404	005237	002316		5#:	INC	INISTP
3948	046410	062705	000002			ADD	#2,R5
3949	046414	012737	000100	002326	6#:	MOV	#100,OUTER
3950	046422	016537	002260	002314		MOV	CMPTBL(R5),SAEXP
3951	046430	012737	037200	002324	7#:	MOV	#16000.,INNER
3952	046436	017464	000002	000012		MOV	#TKSA(R4),TKSASV(R4)
3953	046444	022705	000006			CMP	#6,R5
3954	046450	001005				BNE	8#
3955	046452	033764	002314	000012		BIT	SAEXP,TKSASV(R4)
3956	046460	001024				BNE	10#
3957	046462	000404				BR	9#
3958	046464	023764	002314	000012	8#:	CMP	SAEXP,TKSASV(R4)
3959	046472	001417				BEQ	10#

3960	046474	004737	036406		9\$:	JSR	PC,PDELAY		: ELSE GIVE UUT SOME TIME
3961	046500	005737	002330			TST	TOUT		:IF NO TIMEOUT YET
3962	046504	001751				BEQ	7\$		: THEN GO TAKE ANOTHER LOOK
3963									
3964	046506					ERRDF	26.,EMSG9,PRIINI		: "SA CONTENTS IN ERROR"
	046506	104455				TRAP	C\$ERDF		
	046510	000032				.WORD	26		
	046512	030465				.WORD	EMSG9		
	046514	032740				.WORD	PRIINI		
3965	046516					CKLOOP			
	046516	104406				TRAP	C\$CLP1		
3966	046520					DODU	LOGUNT		
	046520	013700	002312			MOV	LOGUNT,R0		
	046524	104451				TRAP	C\$DODU		
3967	046526					ESCAPE	TST		
	046526	104410				TRAP	C\$ESCAPE		
	046530	000214				.WORD	L10026-		
3968									
3969	046532	016574	002250	000002	10\$:	MOV	STPTBL(R5),@TKSA(R4)		:WRITE NEXT STEP TO UUT
3970	046540	022705	000004			CMP	#4,R5		:IF STEP 3
3971	046544	001404				BEQ	15\$		: THEN DO PURGE/POLL STUFF
3972	046546	022705	000006			CMP	#6,R5		:IF NOT IN STEP 4
3973	046552	001314				BNE	5\$		: THEN GO BACK TO MAIN LOOP
3974	046554	000440				BR	20\$		: ELSE GO CHECK RESULTS
3975									
3976	046556				15\$:	DELAY	1		:GIVE PORT SOME TIME
	046556	012727	000001			MOV	#1,(PC)+		
	046562	000000				.WORD	0		
	046564	013727	002116			MOV	L\$DLY,(PC)+		
	046570	000000				.WORD	0		
	046572	005367	17777c			DEC	-6(PC)		
	046576	001375				BNE	-.4		
	046600	005367	177756			DEC	-22(PC)		
	046604	001367				BNE	-.20		
3977	046606	017464	000002	000012		MOV	@TKSA(R4),TKSASV(R4)		:GET SA CONTENTS
3978	046614	001412				BEQ	16\$		:BRANCH IF OKAY
3979									
3980	046616					ERRDF	27.,EMSG13,PRIINI		:SA NOT 0 IN PURGE/POLL
	046616	104455				TRAP	C\$ERDF		
	046620	000033				.WORD	27		
	046622	030650				.WORD	EMSG13		
	046624	032740				.WORD	PRIINI		
3981	046626					CKLOOP			
	046626	104406				TRAP	C\$CLP1		
3982	046630					DODU	LOGUNT		
	046630	013700	002312			MOV	LOGUNT,R0		
	046634	104451				TRAP	C\$DODU		
3983	046636					ESCAPE	TST		
	046636	104410				TRAP	C\$ESCAPE		
	046640	000104				.WORD	L10026-		
3984									
3985	046642	012774	000000	000002	1E\$:	MOV	#0,@TKSA(R4)		:WRITE 0'S TO SA
3986	046650	005774	000000			TST	@TKIP(R4)		:AND READ IP
3987	046654	000553				BR	5\$		:GO WAIT FOR NEXT TRANSITION
3988									
3989	046656	004737	036610		20\$:	JSR	PC,CHKCOM		:GO CHECK COMM AREA
3990	046662	032764	000001	000014		BIT	#DRPFLG,LUNFLG(R4)		:HAS UUT BEEN DROPPED



```
3991 046670 001021      BNE      T8EXT      ;LEAVE NOW IF SO
3992
3993 046672 062737 002000 172346      ADD      #2000,KPAR3 ;POINT TO NEXT 32KWORDS
3994 046700 103406      BCS      25$        ;DON'T ALLOW OVERFLOW IF 4 MBYTES
3995 046702 023737 002120 172346      CMP      L$HIME,KPAR3 ;IF THERE'S NO MORE MEMORY AVAILABLE
3996 046710 103402      BLO      L5$        ; THEN CHECK FOR MORE ITERATIONS
3997 046712 000137 046150      JMP      3$         ; ELSE DO IT AGAIN
3998
3999 046716 005037 177572      L5$:    CLR      MMUSRO ;SHUT DOWN MEMORY MANAGEMENT
4000 046722 005337 002272      DEC      ITRCNT    ;IF NO MORE ITERATIONS LEFT
4001 046726 001402      BEQ      T8EXT    ; THEN LEAVE TEST
4002 046730 000137 046144      JMP      2$         ; ELSE DO IT AGAIN
4003
4004 046734 005037 177572      T8EXT:  CLR      MMUSRO ;MAKE SURE IT'S OFF
4005 046740      EXIT      TST
      046740 104432      TRAP     C$EXIT
      046742 000002      .WORD   L10026
4006
4007 046744      ENDTST
      046744      L10026:
      046744 104401      TRAP     C$ETST
```

```

4010 .SBTTL TEST 9:GET DUST STATUS
4011 STARS
4012 046746 ;:*****
4013 046746 STARS
4014 ;:*****
4015 ;
4016 ;TEST 9 - GET DUST STATUS
4017 ; THE GET DUST STATUS TEST WILL REQUEST AND TEST THE DUST
4018 ; STATUS OF EACH UNIT UNDER TEST FOR TWO SPECIFIC CASES-
4019 ; 1) NO COMMAND MODIFIERS SET AND 2) ILLEGAL COMMAND
4020 ; MODIFIERS SET. DUST STATUS WILL BE RECEIVED FROM THE
4021 ; UNIT UNDER TEST AFTER THE PROGRAM ISSUES THE GET DUST
4022 ; STATUS COMMAND AVAILABLE IN DUP. THE RESPONSE PACKET
4023 ; RECEIVED FROM THE UNIT WILL BE TESTED AGAINST A KNOWN
4024 ; GOOD MASK.
4025 046746 STARS
4026 046746 ;:*****
4027 STARS
4028 046746 BGNTST
4029 046746 T9::
4030 046746 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
4031 046754 001423 BEQ GO9 ;GO DO TEST IF AVAILABLE
4032 046756 RFLAGS FLAGS ;ELSE GO GET SUPRVISOR FLAGS
4033 046764 032737 001000 002334 TRAP C#RFLA
4034 046772 001412 MOV RO,FLAGS
4035 046774 PRINTF #BYPASS,L#TEST ;SEE IF WE'RE PRINTING TEST NUMBERS
4036 047000 012746 047626 BIT #PNT,FLAGS ;NG, DON'T PRINT BYPASSED
4037 047004 012746 000002 BEQ 1# ; ELSE PRINT THE TEST
4038 047010 010600 MOV L#TEST, (SP)
4039 047012 104417 MOV #BYPASS,-(SP)
4040 047014 062706 000006 MOV #2,-(SP)
4041 047020 104432 1# EXIT TST ;BYPASSED MESSAGE AND GET OUT
4042 047022 000132 TRAP C#EXIT
4043 047024 042764 000010 .WORD L10027-.
4044 047032 012737 032640 002310 GO9: BIC #DUPFLG,LUNFLG(R4) ;CLEAR DUP FLAG
4045 047040 004737 037042 MOV #CTRL,FRUIS ;SET UP FRU POINTER
4046 047044 032764 000001 000014 JSR PC,PRINT ;GO DO A PORT INITIALIZE
4047 047052 001036 BIT #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
4048 047054 012705 002342 BNE T9EXT ;GET OUT IF NOT AVAILABLE
4049 047060 012737 000001 002272 MOV #SCTRLC,R5 ;SET UP TO DO THE SCC COMMAND
4050 047066 012737 032663 023042 MOV #1,ITRCNT ;DO IT ONCE
4051 047074 004737 037274 JSR PC,CLSDRV ;SET UP COMMAND ASCII
4052 047100 032764 000001 000014 BIT #DRPFLG,LUNFLG(R4) ;GO ISSUE THE COMMAND
4053 047106 001020 BNE T9EXT ;IS THE DRIVE AVAILABLE
4054 047110 052764 000010 000014 BIS #DUPFLG,LUNFLG(R4) ;GET OUT IF NOT AVAILABLE
4055 047116 012705 002456 MOV #GDUST,R5 ;SET DUP FLAG FOR FOLLOWING COMMAND
4056 047122 012737 000001 002272 MOV #1,ITRCNT ;SET UP TO DO GET DUST STATUS COMMAND
    
```

4052	047130	012737	032714	023042	MOV	#GDSCMD,CURCMD	:SET UP COMMAND ASCII
4053	047136	004737	037274		JSR	PC,CLSDRV	:GO ISSUE THE COMMAND
4054	047142	042764	000010	000014	BIC	#DUPFLG,LUNFLG(R4)	:CLEAR DUP FLAG
4055	047150				T9EXT: EXIT	TST	
	047150	104432			TRAP	C\$EXIT	
	047152	000002			.WORD	L10027-	
4056	047154				ENDTST		
	047154				L10027:		
	047154	104401			TRAP	C\$ETST	

```

4058          SBTTL TEST 10:EXECUTE LOCAL PROGRAM (Level II microdiagnostics)
4059
4060 047156   STARS
              ;;*****
4061 047156   STARS
              ;;*****
4062
4063          ;TEST 10 EXECUTE LOCAL PROGRAM
4064          ; THIS TEST WILL INVOKE, VIA THE DUP EXECUTE LOCAL
4065          ; PROGRAM COMMAND, THE CONTROLLER RESIDENT LEVEL 2
4066          ; MICRODIAGNOSTICS.
4067          ;
4068 047156   STARS
              ;;*****
4069 047156   STARS
              ;;*****
4070
4071 047156   BGNTST
              T10::
4072
4073 047156   032764 000001 000014   BIT    #DRPFLG,LUNFLG(R4)    ;IS THE DRIVE AVAILABLE
4074 047164   001423                BEQ    GO10                ;GO DO TEST IF AVAILABLE
4075 047166   104421                RFLAGS  FLAGS             ;ELSE GO GET SUPRVISOR FLAGS
              047170   010037 002334   TRAP   C$RFLA
4076 047174   032737 001000 002334   MOV    R0,FLAGS
4077 047202   001412                BIT    #PNT,FLAGS         ;SEE IF WE'RE PRINTING TEST NUMBERS
4078 047204   013746 002114                BEQ    1$                 ;NO, DON'T PRINT BYPASSED
              047210   012746 047626   PRINTF #BYPASS,L$TEST    ; ELSE PRINT THE TEST
              047214   012746 000002   MOV    L$TEST,-(SP)
              047220   010600                MOV    #BYPASS,-(SP)
              047222   104417                MOV    #2,-(SP)
              047224   062706 000006   TRAP   SP,R0
              047230   104432                TRAP   C$PNTF
4079 047230   000372                ADD    #6,SP
              047232   000372                EXIT   TST                 ;BYPASSED MESSAGE AND GET OUT
              047232   000372                TRAP   C$EXIT
              047232   000372                .WORD  L10030 .
4080
4081 047234   042764 000010 000014   GO10: BIC    #DUPFLG,LUNFLG(R4) ;CLEAR DUP FLAG
4082 047242   042764 000020 000014   BIC    #ABTFLG,LUNFLG(R4) ;CLEAR ABORT FLAG
4083 047250   042764 000040 000014   BIC    #CNTRLC,LUNFLG(R4) ;CLEAR CONTROL C FLAG
4084 047256   012737 032640 002310   MOV    #CTRL,FRUIS       ;SET UP FRU POINTER
4085 047264   004737 037042                JSR    PC,PRTIN          ;GO DO A PORT INITIALIZE
4086 047270   032764 000001 000014   BIT    #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
4087 047276   001150                BNE    TSTXEX           ;GET OUT IF NOT AVAILABLE
4088 047300   012705 002342                MOV    #SCTRLC,R5       ;SET UP TO DO THE SCC COMMAND
4089 047304   012737 000001 002272   MOV    #1,ITRCNT         ;DO IT ONCE
4090 047312   012737 032663 023042   MOV    #SCCCMD,CURCMD    ;SET UP COMMAND ASCII
4091 047320   004737 037274                JSR    PC,CLSDRV        ;GO ISSUE THE COMMAND
4092 047324   032764 000001 000014   BIT    #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
4093 047332   001132                BNE    TSTXEX           ;GET OUT IF NOT AVAILABLE
4094 047334   012705 002406                MOV    #ONLINE,R5       ;SET UP TO DO THE ONLINE COMMAND
4095 047340   012737 000001 002272   MOV    #1,ITRCNT         ;DO IT ONCE
4096 047346   012737 032670 023042   MOV    #ONLCMD,CURCMD    ;SET UP COMMAND ASCII
4097 047354   016465 000006 000004   MOV    MSCPUN(R4),P.UNIT(R5) ;PUT THE UNIT NUMBER IN THE PACKET
4098 047362   004737 037274                JSR    PC,CLSDRV        ;GO ISSUE THE COMMAND
4099 047366   032764 000001 000014   BIT    #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
    
```

```

4100 047374 001111      BNE      TSTXEX      ;GET OUT IF NOT AVAILABLE
4101 047376 052764 000010 000014      BIS      #DUPFLG,LUNFLG(R4) ;ALL FOLLOWING COMMANDS ARE DUP
4102 047404 012705 002476      MOV      #XLOCPR,R5      ;SET UP TO DO ELP COMMAND
4103 047410 012737 000001 002272      MOV      #1,IITRCNT      ;DO IT ONCE
4104 047416 012737 032721 023042      MOV      #ELPCMD,CURCMD   ;SET UP COMMAND ASCII
4105 047424 064737 037274      JSR      PC,CLSDRV        ;GO ISSUE THE COMMAND
4106 047430 032764 000001 000014      BIT      #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
4107 047436 001070      BNE      TSTXEX      ;GET OUT IF NOT AVAILABLE
4108 047440 032737 000200 177560 1$:      BIT      #BIT7,RCSR      ;CHECK FOR INPUT FROM KEYBOARD
4109 047446 001413      BEQ      2$            ;IF NONE, CONTINUE
4110 047450 013705 177562      MOV      RBUF,R5          ;GET DATA INPUT FROM KEYBOARD
4111 047454 042705 000200      BIC      #BIT7,R5        ;STRIP PARITY
4112 047460 022705 000003      CMP      #3,R5          ;CHECK FOR "CONTROL C" INPUT
4113 047464 001004      BNE      2$            ;IF NOT, CONTINUE
4114 047466 052764 000040 000014      BIS      #CNTRLC,LUNFLG(R4) ;SET "CONTROL C" FLAG
4115 047474 000432      BR       20$           ;GO ABORT MICRODIAGNOSTICS
4116 047476 012705 002524      MOV      #RCV DAT,R5     ;SET UP TO DO RECEIVE DATA COMMAND
4117 047502 012737 000001 002272      MOV      #1,IITRCNT      ;DO IT ONCE
4118 047510 012737 032726 023042      MOV      #RCV CMD,CURCMD ;SET UP COMMAND ASCII
4119 047516 004737 037274      JSR      PC,CLSDRV        ;GO ISSUE THE COMMAND
4120 047522 032764 000001 000014      BIT      #DRPFLG,LUNFLG(R4) ;IS THE DRIVE AVAILABLE
4121 047530 001033      BNE      TSTXEX      ;GET OUT IF NOT AVAILABLE
4122 047532 032764 000020 000014      BIT      #ABTFLG,LUNFLG(R4) ;ABORT LOCAL PROGRAM ?
4123 047540 001010      BNE      20$           ;YES, ISSUE ABORT COMMAND
4124 047542 005002      CLR      R2             ;NO, SET UP DELAY LOW COUNT
4125 047544 012703 000020      MOV      #20,R3          ;SET UP DELAY HIGH COUNT
4126 047550 005202      INC      R2             ;DELAY
4127 047552 001376      BNE      10$           ;
4128 047554 005303      DEC      R3             ;
4129 047556 001374      BNE      10$           ;
4130 047560 000727      BR       1$            ;
4131 047562 012705 002554      MOV      #ABORT,R5       ;POLL DIAGL2 PROGRESS
4132 047566 012737 000001 002272      MOV      #1,IITRCNT      ;SET UP TO DO ABORT COMMAND
4133 047574 012737 032733 023042      MOV      #ABT CMD,CURCMD ;DO IT ONCE
4134 047602 004737 037274      JSR      PC,CLSDRV        ;SET UP COMMAND ASCII
4135 047606 032764 000040 000014      BIT      #CNTRLC,LUNFLG(R4) ;GO ISSUE THE COMMAND
4136 047614 001401      BEQ      TSTXEX      ;CHECK IF ABORTING DUE TO "CONTROL C"
4137 047616      BREAK      ;IF NOT, EXIT
      047616 104422      TRAP     C$BRK
4138 047620      TSTXEX: EXIT      TST
      047620 104432      TRAP     C$EXIT
      047622 000002      .WORD   L10030-.
4139 047624      L10030: ENDTST
      047624 104401      TRAP     C$ETST
4140 047626      045      116      045 BYPASS:: .ASCIZ  /*N*A TEST *Z3*A BYPASSED*/
4141      .EVEN
4142 047662      ENDMOD
4143      .TITLE  PARAMETER CODING
4144
4145      .SBTTL  HARDWARF PARAMETER CODING SECTION
4146
4147      BGNMOD
4148
4149      ;++
4150      ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
4151      ; THAT ARE USED BY THE SUPERVISOR TO BUILD P TABLES. THE
4152
4153
4154
4155
4156
4157
4158
    
```

```

4189 ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
4190 ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
4191 ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
4192 ; WITH THE OPERATOR.
4193 ;
4194 ;
4195 047662 BGNHRD
      047662 000044 .WORD L10031-L$HARD/2
      047664 L$HARD::

4196
4202
4203 047664 GPRMA TKIPAD,0,0,160002,177564,YES
      047664 000031 .WORD T$CODE
      047666 047722 .WORD TKIPAD
      047670 160002 .WORD T$LLOLIM
      047672 177564 .WORD T$HILIM
4204 047674 GPRMD TKVECT,2,0,777,60,776,YES
      047674 001032 .WORD T$CODE
      047676 047737 .WORD TKVECT
      047700 000777 .WORD 777
      047702 000060 .WORD T$LLOLIM
      047704 000776 .WORD T$HILIM
4205 047706 GPRMD TKUNT,4,0,777,0,251,YES
      047706 002032 .WORD T$CODE
      047710 047751 .WORD TKUNT
      047712 000777 .WORD 777
      047714 000000 .WORD T$LLOLIM
      047716 000251 .WORD T$HILIM

4206
4207 047720 EXIT HRD
      047720 026004 .WORD T$CODE

4208
4209 047722 124 113 111 TKIPAD: .ASCIZ ?TKIP ADDRESS?
4210 047737 124 113 040 TKVECT: .ASCIZ ?TK VECTOR?
4211 047751 124 057 115 TKUNT: .ASCIZ ?T/MSCP UNIT NUMBER?
4212 .EVEN
4213
4214
4215 047774 ENDHRD
      047774 L10031: .EVEN

4216
4223
    
```

```

4226          .SBTTL  SOFTWARE PARAMETER CODING SECTION
4227
4228          ;**
4229          ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
4230          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
4231          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
4232          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
4233          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
4234          ; WITH THE OPERATOR.
4235          ;--
4236
4237 047774          BGNSFT
          047774 000000          .WORD L10032-L$SOFT/2
          047776          L$SOFT::

4238
4245
4246          .EVEN
4247
4248 047776          ENDSFT
          047776          .EVEN
          L10032:

4249
4250
4260          ;*****
4261          ;*****
4262          ;
4263          ;COMMUNICATIONS AREA
4264          ; THIS IS THE COMMUNICATIONS AREA THAT IS USED
4265          ; THROUGHOUT THE PROGRAM IN TESTING THE PERMUTATIONS
4266          ; OF THE UQ-PORT INIT SEQUENCE.  IT IS ESSENTIAL THAT
4267          ; THIS AREA RESIDE IN AN 8KBYTE AREA OF MEMORY FREE
4268          ; OF DIAGNOSTIC CODE SO THAT IT MAY BE SUCCESSFULLY
4269          ; RELOCATED THROUGHOUT UPPER MEMORY VIA MEMORY MAN-
4270          ; AGEMENT.
4271          ;
4272          ;*****
4273          ;*****
4274
4275          067000          .-60000          ;START OF THE THIRD 8KBYTE BLOCK
4276          ;OF VIRTUAL MEMORY SPACE.  ACCESIBLE
4277          ;VIA PAR/PDR 2.
4278 060000          RDBUF::
4279 060000          COMMBF::
4280 060000          .BLKW 20.          ;BUFFER SPACE PRECEDING COMM AREA
4281 060050          COMMAR::
4282 060050          .BLKW 514.          ;MAXIMUM COMM AREA LENGTH
4283 062054          LASTBF::
4284 062054          .BLKW 20.          ;BUFFER SPACE SUCCEEDING COMM AREA
4288
4289 062124          LASTAD
          062124 000000          .EVEN
          062126 000000          .WORD 0
          062130          .WORD 0
          L$LAST::
4290 062130          ENDMOD
4291          .END

```

ABORT 002554	CMDCNT 003034 G	C\$MMU = 000103	EXIT 040610	G\$RADL = 000120
ABTCMD 032733 G	CMDREF 003032 G	C\$MSG = 000023	EXTINT 036070	G\$RADG = 000020
ABTFLG = 000020 G	CMDRNG 003014 G	C\$OPNR = 000034	EXTVEC 036404	G\$XFER = 000004
ADR = 000020 G	CMDSAV 023036 G	C\$OPNW = 000104	E\$END = 002100	G\$YES = 000010
ASSEMB = 000010	CMHERR 002302 G	C\$PNTB = 000014	E\$LOAD = 000035	MBFADD 023362
BAKPAT 036560 G	CMPTBL 002260 G	C\$PNTF = 000017	FLAG = 040000 G	HELP = 000000
BIT0 = 000001 G	CMTBLG 002304 G	C\$PNTS = 000016	FLAGS 002334 G	HIADDR = 000002 G
BIT00 = 000001 G	CNTER = 000000 G	C\$PNTX = 000015	FRUIS 002310 G	HOE = 100000 G
BIT01 = 000002 G	CNTERR 023322	C\$PUTB = 000072	F\$AU = 000015	HSTIMO = 000000 G
BIT02 = 000004 G	CNTFLG 003026 G	C\$PUTW = 000073	F\$AUTO = 000020	IBE = 010000 G
BIT03 = 000010 G	CNTHI 003024 G	C\$QIO = 000377	F\$BGN = 000040	IDU = 000040 G
BIT04 = 000020 G	CNTRLC = 000040 G	C\$RDBU = 000007	F\$CLEA = 000007	IER = 020000 G
BIT05 = 000040 G	COMMAR 060050 G	C\$REFG = 000047	F\$DU = 000016	ILLINT 036072 G
BIT06 = 000100 G	COMMBF 060000 G	C\$REL = 000077	F\$END = 000041	ILOO = 037072
BIT07 = 000200 G	CONID = 177777 G	C\$RESE = 000033	F\$HARD = 000004	IMM = 000200 G
BIT08 = 000400 G	CRD = 177776 G	C\$REVI = 000004	F\$HW = 000013	IMSG 041046
BIT09 = 001000 G	CTRL 032640 G	C\$RFLA = 000021	F\$INIT = 000006	INISTP 002316 G
BIT1 = 000002 G	CURCMD 023042 G	C\$RPT = 000025	F\$JMP = 000050	INNER 002324 G
BIT10 = 002000 G	C\$AU = 000052	C\$SEFG = 000046	F\$MOD = 000000	INTFLG = 000002 G
BIT11 = 004000 G	C\$AUTO = 000061	C\$SPRI = 000041	F\$MSG = 000011	INTMMU 036742 G
BIT12 = 010000 G	C\$BRK = 000022	C\$SVEC = 000037	F\$PROT = 000021	INTRCV 036062 G
BIT13 = 020000 G	C\$BSEG = 000004	C\$TOME = 000076	F\$PWR = 000017	INTTBL 037264
BIT14 = 040000 G	C\$BSUB = 000002	DCERR 023312	F\$RPT = 000012	INVSTA 023332
BIT15 = 100000 G	C\$CLCK = 000062	DFPTBL 002202 G	F\$SEG = 000003	ISR = 000100 G
BIT2 = 000004 G	C\$CLEA = 000012	DIAGMC = 000000	F\$SOFT = 000005	ITRCNT 002272 G
BIT3 = 000010 G	C\$CLOS = 000035	DRPFLG = 000001 G	F\$SRV = 000010	IXE = 004000 G
BIT4 = 000020 G	C\$CLP1 = 000006	DRVE 032655 G	F\$SUB = 000002	I\$AU = 000041
BIT5 = 000040 G	C\$CPBF = 000074	DRVER = 000011 G	F\$SW = 000014	I\$AUTO = 000041
BIT6 = 000100 G	C\$CPME = 000075	DSCEND 003024 G	F\$TEST = 000001	I\$CLN = 000041
BIT7 = 000200 G	C\$CVEC = 000036	DSCRNG 003000 G	GDSCMD 032714 G	I\$DU = 000041
BIT8 = 000400 G	C\$DCLN = 000044	DUPFLG = 000010 G	GDUST 002456	I\$HRD = 000041
BIT9 = 001000 G	C\$DODU = 000051	DUSTFL = 000017 G	GO = 000001 G	I\$INIT = 000041
BOE = 000400 G	C\$DRPT = 000024	EF.CON = 000036 G	GOABO 040600	I\$MOD = 000041
BPERR 023342	C\$DU = 000053	EF.NEW = 000035 G	GO10 047234	I\$MSG = 000041
BRFLAG = 000004 G	C\$EDIT = 000000	EF.PWR = 000034 G	GO2 041726	I\$PROT = 000040
BYPASS 047626 G	C\$ERDF = 000055	EF.RES = 000037 G	GO3 042510	I\$PTAB = 000041
B.DI = 000400 G	C\$ERHR = 000056	EF.STA = 000040 G	GO4 043176	I\$PWR = 000041
B.ER = 100000 G	C\$ERRO = 000060	ELPCMD 032721 G	GO5 043772	I\$RPT = 000041
B.GO = 000001 G	C\$ERSF = 000054	EMSG10 030512 G	GO6 044440	I\$SEG = 000041
B.IE = 000200 G	C\$ERSO = 000057	EMSG11 030540 G	GO7 045260	I\$SETU = 000041
B.LF = 000002 G	C\$ESCA = 000010	EMSG12 030601 G	GO8 046100	I\$SFT = 000041
B.MP = 000100 G	C\$ESEG = 000005	EMSG13 030650 G	GO9 047024	I\$SRV = 000041
B.NV = 002000 G	C\$ESUB = 000003	EMSG14 030677 G	G\$CNT0 = 000200	I\$SUB = 000041
B.OD = 000200 G	C\$ETST = 000001	EMSG15 030726 G	G\$DELM = 000372	I\$TST = 000041
B.PI = 000001 G	C\$EXIT = 000032	MSG5 030334 G	G\$DISP = 000003	J\$JMP = 000167
B.PP = 100000 G	C\$FREQ = 000101	MSG6 030355 G	G\$EXCP = 000400	KPAR0 = 172340 G
B.QB = 001000 G	C\$FRME = 000100	MSG7 030406 G	G\$HILI = 000002	KPAR1 = 172342 G
B.S1 = 004000 G	C\$GETB = 000026	MSG8 030427 G	G\$LOLI = 000001	KPAR2 = 172344 G
B.S2 = 010000 G	C\$GETW = 000027	MSG9 030465 G	G\$NO = 000000	KPAR3 = 172346 G
B.S3 = 020000 G	C\$GMAN = 000043	END 041042	G\$OFFS = 000400	KPAR4 = 172350 G
B.S4 = 040000 G	C\$GPHR = 000042	ENDCLE 041110	G\$OF SI = 000376	KPAR5 = 172352 G
B.WR = 040000 G	C\$GPRI = 000040	ERR = 100000 G	G\$PRMA = 000001	KPAR6 = 172354 G
CDRECV 037774 G	C\$INIT = 000011	ERRBLK 023060 G	G\$PRMD = 000002	KPAR7 = 172356 G
CHKCOM 036610 G	C\$INLP = 000020	ERRMSG 023056 G	G\$PRML = 000000	KPDR0 = 172300 G
CKCMEX 036740	C\$MANI = 000050	ERRNBR 023054 G	G\$RADA = 000140	KPDR1 = 172302 G
CLSDRV 037274 G	C\$MAP = 000102	ERRTYP 023052 G	G\$RADB = 000000	KPDR2 = 172304 G
CMARLG 002306 G	C\$MEM = 000031	EVL = 000004 G	G\$RADD = 000040	KPDR3 = 172306 G



KPDR4 = 172310 G	L#LUN 002074 G	L2ELDF = 000156 G	L21 024417 G	MSCPER 040520
KPDR5 = 172312 G	L#MREV 002050 G	L2ELDS = 000166 G	L210 025073 G	MSCPUN = 000006 G
KPDR6 = 172314 G	L#NAME 002000 G	L2ELEC = 000140 G	L211 025137 G	MSCPVR = 000000 G
KPDR7 = 172316 G	L#PRIO 002042 G	L2ELEV = 000136 G	L212 025203 G	MSGLEN = 177774 G
KTEXT 036252	L#PROT 023044 G	L2ELFL = 000135 G	L213 025247 G	NEXT 040740
KTFLAG 002274 G	L#PRT 002112 G	L2ELFM = 000134 G	L214 025313 G	NOKT 036246
KTTEST 036102 G	L#REPP 002062 G	L2ELLB = 000162 G	L215 025357 G	NUPASS 040724
LASTBF 062054 G	L#REV 002010 G	L2ELOF = 000172 G	L216 025423 G	ONEFIL = 000001
LINE1 023420 G	L#RPT 035002 G	L2ELPB = 000160 G	L219 025467 G	ONLCMD 032670 G
LINE2 023454 G	L#SOFT 047776 G	L2ELP1 = 000146 G	L22 024473 G	ONLINE 002406
LINE3 023534 G	L#SPC 002056 G	L2ELP2 = 000150 G	L220 025533 G	OP.ABT = 000006 G
LINE4 023564 G	L#SPCP 002020 G	L2ELRL = 000144 G	L221 025601 G	OP.ELP = 000003 G
LINE5 023626 G	L#SPTP 002024 G	L2ELRT = 000145 G	L222 025646 G	OP.END = 000200 G
LINE6 023703 G	L#STA 002030 G	L2ELRW = 000170 G	L223 025710 G	OP.GDS = 000001 G
LINE7 023746 G	L#SW 002212 G	L2ELST = 000154 G	L224 025752 G	OP.ONL = 000011 G
LOE = 040000 G	L#TEST 002114 G	L2ELTN = 000157 G	L225 026011 G	OP.REC = 000005 G
LOGUNT 002312 G	L#TIML 002014 G	L2ELT0 = 000163 G	L226 026053 G	OP.SCC = 000004 G
LOOP 037062	L#UNIT 002012 G	L2ELT1 = 000164 G	L227 026115 G	OUTER 002326 G
LOT = 000010 G	L10000 002210	L2ELT2 = 000165 G	L228 026154 G	OWN = 100000 G
LUNBLK 002212 G	L10001 002212	L2ER1 030763 G	L229 026213 G	O#APTS = 000000
LUNFLG = 000014 G	L10003 033542	L2ER10 031211 G	L23 024531 G	O#AU = 000000
L#ACP 002110 G	L10004 035000	L2ER11 031232 G	L230 026255 G	O#BGNR = 000001
L#APT 002036 G	L10005 036052	L2ER12 031246 G	L231 026317 G	O#BGNS = 000000
L#AU 041144 G	L10006 036060	L2ER13 031266 G	L232 026361 G	O#DU = 000001
L#AUT 002070 G	L10007 036070	L2ER14 031306 G	L233 026423 G	O#ERRT = 000001
L#AUTO = ***** GX	L10010 036100	L2ER15 031334 G	L234 026462 G	O#GNSW = 000000
L#CCP 002106 G	L10011 041076	L2ER16 031364 G	L235 026524 G	O#POIN = 000001
L#CLEA 041100 G	L10012 041126	L2ER17 031412 G	L236 026566 G	O#SETU = 000000
L#CO 002032 G	L10013 041142	L2ER18 031430 G	L237 026630 G	PAROFF 002300 G
L#DEPO 002011 G	L10014 041150	L2ER19 031451 G	L238 026672 G	PASCNT 002270 G
L#DESC 002150 G	L10015 041646	L2ER2 031003 G	L239 026731 G	PCKSIZ 003030 G
L#DESP 002076 G	L10016 041336	L2ER20 031467 G	L24 024571 G	PDELAY 036406 G
L#DEVP 002060 G	L10017 041602	L2ER21 031551 G	L240 026770 G	PDLYEX 036432
L#DISP 002124 G	L10020 042430	L2ER22 031631 G	L241 027027 G	PDRECV 040102 G
L#DLY 002116 G	L10021 043116	L2ER23 031720 G	L242 027071 G	PNT = 001000 G
L#DTP 002040 G	L10022 043712	L2ER24 032010 G	L244 027130 G	PRI = 002000 G
L#DTYP 002034 G	L10023 044360	L2ER25 032076 G	L245 027224 G	PRIDAT 033074 G
L#DU 041130 G	L10024 045200	L2ER26 032167 G	L246 027320 G	PRIERR 033146 G
L#DUT 002072 G	L10025 046020	L2ER27 032261 G	L247 027414 G	PRIEX 033536
L#DVTY 023412 G	L10026 046744	L2ER3 031023 G	L248 027510 G	PRIINI 032740 G
L#EF 002052 G	L10027 047154	L2ER4 031040 G	L249 027604 G	PRIIP 033122 G
L#ENVI 002044 G	L10030 047624	L2ER5 031057 G	L25 024631 G	PRIPAD 033022 G
L#ERRT 023052 G	L10031 047774	L2ER6 031107 G	L250 027700 G	PRISA 032764 G
L#ETP 002102 G	L10032 047776	L2ER7 031127 G	L251 027774 G	PRIVAD 033046 G
L#EXP1 002046 G	L2BRD1 = 000024 G	L2ER8 031145 G	L252 030070 G	PRI00 = 000000 G
L#FXP4 002064 G	L2BRD2 = 000030 G	L2ER9 031160 G	L253 030164 G	PRI01 = 000040 G
L#EXP5 002066 G	L2BWR1 = 000014 G	L2ETBL 023062 G	L254 030260 G	PRI02 = 000100 G
L#HARD 047664 G	L2BWR2 = 000020 G	L2MSG 023312	L26 024671 G	PRI03 = 000140 G
L#HIME 002120 G	L2CMD = 000010 G	L2REP1 = 000050 G	L27 024731 G	PRI04 = 000200 G
L#MPCP 002016 G	L2CRC1 = 000044 G	L2REP2 = 000052 G	L28 024771 G	PRI05 = 000240 G
L#MPTP 002022 G	L2CRC2 = 000046 G	L2RSP = 000011 G	L29 025026 G	PRI06 = 000300 G
L#HW 002202 G	L2DATA 040224 G	L2STA = 000002 G	MMON = 000001 G	PRI07 = 000340 G
L#ICP 002104 G	L2DRV = 000004 G	L2SWR1 = 000034 G	MMUSRO = 177572 G	PRTRV 037632 G
L#INIT 040620 G	L2DUMP 033544 G	L2SWR2 = 000036 G	MMUSR1 = 177574 G	PRINT 037042 G
L#LADP 002026 G	L2ECC1 = 000040 G	L2TRK = 000007 G	MMUSR2 = 177576 G	P.BCNT = 000014 G
L#LAST 062130 G	L2ECC2 = 000042 G	L2TST = 000006 G	MMUSR3 = 172516 G	P.BUFF = 000020 G
L#LOAD 002100 G	L2ELDE = 000155 G	L2UNT = 000012 G	MM220N = 000020 G	P.CRF = 000000 G

P.EXT1= 000014 G	STEPST 002320 G	T\$EXCP= 000000	T\$SOF= 010032	WRCMD 032702 G
P.EXT2= 000015 G	STEP1 036434 G	T\$FLAG= 000041	T\$SRV= 010010	WRCMDE 033256 G
P.EXT3= 000016 G	STPTBL 002250 G	T\$GMAN= 000000	T\$SUB= 010017	WRDATA 002322 G
P.FLGS= 000011 G	STP1ER 036552	T\$HILI= 000251	T\$SM = 010001	WRER1 032347 G
P.IND1= 000020 G	STP1EX 036556	T\$LAST= 000001	T\$TES= 010030	WRER2 032401 G
P.IND2= 000022 G	SVCGBL= 000000	T\$LOLI= 000000	T1 041152 G	WRER3 032422 G
P.MOD = 000012 G	SVCINS= 000000	T\$LSYM= 010000	T1.1 041200	WRER4 032450 G
P.OPCD= 000010 G	SVCSUB= 000000	T\$LTNO= 000012	T1.2 041364	WRER5 032474 G
P.STS = 000012 G	SVCTAG= 000000	T\$NEST= 177777	T10 047156 G	WRER6 032525 G
P.UNIT= 000004 G	SVCTST= 000000	T\$NS0 = 000000	T2 041650 G	WRER7 032546 G
RBUF = 177562 G	S\$LSYM= 010000	T\$NS1 = 000005	T2EXT 042424	WRER8 032614 G
RCSR = 177560 G	S1 = 004000 G	T\$NS2 = 000002	T3 042432 G	WREX 033512 G
RCVCMO 032726 G	TEMP 002332 G	T\$PTNU= 000000	T3EXT 043112	WRINTO 033176 G
RCVDAT 002524	TF.BLK= 000010 G	T\$SAVL= 177777	T4 043120 G	WRPRTE 033222 G
RDBUF 060000 G	TIMOUT= 000024 G	T\$SEGL= 177777	T4EXT 043702	WRSEQE 033410 G
RDCMD 032707 G	TKIP = 000000 G	T\$SUBN= 000000	T5 043714 G	WRTOE 033466 G
RESPBF 002570 G	TKIPAD 047722	T\$TAGL= 177777	T5EXT 044350	WR1 023772 G
REWCMD 032675 G	TKIPSV= 000010 G	T\$TAGN= 010033	T6 044362 G	WR2 024035 G
RNGS*P= 000004 G	TKSA = 000002 G	T\$TEMP= 000000	T6EXT 045174	WR3 024071 G
RSPADD 023352	TKSASV= 000012 G	T\$TEST= 000012	T7 045202 G	WR4 024122 G
RSPBUF 002574 G	TKUNT 047751	T\$TSTM= 177777	T7EXT 046014	WR5 024147 G
RSPEND 003004 G	TKVEC = 000004 G	T\$TSTS= 000001	T8 046022 G	WR6 024211 G
RSPRNG 003004 G	TKVECT 047737	T\$AU = 010014	T8EXT 046734	WR7 024214 G
RSPSAV 023040 G	TOUT 002330 G	T\$CLE= 010012	T9 046746 G	WR8 024265 G
RSPSTP= 000104 G	TRAP4 036054 G	T\$DU = 010013	T9EXT 047150	WR9 024342 G
RSPTO 023402	TRP4FG 002276 G	T\$HAR= 010031	T9FLAG= ***** GX	W1 033322
RSTVEC 036266 G	TSTXEX 047620	T\$HW = 010000	UAM = 000200 G	XLOCPR 002476
SAEXP 002314 G	TXFER = 000005 G	T\$INI= 010011	UNERLG 023372	X\$ALWA= 000000
SCCCMD 032663 G	T\$ARGC= 000002	T\$MSG= 010004	VECTOR 036316 G	X\$FALS= 000040
SCTRLC 002342	T\$CODE= 026004	T\$PRD= 010002	VEC4 = 000004 G	X\$OFFS= 000400
SFPTBL 002212 G	T\$ERRN= 000033	T\$RPT= 010005	WRBUF 003036 G	X\$TRUE= 000020
START 040672				

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

Errors detected: 0

\*\*\* Assembler statistics

Work file reads: 353  
Work file writes: 347  
Size of work file: 34208 Words ( 134 Pages)  
Size of core pool: 19402 Words ( 74 Pages)  
Operating system: RSX-11M/PLUS (Under VAX/VMS)

Elapsed time: 00:18:51.24  
CZTKAB.BIN,CZTKAB.LST/-SP/CR=SVC40R.MLB/ML,CZTKAB.MAC