```
                .REM @
```

IDENTIFICATION

PRODUCT CODE:	AC - FG158 - MC
PRODUCT NAME:	CZTU2B0 TU81 FRONT END FUNC TEST
PRODUCT DATE:	09 - OCT - 1985
MAINTAINER:	TAPE AND OPTICAL DIAGNOSTIC ENGINEERING
AUTHOR:	RAYMOND CHANG

```
49                              REVISION HISTORY
50
51                                      JUL 1985        NEW RELEASE
52
```

1 GENERAL INFORMATION

1.1 Product Description

The TU81 Functional Diagnostic is intended to provide confidence in the basic functionality of the TU81 subsystem. As such, this should be the first host level diagnostic run on the TU81 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 TU81'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 TU81's in a sequential manner. To run a full pass of the program, a scratch tape must be mounted on 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 TU81; 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.

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 TU81 tape drives with controllers
7. 1 to 4 TU81 scratch tapes (optional)
8. LCP-5 UFD software (optional)

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) 260?
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 TU81'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.

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.

```
CZTU2 error eeeee on unit ll test ttt sub sss PC: xxxxxx
SA CONTENTS IN ERROR
INIT SEQUENCE STEP #: n
SA RE: wwwwww EXPCTD: yyyyyy ACTUAL SA: zzzzzz

****FAILING FRU: LESI/CONTROLLER/CABLE****
```

In this example, the fields have the following meanings:

- eeeee = discrete error number as defined by program
- ll = logical unit number assigned to unit-in-error during hardware questions
- ttt = test number during which error occurred
- sss = subtest number
- xxxxxx = program location of error call
- n = step number of the UQSSP initialization sequence which detected the error condition
- wwwwww = physical address of the SA register
- 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.

```
CZTU2 DVC FTL error  eeeee on unit ll test ttt sub sss  PC: xxxxxx
INTERNAL DRIVE TEST FAILED


FAULT CODE: ff    SUB-FAULT CODE: cc
REFER TO PATHFINDER FOR EXPLANATION OF CODES.

****FAILING FRU: DRIVE****
```

In this example, the fields have the following meanings:

- eeeee = see above
- ll = see above
- ttt = see above
- sss = see above
- xxxxxx = see above
- ff = refer to pathfinder
- cc = refer to pathfinder

```
292                     4  FUNCTIONAL DESCRIPTION
293
294                           The following test descriptions all have certain points in
295     common.  All errors specified below will cause the unit to be dropped
296     from the test, unless specifically noted to the contrary.
297     Furthermore, if the operator has chosen loop-on-error (LOE flag set)
298     scope loops will return to the beginning of the test containing the
299     failure.  Exceptions to this will also be noted explicitly below.  To
300     understand the normal four step initialization sequence, refer to the
301     UQSSP; the descriptions of tests that use this sequence will only
302     highlight unique features utilized by that specific test.
303
304
305
306                     4.2     TEST 1   < Existence Verfication Test > -
307
308                      TEST DESCRIPTION:
309
310                           This test verifies the TU81 IP and SA registers can
311                           be accessed on the unibus through the UBA.
312
313
314                      TEST STEPS:
315
316                         BGNTEST
317                             Initialize the Unibus
318                             IF error on initialize
319                                 THEN Print System error and ABORT program
320                             Clear UBA status
321                             IF error on Clear status
322                                 THEN  Print System error and ABORT program
323                             Read the IP register
324                             Wait 100 microseconds for possible Unibus timeout
325                             Read UBA status
326                             IF Unibus timeout error
327                                 THEN Print Fatal device error and drop unit
328                             IF any UBA error
329                                 THEN Print Fatal device error and ABORT program
330                             Read the SA register
331                             Wait 100 microseconds for possible Unibus timeout
332                             Read UBA status
333                             IF any UBA error
334                                 THEN Print Fatal device error and ABORT program
335                         ENDTEST
336
337                      DEBUG:
338
339                           No error looping is allowed all errors abort the test or program
                             The FRU is the Lesi Adapter for all errors in this test.
```

```
                        4.2     TEST 2   < Initialization Test > -

                         TEST DESCRIPTION:

                                This test will do a TU81 controller hard initialize
                                to cause the rom resident power up diagnostics
                                in the tu81 to be run.

                         TEST STEPS:

                            BGNTEST
                                Call dup__ipinit to write to the Ip register to begin
                                        hard initialize and wait for STEP 1.
                                IF the TU81 fails to enter STEP 1
                                    THEN print fatal device error and drop unit
                                Compare step 1 data expd with recv
                                IF data compare error
                                    THEN print fatal device error and drop unit
                            ENDTEST


                         DEBUG:

                                If loop on error specified then loop to start of test.
                                The FRU is the Lesi Adapter for all errors in this test.
```

```
                        4.3     TEST 3 < Initialization Test > -

                         TEST DESCRIPTION:

                               This test will do a TU81 controller hard initialize
                               then do initialization steps 1 through 3.
                               It will wait for step 4 to be entered but no step 4
                               testing will be done in this test.

                         TEST STEPS:

                            BGNTEST
                                Call dup__ipinit to write to the Ip register to begin
                                       hard initialize and wait for STEP 1.
                                IF the TU81 fails to enter STEP 1
                                    THEN print fatal device error and drop unit
                                Compare step 1 data expd with recv
                                IF data compare error
                                    THEN print fatal device error and drop unit

                                Call dup__step1 to write step 1 bit pattern and wait step 2
                                IF the TU81 fails to enter STEP 2
                                    THEN print fatal device error and drop unit
                                Compare step 2 data expd with recv
                                IF data compare error
                                    THEN print fatal device error and drop unit

                                Call dup__step2 to write step 2 bit pattern and wait step 3
                                IF the TU81 fails to enter STEP 3
                                    THEN print fatal device error and drop unit
                                Compare step 3 data expd with recv
                                IF data compare error
                                    THEN print fatal device error and drop unit

                                Call dup__step3 to write step 3 bit pattern and wait step 4
                                IF the TU81 fails to enter STEP 4
                                    THEN print fatal device error and drop unit
                                Compare step 4 data expd with recv
                                IF data compare error
                                    THEN print fatal device error and drop unit
                            ENDTEST


                         DEBUG:

                               If loop on error specified then loop to start of test.
                               The FRU is the Lesi Adapter for all errors in this test.
```

```
                        4.4     TEST 4 < SA Register Wrap Test > -

                        TEST DESCRIPTION:

                               The TU81 will be initialized in diagnostic wrap mode
                               and then a one (1) bit will be floated through the
                               SA register to see that it echoes properly.
                               The process will be repeated to float a zero (0)
                               through the SA register.

                        TEST STEPS:

                           BGNTEST
                               Call dup__ipinit to write to the Ip register to begin
                                       hard initialize and wait for STEP 1.
                               IF the TU81 fails to enter STEP 1
                                  THEN print fatal device error and drop unit
                               Call dup__step__1 to set diagnostic wrap mode
                               REPEAT for all data in FLOAT__table
                                   Write data pattern into SA register
                                   Start a 10 second  timer
                                   Read SA register until the read pattern equals the
                                       write pattern or 10 second timer times out.
                                   IF 10 second timer expired
                                       THEN  Print Fatal device error and drop unit
                               END-REPEAT
                               Call dup__ipinit to write to the Ip register to begin
                                       hard initialize and wait for STEP 1.
                               IF the TU81 fails to enter STEP 1
                                   THEN print fatal device error and drop unit
                           ENDTEST

                               FLOAT__table:
                                       FLOATING 1'S    1,2,4,10,20,40,100,200,400,1000,2000
                                                       4000,10000,20000,40000,100000
                                       FLOATING 0'S    Floating 1's complemented

                        DEBUG:

                               If loop on error specified then loop on failing write and read.
                               The FRU is the Lesi Adapter and tu81 controller
                               for all errors in this test.
```

```
                                4.5     TEST 5   < Vector And BR Level Test > -

                                TEST DESCRIPTION:

                                        The TU81 will be initialized with interrupt enable
                                        set to verify that the TU81 interrupts to the
                                        correct vector and BR level.
                                        This test is only run on the first pass.

                                TEST STEPS:

                                   BGNTEST
                                        Call dup__ipinit to write to the Ip register to begin
                                                hard initialize and wait for STEP 1.
                                        IF the TU81 fails to enter STEP 1
                                            THEN Print fatal device error and drop unit

                                        Set IPL to highest priority to lock out interrupts
                                        Clear UBA status
                                        IF error on Clear status
                                            THEN  Print System error and ABORT program
                                        Enable UBA interrupts
                                        IF error on enable uba interrupts
                                            THEN Print System error and ABORT program

                                        Call dup__step__1 to set interrupt enable
                                        IF the TU81 fails to enter STEP 2
                                            THEN Print Fatal device error and drop unit
                                        (A tu81 step 2 interrupt should be pending here)
                                        Lower the IPL until interrupt occurs or level equals X10 (lowest)
                                        IF no Tu81 interrupt occurred
                                            THEN Print Fatal device error and drop unit
                                        IF any error detected in interrupt service
                                            THEN Print Fatal system error and ABORT test
                                        IF the interrupt occurred at the wrong vector
                                            THEN Print Fatal device error and drop unit
                                        IF the interrupt occurred at the wrong BR level
                                            THEN Print Fatal device error and drop unit

                                        Disable UBA interrupts
                                        IF error on Disable uba interrupts
                                            THEN Print System error and ABORT program

                                        Call dup__ipinit to write to the Ip register to begin
                                                hard initialize and wait for STEP 1.
                                        IF the TU81 fails to enter STEP 1
                                            THEN Print Fatal device error and drop unit
                                   ENDTEST
```

```
510                             DEBUG:
511
512                                     Possible reasons for incorrect interrupt vector include:
513
514                                         1  Incorrect hardware configuration
515                                         2. The ATTACH command specified the wrong vector
516                                         3. Bad Lasi adapter
517                                         4. Bad TUBI controller
518
519                                     If loop on error specified then loop to start of the test
520
521                                     The FRU is the Lasi Adapter and tu8i controller
52?                                     for all errors in this test.
```

```
                                4.6     TEST 6   < Purge And Poll Test > -

                                 TEST DESCRIPTION:

                                        This test will perform steps 1-3 of the initialize sequence
                                        then set the purge/poll bit in step 3.
                                        The purge/poll sequence will then proceed to:
                                            1. Write 0's to the SA register to simulate uba purge complete
                                            2. Read and disregard the IP register to start polling
                                            3. Wait for the controller to go into step 4.

                                 TEST STEPS:

                                    BGNTEST
                                        Call dup__ipinit to write to the Ip register to begin
                                                hard initialize and wait for STEP 1.
                                        IF the TU81 fails to enter STEP 1
                                            THEN Print fatal device error and drop unit
                                        Compare step 1 data expd with recv
                                        IF data compare error
                                            THEN Print fatal device error and drop unit

                                        Call dup__step1 to write step 1 bit pattern and wait step 2
                                        IF the TU81 fails to enter STEP 2
                                            THEN Print fatal device error and drop unit
                                        Compare step 2 data expd with recv
                                        IF data compare error
                                            THEN Print fatal device error and drop unit

                                        Call dup__step2 to write step 2 bit pattern and wait step 3
                                        IF the TU81 fails to enter STEP 3
                                            THEN Print fatal device error and drop unit
                                        Compare step 3 data expd with recv
                                        IF data compare error
                                            THEN Print fatal device error and drop unit
                                *
                                        Call dup__step3 to write purge/poll bit (sa_pp_3)
                                        IF the controller fails to clear the SA within 100 micros
                                            THEN Print fatal device error and drop unit
                                        Write 0's to the SA to simulate uba purge complete
                                        Read and disregard the IP register to start polling
                                *
                                        IF the TU81 fails to enter STEP 4 within 10 seconds
                                            THEN Print fatal device error and drop unit
                                    ENDTEST


                                 DEBUG:

                                        If loop on error specified then loop to start of test.
                                        The FRU is the Lesi Adapter for all errors in this test.
```

```
                        4.7     TEST 7   < Small Ring Test > -

                        TEST DESCRIPTION:

                               This test will do steps 1-4 of the TU81 initialization,
                               with the smallest ring buffer size (1 cmd and 1 rsp buffer)
                               and interrupts disabled.
                               The test will verify the controller clears the ring
                               descriptor field in the host communications area.
                               This is the first time the initialize sequence is carried
                               out to the point where the controller npr's to memory
                               are verified.

                        TEST STEPS:

                           BGNTEST
                               Set cmd and rsp ring descriptors to -1
                               Set cmd ring length word to 0 to indicate 1 cmd buffer
                               Set rsp ring length word to 0 to indicate 1 rsp buffer
                               Call Dup_Init to write to the Ip register to force
                                  a hard initialize, then perform steps 1-4.
                               IF the TU81 fails to enter any step
                                  THEN print fatal device error and drop unit
                               IF the cmd and rsp ring descriptors not cleared
                                  THEN print fatal device error and drop unit
                           ENDTEST


                        DEBUG:

                               If loop on error specified then loop to start of test.
                               The FRU is the Lesi Adapter and TU81 controller
                                  for all errors in this test.
```

```
610                         4.8     TEST 8   < Maximum Ring Buffer Test > -
611
612                         TEST DESCRIPTION:
613
614
615
616                               This test will do steps 1-4 of the TU81 initialization,
617                               with the largest number of ring descriptors allowed
618                               (128 cmd and 128 rsp buffers) and interrupts disabled.
619                               The test will verify the controller clears the ring
620                               descriptor field in the host communications area.
621                               This test verifies the controller can access the complete
622                               host commmunication area in Vax memory (1024+4 words).
623
624                         TEST STEPS:
625
626                            BGNTEST
627                                Set cmd and rsp ring descriptors to -1
628                                Set cmd ring length word to 7 to indicate 128 cmd buffers (2**7=128)
629                                Set rsp ring length word to 7 to indicate 128 rsp buffers (2**7=128)
630                                Call Dup__Init to write to the Ip register to force
631                                    a hard initialize, then perform steps 1-4.
632                                IF the TU81 fails to enter any step
633                                    THEN print fatal device error and drop unit
634                                IF the cmd and rsp ring descriptors not cleared
635                                    THEN print fatal device error and drop unit
636                            ENDTEST
637
638
639                         DEBUG:
640
641                               If loop on error specified then loop to start of test.
642                               The FRU is the Lesi Adapter and TU81 controller
643                                   for all errors in this test.
644                               Note:
645                                   This test overlays the host communications area with
646                                   128 cmd ring descriptors and 128 rsp ring descriptors.
647                                   The actual associated ring buffers are not allocated.
648                                   The rest of the tests use just one cmd and one rsp
649                                   buffer.
```

```
                        4.9     TEST 9   < Get DUST Status > -

                         TEST DESCRIPTION:

                               This test will request the DUST status and verify the
                               response packet is received as expected.
                               It is also verifies invalid command status is returned
                               when illegal modifiers are specified in the command packet.
                               The GET DUST command does not allow any command modifiers.
                               This is the first time a command packet is actually sent to
                               the controller and a response packet received.

                         TEST STEPS:

                            BGNSUB 1 *Get DUST command with valid modifiers*
                                Set cmd and rsp ring descriptors to -1
                                Set cmd ring length word to 0 to indicate 1 cmd buffer
                                Set rsp ring length word to 0 to indicate 1 rsp buffer
                                Call Dup__Init to write to the Ip register to force
                                    a hard initialize, then perform steps 1-4. Go bit set to 1
                                IF the TU81 fails to enter any step
                                    THEN print fatal device error and drop unit
                                IF the cmd and rsp ring descriptors are not cleared
                                    THEN print fatal device error and drop unit
                                Call exe__getdust to execute a GET DUST command
                                IF Exe__getdust returns SS$__TIMEOUT code
                                    THEN print fatal device timeout error and drop unit
                                IF the rsp Command reference number NOT = 1
                                    THEN print hard device error
                                IF the rsp Endcode NOT= (get_dust code + 200 octal)
                                    THEN print hard device error
                                IF the rsp Status NOT= success
                                    THEN print hard device error
                                IF the rsp buffer FLAGS data is NOT as follows:
                                    1. Bit<0> = 1  !du_p_dust__flag__dis     - disable other servers
                                    2. Bit<1> = 1  !dup__dust__flag__media   - server has local media (rom)
                                    3. Bit<2> = 1  !dup__dust__flag__nosup   - exe__supplied cmd not allowed
                                    4. Bit<3> = 0  !dup__dust__fla_g_act         - server not active
                                    THEN print hard device error
                            ENDSUB 1

                            BGNSUB 2 *Get DUST command with illegal modifiers*
                                Call exe__getdust to execute a GET DUST command
                                IF Exe__getdust returns SS$__TIMEOUT code
                                    THEN print fatal device timeout error and drop unit
                                IF the rsp Command reference number NOT = 2
                                    THEN print hard device error
                                IF the rsp Endcode NOT= (get__dust code + 80 hex)
                                    THEN print hard device error
                                IF the rsp Status NOT= INVALID COMMAND
                                    THEN print hard device error
                            ENDSUB 2

                            ENDTEST
```

```
706                             DEBUG:
707
708                                     If loop on error specified then loop to start of test.
709                                     The FRU is the lesi adapter or the TU81 controller/server
710                                     for all errors in this test.
```

```
                        4.10    TEST 10  < Functional Fault Detection Test (Internal Drive Test 1 > -
                        TEST DESCRIPTION:

                              This is a manual (/sec:manual) intervention test that will execute
                              the TU81 internal microdiagnostic _@1.


                        TEST STEPS:

                           BGNTEST <MANUAL>
                               Print message to mount tape untensioned but loaded
                                           "Is the tape ready?"
                               Call dup_init to write to the Ip register to force
                                   a hard initialize, then perform steps 1-4. Go bit set to 1
                               IF the TU81 fails to enter any step
                                   THEN print fatal device error and drop unit

                               Call DUP_EXELOCAL to execute an EXECUTE LOCAL PROGRAM command
                                   IF Dup_exelocal returns SS_GETDUSTMO
                                     THEN print Get dust command timeout
                                   IF Dup_exelocal returns SS_NOTIDLE
                                     THEN print controller not in idle state
                                   IF Dup_exelocal returns SS$_TIMEOUT
                                     THEN print controller failed to return packet
                                   IF Dup_exelocal returns SS_EXEBADREF
                                     THEN print invalid command reference
                                   IF Dup_exelocal returns SS_NOTSUCCESS
                                     THEN print controller failed to return success in packet
                                   IF Dup_EXELOCAL returns SS_DUSTBADREF
                                     THEN print invalid command reference
                                   IF Dup_exelocal returns SS$_DEVINACT
                                     THEN print controller failed to enter active state
                                   IF Dup_exelocal returns SS_RECVTMO
                                     THEN print Controller failed to accept receive data command
                                   IF Dup_exelocal returns SS_PROGTMO
                                     THEN print progress indicator not updated before timeout
                                   IF Dup_exelocal returns SS_RECVINVMSG
                                     THEN print Receive data returned invalid message number
                                   IF Dup_exelocal returns SS_RECVERR2
                                     THEN print Receive data returned internal test failed
                                          and print the message buffer fault code and subcode.
                                          and print refer to SAMS for fault code meanings.
                                   IF Dup_exelocal returns SS_SAERR
                                     THEN print controller error while in execute local program
                           ENDTEST


                        DEBUG:

                              If loop on error specified then loop to start of test.
                              The FRU is lesi Adapter for initialize errors
                              or the TU81 controller/server for all other errors.
```

```
765                          4.11    TEST 11  < Tension Fault Isolation Test (Internal Drive Test 2)>
766
767                          TEST DESCRIPTION:
768
769                                This is a Fault (/sec:Fault) intervention test that will execute
770                                the TU81 internal microdiagnostic _#2.
771                                Internal test _#2 isolates servo faults by checking different
772                                assemblies of the STU.
773
774
775                          TEST STEPS:
776
777                             BGNTEST <Fault>
778                                 Print message "Mount a scratch tape THREADED but UNTENSIONED"
779                                               "Is the tape ready?"
780                                 Call dup__init to write to the Ip register to force
781                                     a hard initialize, then perform steps 1-4. Go bit set to 1
782                                 IF the TU81 fails to enter any step
783                                     THEN print fatal device error and drop unit
784                                 Call DUP_EXELOCAL to execute an EXECUTE LOCAL PROGRAM command
785                                     IF Dup_exelocal returns SS_GETDUSTMO
786                                         THEN print Get dust command timeout
787                                     IF Dup_exelocal returns SS_NOTIDLE
788                                         THEN print controller not in idle state
789                                     IF Dup_exelocal returns SS$_TIMEOUT
790                                         THEN print controller failed to return packet
791                                     IF Dup_exelocal returns SS_EXEBADREF
792                                         THEN print invalid command reference
793                                     IF Dup_exelocal returns SS_NOTSUCCESS
794                                         THEN print controller failed to return success in packet
795                                     IF Dup_EXELOCAL returns SS_DUSTBADREF
796                                         THEN print invalid command reference
797                                     IF Dup_exelocal returns SS$_DEVINACT
798                                         THEN print controller failed to enter active state
799                                     IF Dup_exelocal returns SS_RECVTMO
800                                         THEN print Controller failed to accept receive data command
801                                     IF Dup_exelocal returns SS_PROGTMO
802                                         THEN print progress indicator not updated before timeout
803                                     IF Dup_exelocal returns SS_RECVINVMSG
804                                         THEN print Receive data returned invalid message number
805                                     IF Dup_exelocal returns SS_RECVERR2
806                                         THEN print Receive data returned internal test failed
807                                              and print the message buffer fault code and subcode.
808                                              and print refer to SAMS for fault code meanings.
809                                     IF Dup_exelocal returns SS_SAERR
810                                         THEN print controller error while in execute local program
811                             ENDTEST
812
813
814                          DEBUG:
815
816                                If loop on error specified then loop to start of test.
```

```
818                          4.12    TEST 12  < Velocity Fault Isolation Test (Internal Drive Test 3)> -
819
820                          TEST DESCRIPTION:
821
822                                This is a Fault (/sec:Fault) intervention test that will execute
823                                the TU81 internal microdiagnostic _#3.
824                                Internal test _#3 isolates velocity servo faults by checking
825                                the take__up motor/tach assembly and the velocity servo
826                                loop.
827
828
829                          TEST STEPS:
830
831                             BGNTEST <Fault>
832                                 Print message "Remove the tape from the drive"
833                                               "Is the tape REMOVED?"
834                                 Call dup__init to write to the Ip register to force
835                                     a hard initialize, then perform steps 1-4. Go bit set to 1
836                                 IF the TU81 fails to enter any step
837                                     THEN print fatal device error and drop unit
838                                 Call DUP_EXELOCAL to execute an EXECUTE LOCAL PROGRAM command
839                                     IF Dup_exelocal returns SS_GETDUSTMO
840                                       THEN print Get dust command timeout
841                                     IF Dup_exelocal returns SS_NOTIDLE
842                                       THEN print controller not in idle state
843                                     IF Dup_exelocal returns SS$_TIMEOUT
844                                       THEN print controller failed to return packet
845                                     IF Dup_exelocal returns SS_EXEBADREF
846                                       THEN print invalid command reference
847                                     IF Dup_exelocal returns SS_NOTSUCCESS
848                                       THEN print controller failed to return success in packet
849                                     IF Dup_EXELOCAL returns SS_DUSTBADREF
850                                       THEN print invalid command reference
851                                     IF Dup_exelocal returns SS$_DEVINACT
852                                       THEN print controller failed to enter active state
853                                     IF Dup_exelocal returns SS_RECVTMO
854                                       THEN print Controller failed to accept receive data command
855                                     IF Dup_exelocal returns SS_PROGTMO
856                                       THEN print progress indicator not updated before timeout
857                                     IF Dup_exelocal returns SS_RECVINVMSG
858                                       THEN print Receive data returned invalid message number
859                                     IF Dup_exelocal returns SS_RECVERR2
860                                       THEN print Receive data returned internal test failed
861                                            and print the message buffer fault code and subcode.
862                                            and print refer to SAMS for fault code meanings.
863                                     IF Dup_exelocal returns SS_SAERR
864                                       THEN print controller error while in execute local program
865                             ENDTEST
866
867
868                          DEBUG:
869
870                                If loop on error specified then loop to start of test.
871                                The FRU is lesi Adapter for initialize errors
872                                or the TU81 controller/server for all other errors.
```

```
874                          4.13    TEST 13 < Select A Drive Resident Test (Internal Drive Tests 1-99) > -
875
876                          TEST DESCRIPTION:
877
878                                 This section (/sec:FAULT) will ask the operator
879                                 to select a drive resident microdiagnostic. The resident
880                                 test will be started using the Dup Execute local program
881                                 function and monitored by Dup Get Dust status function calls.
882                                 The internal tests are described in the Drive maintenance manual.
883
884
885                          TEST STEPS:
886
887                             BGNTEST <FAULT>
888
889                                 Print message   "Enter drive unit number :"
890                                     IF the unit number is invalid
891                                         THEN Print error message and ask again
892                                 Print message   "Enter controller internal test number <1-99>:"
893                                     IF the resident test name is not in the valid name table
894                                         THEN Print error message and ask again
895
896                                 Print message   "Setup the tape drive per the Maintenance
897                                                  manual for this internal test
898                                                  READY?
899                                 Accept any response as ready
900
901                                 Call dup__init to write to the Ip register to force
902                                     a hard initialize, then perform steps 1-4. Go bit set to 1
903                                 IF the TU81 fails to enter any step
904                                     THEN print fatal device error and drop unit
905                                 Call DUP_EXELOCAL to execute an EXECUTE LOCAL PROGRAM command
906                                     IF Dup_exelocal returns SS_GETDUSTMO
907                                         THEN print Get dust command timeout
908                                     IF Dup_exelocal returns SS_NOTIDLE
909                                         THEN print controller not in idle state
910                                     IF Dup_exelocal returns SS$_TIMEOUT
911                                         THEN print controller failed to return packet
912                                     IF Dup_exelocal returns SS_EXEBADREF
913                                         THEN print invalid command reference
914                                     IF Dup_exelocal returns SS_NOTSUCCESS
915                                         THEN print controller failed to return success in packet
916                                     IF Dup_EXELOCAL returns SS_DUSTBADREF
                                            THEN print invalid command reference
917                                     IF Dup_exelocal returns SS$_DEVINACT
919                                         THEN print controller failed to enter active state
920                                     IF Dup_exelocal returns SS_RECVTMO
921                                         THEN print Controller failed to accept receive data command
922                                     IF Dup_exelocal returns SS_PROGTMO
923                                         THEN print progress indicator not updated before timeout
924                                     IF Dup_exelocal returns SS_RECVINVMSG
925                                         THEN print Receive data returned invalid message number
```

```
                                        IF Dup_exelocal returns SS_RECVERR2
                                            THEN print Receive data returned internal test failed
                                                 and print the message buffer fault code and subcode.
                                                 and print refer to SAMS for fault code meanings.
                                        IF Dup_exelocal returns SS_RECVMSG3
                                            THEN print contents of receive data message buffer (not an error)

                                        IF Dup_exelocal returns SS_SAERR
                                            THEN print controller error while in execute local program
                                    ENDTEST


                                 DEBUG:

                                        If loop on error specified then loop to start of test.
                                        The FRU is lesi Adapter for initialize errors
                                        or the TU81 controller/server for all other errors.
                                @
                                .TITLE PROGRAM HEADER AND TABLES
                                .SBTTL  PROGRAM HEADER

                                        .ENABL  ABS,AMA
000000                          .       =       2000
        002000
                                        .NLIST  BEX

002000                                  BGNMOD

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

002000                                  POINTER BGNDU,ERRTBL,BGNRPT


002000                                  HEADER  CZTU2.B,0,120.,0,PRI00
002000                          L$NAME::                ;DIAGNOSTIC NAME
002000  103                             .ASCII /C/
002001  132                             .ASCII /Z/
002002  124                             .ASCII /T/
002003  125                             .ASCII /U/
002004  062                             .ASCII /2/
002005  000                             .BYTE   0
002006  000                             .BYTE   0
002007  000                             .BYTE   0
002010                          L$REV::                 ;REVISION LEVEL
002010  102                             .ASCII  /B/
002011                          L$DEPO::                ;0
002011  060                             .ASCII  /0/
002012                          L$UNIT::                ;NUMBER OF UNITS
002012  000000                          .WORD   0
002014                          L$TIML::                ;LONGEST TEST TIME
002014  000170                          .WORD   120.
002016                          L$HPCP::                ;POINTER TO H.W. QUES.
002016  043110                          .WORD   L$HARD
```

PROGRAM HEADER

```
002020                          L$SPCP::                ;POINTER TO S.W. QUES.
002020  000000                  .WORD   0
002022                          L$HPTP::                ;PTR. TO DEF. H.W. PTABLE
002022  002224                  .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::
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$PRIO::                ;DIAGNOSTIC RUN PRIORITY
002042  C00000                  .WORD   PRIOO
002044                          L$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$DEVP::                ; POINTER TO DEVICE TYPE LIST
002060  022766                  .WORD   L$DVTYP
002062                          L$REPP::                ;PTR. TO REPORT CODE
002062  000000G                 .WORD   L$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  033774                  .WORD   L$DU
002074                          L$LUN::                 ;LUN FOR EXERCISERS TO FILL
002074  000000                  .WORD   0
002076                          L$DESP::                ;POINTER TO DIAG. DESCRIPTION
002076  002156                  .WORD   L$DESC
002100                          L$LOAD::                ;GENERATE SPECIAL AUTOLOAD EMT
002100  104035                  EMT     E$LOAD
002102                          L$ETP::                 ;POINTER TO ERRTBL
002102  000000G                 .WORD   L$ERRTBL
002104                          L$ICP::                 ;PTR. TO INIT CODE
002104  033464                  .WORD   L$INIT
002106                          L$CCP::                 ;PTR. TO CLEAN-UP CODE
002106  033744                  .WORD   L$CLEAN
002110                          L$ACP::                 ;PTR. TO AUTO CODE
```

PROGRAM HEADER

```
002110  000000G                 .WORD   L$AUTO
002112                  L$PRT::                 ;PTR. TO PROTECT TABLE
002112  022760                  .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
1009
```

```
                                .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                                  DISPATCH 13.
002122  000015                          .WORD   13
002124                          L$DISPATCH::
002124  034016                          .WORD   T1
002126  034514                          .WORD   T2
002130  034744                          .WORD   T3
002132  035362                          .WORD   T4
002134  036124                          .WORD   T5
002136  037256                          .WORD   T6
002140  040702                          .WORD   T7
002142  041444                          .WORD   T8
002144  042206                          .WORD   T9
002146  042270                          .WORD   T10
002150  042452                          .WORD   T11
002152  042574                          .WORD   T12
002154  042716                          .WORD   T13

002156                                  DESCRIPT        <CZTU28O TU81 FUNCTIONAL DIAGNOSTIC>
002156                          L$DESC::
002156  103  132  124                   .ASCIZ  /CZTU28O TU81 FUNCTIONAL DIAGNOSTIC/
                                        .EVEN
```

```
                                .SBTTL  DEFAULT HARDWARE P TABLE

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

002222                                  BGNHW   DFPTBL
002222  000003                          .WORD   L10000-L$HW/2
002224                          L$HW::
002224                          DFPTBL::

002224  174500                          .WORD   174500          ;TUIP BASE ADDRESS
002226  000260                          .WORD   260             ;VECTOR
002230  000000                          .WORD   0               ;T/MSCP UNIT NUMBER
002232                                  ENDHW
002232                          L10000:
```

```
1056                                    .SBTTL  SOFTWARE P-TABLE
1057
1058                                    ;++
1059                                    ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
1060                                    ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
1061                                    ;--
1062
1063 002232                                     BGNSW   SFPTBL
     002232  000000                            .WORD   L10001-L$SW/2
     002234                             L$SW::
     002234                             SFPTBL::
1064
1071
1072 002234                                     ENDSW
     002234                             L10001:
1073
1074 002234                                     ENDMOD
1086                                    .TITLE GLOBAL AREAS
1087                                    .SBTTL  GLOBAL EQUATES SECTION
1115
1116
1117 002234                                     BGNMOD
1118
1119                                    ;++
1120                                    ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
1121                                    ; ARE USED IN MORE THAN ONE TEST.
1122                                    ;--
1123
1124 002234                                     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
```

GLOBAL EQUATES SECTION

```
000002                  BIT1==  BIT01
000001                  BIT0==  BIT00
                        ;
                        ; EVENT FLAG DEFINITIONS
                        ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                        ;
                                                        ;  BIT POSITION IN SECOND STATUS WORD
000040                  EF.START==      32.             ; (100000) START COMMAND WAS ISSUED
000037                  EF.RESTART==    31.             ; (040000) RESTART COMMAND WAS ISSUED
000036                  EF.CONTINUE==   30.             ; (020000) CONTINUE COMMAND WAS ISSUED
000035                  EF.NEW==        29.             ; (010000) A NEW PASS HAS BEEN STARTED
000034                  EF.PWR==        28.             ; (004000) A POWER-FAIL/POWER-UP OCCURRED
                        ;
                        ;
                        ; PRIORITY LEVEL DEFINITIONS
                        ;
000340                  PRI07== 340
000300                  PRI06== 300
000240                  PRI05== 240
000200                  PRI04== 200
000140                  PRI03== 140
000100                  PRI02== 100
000040                  PRI01== 40
000000                  PRI00== 0
                        ;
                        ;OPERATOR FLAG BITS
                        ;
000004                  EVL==         4
000010                  LOT==        10
000020                  ADR==        20
000040                  IDU==        40
000100                  ISR==       100
000200                  UAM==       200
000400                  BOE==       400
001000                  PNT==      1000
002000                  PRI==      2000
004000                  IXE==      4000
010000                  IBE==     10000
020000                  IER==     20000
040000                  LOE==     40000
100000                  HOE==    100000
```

```
1129                            ;;*****************************************************************
1130                            ;;*****************************************************************
1131                            ;
1132                            ;LUN_BLOCK OFFSETS
1133                            ;       THESE LITERALS ARE USED AS WORD OFFSETS INTO THE LUNBLK, WHICH
1134                            ;       IS POINTED TO THROUGHOUT THE PROGRAM BY R4.
1135                            ;
1136                            ;;*****************************************************************
1137                            ;;*****************************************************************
1138
1139    000000                  TUIP    ==      0               ;TUIP REGISTER ADDRESS
1140    000002                  TUSA    ==      2               ;TUSA REGISTER ADDRESS
1141    000004                  TUVEC   ==      4               ;TU INTERRUPT VECTOR
1142    000006                  MSCPUN  ==      6               ;T/MSCP UNIT NUMBER
1143    000010                  TUIPSV  ==     10               ;SAVE LOCATION FOR IP CONTENTS
1144    000012                  TUSASV  ==     12               ;SAVE LOCATION FOR SA CONTENTS
1145    000014                  LUNFLG  ==     14               ;BIT-SPECIFIC MEANINGS AS DEFINED BELOW
1146
1147
1148                            ;;*****************************************************************
1149                            ;;*****************************************************************
1150                            ;
1151                            ;LUNFLG
1152                            ;       THIS WORD IN LUNBLK IS USED TO CONVEY VARIOUS INFORMATION
1153                            ;       IN A BIT-SPECIFIC MANNER.  BITS USED BY THE PROGRAM ARE
1154                            ;       DEFINED AS FOLLOWS.
1155                            ;
1156                            ;;*****************************************************************
1157                            ;;*****************************************************************
1158
1159    000001                  DRPFLG  ==      BIT0            ;=0 UUT AVAILABLE FOR TEST
1160                                                            ;=1 UUT HAS BEEN DROPPED
1161    000002                  INTFLG  ==      BIT1            ;=1 EXPECTED INTERRUPT OCCURRED
1162
1163    000004                  BRFLAG  ==      BIT2            ;=1 INTERRUPT PRIORITY TEST
1164
1165    000010                  TEST.9  ==      BIT3            ;=1 TEST 9 FLAG
1166
1167    000020                  DONEFL  ==      BIT4            ;=1 INTERNAL DRIVE TEST DONE
1168
```

```
1170                            ;;*****************************************************************
1171                            ;;*****************************************************************
1172                            ;
1173                            ;UQ-PORT EQUATES
1174                            ;       THIS SECTION DEFINES THOSE LITERALS USED
1175                            ;       BY THE DIAGNOSTIC IN THE UQ-PORT PROTOCOL.
1176                            ;       IN GENERAL THEY HAVE BEEN FORMED BY USING
1177                            ;       THE TWO LETTER MNEMONIC DEFINED IN UQSSP,
1178                            ;       PRECEDED BY "B." INDICATING THEY ARE BITS.
1179                            ;
1180                            ;;*****************************************************************
1181                            ;;*****************************************************************
1182
1183                            ;READ-ONLY BITS
1184
1185     004000                 B.S1    ==      BIT11   ;STEP 1
1186     010000                 B.S2    ==      BIT12   ;STEP 2
1187     020000                 B.S3    ==      BIT13   ;STEP 3
1188     040000                 B.S4    ==      BIT14   ;STEP 4
1189
1190     100000                 B.ER    ==      BIT15   ;ERROR INDICATION
1191     002000                 B.NV    ==      BIT10   ;=0 VECTOR IS HOST SETTABLE
1192     001000                 B.QB    ==      BIT9    ;=1 SUPPORTS 22 BIT HOST BUS
1193     000400                 B.DI    ==      BIT8    ;=1 SUPPORTS ENHANCED DIAGNOSTICS
1194     000200                 B.OD    ==      BIT7    ;=1 SUPPORTS ODD BUFFER ADDRESSES
1195     000100                 B.MP    ==      BIT6    ;=1 SUPPORTS ADDRESS MAPPING
1196
1197                            ;WRITE-ONLY BITS
1198
1199     100000                 B.PP    ==      BIT15   ;PERFORM PURGE AND POLL TESTS
1200     040000                 B.WR    ==      BIT14   ;ENTER DIAGNOSTIC WRAP MODE
1201     000002                 B.LF    ==      BIT1    ;LAST FAIL REQUEST
1202     000001                 B.PI    ==      BIT0    ;ENABLE ADAPTER PURGE INTERRUPTS
1203     000001                 B.GO    ==      BIT0    ;GO BIT - START RUNNING
1204
1205                            ;READ/WRITE BITS
1206
1207     000200                 B.IE    ==      BIT7    ;STEP X-TION INTERRUPT ENABLE
1208
```

```
1210                                    ;;*****************************************************************
1211                                    ;;*****************************************************************
1212                                    ;
1213                                    ;GENERAL PURPOSE EQUATES
1214                                    ;
1215                                    ;;*****************************************************************
1216                                    ;;*****************************************************************
1217
1218            000004                  VEC4    ==      4       ;VECTOR FOUR - NXM TIMEOUTS, ETC.
1219            000003                  CNTRLC  ==      3       ;CONTROL C (ASCII)
1220            000014                  DISCAC  ==      14      ;BIT POSITIONS 2 AND 3 DISABLE CACHE IN CCR
1221            177560                  RCSR    ==      177560  ;TERMINAL RECEIVE CONTROL/STATUS REGISTER ADDRESS
1222            177562                  RBUF    ==      177562  ;TERMINAL RECEIVE BUFFER ADDRESS
1223            177746                  CCR     ==      177746  ;CACHE CONTROL REGISTER ADDRESS
1224
```

```
1226                            ;;*****************************************************************
1227                            ;;*****************************************************************
1228                            ;
1229                            ;MEMORY MANAGEMENT EQUATES
1230                            ;
1231                            ;;*****************************************************************
1232                            ;;*****************************************************************
1233
1234        177572              MMUSR0  ==      177572  ;STATUS REG 0
1235        177574              MMUSR1  ==      177574
1236        177576              MMUSR2  ==      177576
1237        172516              MMUSR3  ==      172516  ;SHOULD ONLY BE PRESENT ON 22 BIT CPU'S
1238
1239        172340              KPAR0   ==      172340  ;KERNEL MODE PAGE ADDRESS REG 0
1240        172342              KPAR1   ==      172342
1241        172344              KPAR2   ==      172344
1242        172346              KPAR3   ==      172346
1243        172350              KPAR4   ==      172350
1244        172352              KPAR5   ==      172352
1245        172354              KPAR6   ==      172354
1246        172356              KPAR7   ==      172356  ;ALWAYS FOR I/O PAGE
1247
1248        172300              KPDR0   ==      172300  ;KERNEL MODE PAGE DESCRIPTOR REG 0
1249        172302              KPDR1   ==      172302
1250        172304              KPDR2   ==      172304
1251        172306              KPDR3   ==      172306
1252        172310              KPDR4   ==      172310
1253        172312              KPDR5   ==      172312
1254        172314              KPDR6   ==      172314
1255        172316              KPDR7   ==      172316
1256
1257        000001              MMON    ==      BIT0    ;ENABLE MMU - MMUSR0
1258        000020              MM22ON  ==      BIT4    ;ENABLE 22 BIT MMU - MMUSR3
1259
```

```
1261                            ;;*****************************************************************
1262                            ;;*****************************************************************
1263                            ;
1264                            ;COMMAND PACKET OPCODES
1265                            ;
1266                            ;;*****************************************************************
1267                            ;;*****************************************************************
1268
1269    000001                  OP.GDS  ==      01              ;GET DUST STATUS OPCODE
1270    000003                  OP.ELP  ==      03              ;EXECUTE LOCAL PROGRAM OPCODE
1271    000005                  OP.REC  ==      05              ;RECEIVE DATA OPCODE
1272    000006                  OP.ABT  ==      06              ;ABORT PROGRAM OPCODE
1273    000200                  OP.END  ==      200             ;END MESSAGE FLAG OPCODE
1274
1275
1276                            ;;*****************************************************************
1277                            ;;*****************************************************************
1278                            ;
1279                            ;DUP COMMAND AND END MESSAGE OFFSETS
1280                            ;
1281                            ;;*****************************************************************
1282                            ;;*****************************************************************
1283
1284    000000                  P.CRF   ==       0              ;COMMAND REFERENCE NUMBER
1285    000010                  P.OPCD  ==      10              ;COMMAND OPCODE
1286    000012                  P.MOD   ==      12              ;COMMAND MODIFIERS
1287    000014                  P.BCNT  ==      14              ;BYTE COUNT
1288    000020                  P.BUFF  ==      20              ;BUFFER DESCRIPTOR
1289    000010                  P.ENDC  ==      10              ;END MESSAGE ENDCODE
1290    000012                  P.STS   ==      12              ;END MESSAGE STATUS
1291    000017                  P.FLGS  ==      17              ;END MESSAGE FLAGS
1292    000020                  P.IND1  ==      20              ;1ST WORD OF PROGRESS INDICATOR
1293    000022                  P.IND2  ==      22              ;2ND WORD OF PROGRESS INDICATOR
1294    000024                  P.TIMO  ==      24              ;TIMEOUT VALUE
1295
```

```
1297                            ;;*****************************************************************
1298                            ;;*****************************************************************
1299                            ;
1300                            ;TUSA 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 INTERUPT 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                  MIADDR  ==      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
```

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

002234                          LUNBLK::        .BLKW   15.


                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;
                                ;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.
                                ;
                                ;;**********************************************************************
                                ;;**********************************************************************

002272                          STPTBL::        .BLKW   4       ;VALUES WRITTEN TO THE PORT

002302                          CMPTBL::        .BLKW   4       ;COMPARISON VALUES
```

GLOBAL DATA SECTION

```
1393                                    ;;*****************************************************************
1394                                    ;;*****************************************************************
1395                                    ;
1396                                    ;PROGRAM CONTROL VARIABLES
1397                                    ;       THESE GLOBAL VARIABLES ARE GENERALLY USED TO CONTROL THE
1398                                    ;       OVERALL EXECUTION OF THE DIAGNOSTIC.
1399                                    ;
1400                                    ;;*****************************************************************
1401                                    ;;*****************************************************************
1402
1403 002312  000000                     PASCNT::        .WORD   0       ;CUMULATIVE PROGRAM PASS COUNTER
1404 002314  000000                     KTFLAG::        .WORD   0       ;=0 MEMORY MANAGEMENT NOT AVAILABLE
1405                                                                     ;=1 MEMORY MANAGEMENT IS AVAILABLE
1406 002316  000000                     TRP4FG::        .WORD   0       ;=1 TRAP TO VECTOR OCCURRED
1407 002320  000000                     PAROFF::        .WORD   0       ;USED IN TEST 7 TO STEP THROUGH UPPER MEMORY
1408 002322  000000                     CMMERR::        .WORD   0       ;=0 NO ERROR IN COMMUNICATION AREA
1409                                                                     ;=1 ERROR WITHIN COMMUNICATION AREA
1410                                                                     ;=-1 ERROR BEYOND BOUNDS OF COMM AREA
1411 002324  000000                     CMTBLG::        .WORD   0       ;# OF CONTIGUOUS WORDS IN ERROR IN COMM AREA
1412 002326  000000                     CMARLG::        .WORD   0       ;LENGTH OF COMM AREA FOR TEST N
1413 002330  000000                     FRUIS::         .WORD   0       ;POINTER TO FAULTY FRU ASCII FOR PRINTOUT
1414 002332  000000                     LOGUNT::        .WORD   0       ;LOGICAL UNIT # OF CURRENT UUT
1415 002334  000000                     SAEXP::         .WORD   0       ;LOADED WITH EXPECTED SA FOR ERROR CHECKING
1416 002336  000000                     INISTP::        .WORD   0       ;CURRENT STEP OF INIT SEQUENCE
1417 002340  000000                     STEPST::        .WORD   0       ;SUCCESS/FAIL STATUS FROM STEP SUBROUTINES
1418 002342  000000                     WRDATA::        .WORD   0       ;LOADED WITH DATA FRO WRAP MODE TEST
1419 002344  000000                     INNER::         .WORD   0       ;COUNTER FOR PDELAY ROUTINE
1420 002346  000000                     OUTER::         .WORD   0       ;OTHER COUNTER FOR PDELAY
1421 002350  000000                     TOUT::          .WORD   0       ;TIMEOUT INDICATOR FOR PDELAY
1422 002352  000000                     TEMP::          .WORD   0       ;TEMPORARY STORAGE LOCATION
1423 002354  000000                     ANSWER::        .WORD   0       ;LOGICAL ANSWER IN MANUAL TEST SECTION
1424 002356  000000                     PROGRL::        .WORD   0       ;SAVE LOCATION FOR 1ST WORD OF PROGRESS INDICATOR
1425 002360  000000                     PROGRH::        .WORD   0       ;SAVE LOCATION FOR 2ND WORD OF PROGRESS INDICATOR
1426 002362  000000                     CPFLAG::        .WORD   0       ;CACHE PRESENT FLAG
1427
1428
```

```
1430                                    ;;**********************************************************************
1431                                    ;;**********************************************************************
1432                                    ;
1433                                    ;DUP COMMAND PACKETS
1434                                    ;
1435                                    ;;**********************************************************************
1436                                    ;;**********************************************************************
1437
1438                                    ;;**********************************************************************
1439                                    ;
1440                                    ;GET DUST STATUS COMMAND PACKET
1441                                    ;
1442                                    ;;**********************************************************************
1443
1444 002364  000020                             .WORD   16.                     ;PACKET LENGTH IN BYTES
1445 002366     020                             .BYTE   20                      ;MSGTYP = 1 (DATAGRAM); CREDITS = 0
1446 002367     002                             .BYTE   2                       ;CONNECTION ID = 2 (DUP)
1447 002370  000001  000000            GDUST:   .WORD   1,0                     ;COMMAND REFERENCE NUMBER = 1
1448 002374  000000  000000                     .WORD   0,0
1449 002400  C00001  000000                     .WORD   OP.GDS,0                ;OPCODE = 1 (GET DUST STATUS)
1450
1451
1452                                    ;;**********************************************************************
1453                                    ;
1454                                    ;EXECUTE LOCAL PROGRAM COMMAND PACKET
1455                                    ;
1456                                    ;;**********************************************************************
1457
1458 002404  000022                             .WORD   18.                     ;PACKET LENGTH IN BYTES
1459 002406     020                             .BYTE   20                      ;MSGTYP = 1 (DATAGRAM); CREDITS = 0
1460 002407     002                             .BYTE   2                       ;CONNECTION ID = 2 (DUP)
1461 002410  000002  000000            EXELOC:  .WORD   2,0                     ;COMMAND REFERENCE NUMBER = 2
1462 002414  000000  000000                     .WORD   0,0
1463 002420  000003  000001                     .WORD   OP.ELP,1                ;OPCODE = 3 (EXECUTE LOCAL PROGRAM)
1464 002424     040     040     040    TSTNAM:  .ASCII  /      /                ;LOCAL PROGRAM NAME (FILLED AT TEST)
1465
1466
1467                                    ;;**********************************************************************
1468                                    ;
1469                                    ;RECEIVE DATA COMMAND PACKET
1470                                    ;
1471                                    ;;**********************************************************************
1472
1473 002432  000024                             .WORD   20.                     ;PACKET LENGTH IN BYTES
1474 002434     000                             .BYTE   0                       ;MSGTYP = 0 (SEQUENTIAL); CREDITS = 0
1475 002435     002                             .BYTE   2                       ;CONNECTION ID = 2 (DUP)
1476 002436  000003  000000            RCVDAT:  .WORD   3,0                     ;COMMAND REFERENCE NUMBER = 3
1477 002442  000000  000000                     .WORD   0,0
1478 002446  000005  000000                     .WORD   OP.REC,0                ;OPCODE = 5 (RECEIVE DATA)
1479 002452  000156  000000                     .WORD   110.,0                  ;BUFFER SIZE IN BYTES
1480 002456  060C00  000000                     .WORD   RDBUF,0                 ;BUFFER ADDRESS
1481
```

```
1483                                    ;;**********************************************************************
1484                                    ;
1485                                    ;ABORT COMMAND PACKET
1486                                    ;
1487                                    ;;**********************************************************************
1488
1489 002462  000014                             .WORD   12.                     ;PACKET LENGTH IN BYTES
1490 002464     020                             .BYTE   20                      ;MSGTYP = 1 (DATAGRAM); CREDITS = 0
1491 002465     002                             .BYTE   2                       ;CONNECTION ID = 2 (DUP)
1492 002466  000004  000000            ABORT:   .WORD   4,0                     ;COMMAND REFERENCE NUMBER = 4
1493 002472  000000  000000                     .WORD   0,0
1494 002476  000006  000000                     .WORD   OP.ABT,0                ;OPCODE = 6 (ABORT)
1495
```

```
1497                                    ;;**********************************************************************
1498                                    ;;**********************************************************************
1499                                    ;
1500                                    ;CLASS DRIVER BUFFERS
1501                                    ;
1502                                    ;;**********************************************************************
1503                                    ;;**********************************************************************
1504
1505 002502                            RESPBF::        .BLKW   2.      ;TOP 4 LOCATIONS OF RESPONSE BUFFER
1506 002506                            RSPBUF::        .BLKW   66.     ;DRIVER RESPONSE BUFFER
1507
1508
1509                                    ;;**********************************************************************
1510                                    ;;**********************************************************************
1511                                    ;
1512                                    ;U/Q PORT DESCRIPTOR RINGS
1513                                    ;
1514                                    ;;**********************************************************************
1515                                    ;;**********************************************************************
1516
1517 002712                            DSCRNG::        .BLKW   2.      ;DESCRIPTOR RING
1518 002716                            RSPEND::                        ;END OF RESPONSE BUFFER
1519 002716                            RSPRNG::        .BLKW   4.      ;RESPONSE DESCRIPTOR RING
1520 002726                            CMDRNG::        .BLKW   4.      ;COMMAND DESCRIPTOR RING
1521 002736                            DSCEND::                        ;END OF DESCRIPTOR RING
1522
1523
1524                                    ;;**********************************************************************
1525                                    ;;**********************************************************************
1526                                    ;
1527                                    ;CLASS AND PORT DRIVER VARIABLES
1528                                    ;
1529                                    ;;**********************************************************************
1530                                    ;;**********************************************************************
1531
1532 002736  000000                     CNTHI::         .WORD   0       ;VALUE OF THE HIGH TIMEOUT
1533 002740  000000                     CNTFLG::        .WORD   0       ;CONTROLLER FLAGS
1534 002742  000000                     PCKSIZ::        .WORD   0       ;PACKET SIZE IN BYTES
1535 002744  000000                     CMDREF::        .WORD   0       ;COMMAND REFERENCE NUMBER
1536 002746  000000                     CMDCNT::        .WORD   0       ;COMMAND COUNT
1537 002750                            WRBUF::         .BLKW   4096.   ;WRITE BUFFER
1538 022750  000000                     CMDSAV::        .WORD   0       ;COMMAND DESCRIPTOR SAVE
1539 022752  000000                     RSPSAV::        .WORD   0       ;RESPONSE DESCRIPTOR SAVE
1540
1541
1542                                    ;;**********************************************************************
1543                                    ;;**********************************************************************
1544                                    ;
1545                                    ;MANUAL INTERVENTION INPUT DATA TABLE
1546                                    ;
1547                                    ;;**********************************************************************
1548                                    ;;**********************************************************************
1549
1550 022754                            MANTBL::        .BLKB   3       ;TWO BYTES OF INPUT, 3RD BYTE ZERO
1551                                                    .EVEN
```

```
 1553                                   ;;**********************************************************************
 1554                                   ;;**********************************************************************
 1555                                   ;
 1556                                   ;PROTECTION TABLE
 1557                                   ;
 1558                                   ;;**********************************************************************
 1559                                   ;;**********************************************************************
 1563
 1564 022760                            BGNPROT
      022760                            L$PROT::
 1565 022760  000000                               .WORD   0
 1566 022762  177777                               .WORD   -1
 1567 022764  177777                               .WORD   -1
 1568
 1569 022766                            ENDPROT
 1570
```

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

022766                                  DEVTYP  <TU81>
022766                          L$DVTYP::
022766  124     125     070             .ASCIZ  %TU81%
                                        .EVEN
```

```
1602
1603                                    ;;*****************************************************************
1604                                    ;
1605                                    ;FORMAT STATEMENTS
1606                                    ;
1607                                    ;;*****************************************************************
1608
1609 022774    045     101     111    LINE1:: .ASCIZ  ?%AINIT SEQUENCE STEP #: %D1?
1610 023030    045     116     045    LINE2:: .ASCIZ  ?%N%ASA REG: %O6%A EXPCTD: %O6%A  ACTUAL SA: %O6?
1611 023110    045     116     045    LINE3:: .ASCIZ  ?%N%AIP REG ADDRESS: %O6?
1612 023140    045     116     062    LINE4:: .ASCIZ  ?%N2%A****FAILING FRU: %T%A****%N%N?
1613 023203    045     101     122    LINE5:: .ASCIZ  ?%ARELOCATION CONSTANT: %O6%A  VIRT. ADD: %O6?
1614 023260    045     116     045    LINE6:: .ASCIZ  ?%N%AEXPECTED: %O6%A  RECEIVED: %O6?
1615 023323    045     101     120    LINE7:: .ASCIZ  ?%APHYSICAL ADD: %O6?
1616                                            .EVEN
1617
1618 023350    045     116     045    WR1::   .ASCIZ  ?%N%ASA REG: %O6%A SA CONTENTS: %O6?
1619                                            .EVEN
1620
1621 023414    045     116     062    PKSENT::.ASCIZ  ?%N2%APACKET SENT:?
1622 023436    045     116     045    CREFNO::.ASCIZ  ?%N%ACOMMAND REFERENCE NUMBER: %O6?
1623 023500    045     116     045    OPCODE::.ASCIZ  ?%N%AOPCODE: %O3?
1624 023520    045     116     045    MODIFY::.ASCIZ  ?%N%AMODIFIERS: %O6?
1625 023543    045     116     045    PRGNAM::.ASCIZ  ?%N%APROGRAM NAME: %O3%A %O3%A %O3%A %O3%A %O3%A %O3?
1626 023627    045     116     045    BYTCNT::.ASCIZ  ?%N%ABYTE COUNT: %O6?
1627 023653    045     116     045    BUFDES::.ASCIZ  ?%N%ABUFFER DESCRIPTOR: %O6?
1628 023706    045     116     062    PKRECV::.ASCIZ  ?%N2%APACKET RECEIVED:?
1629 023734    045     116     045    ENCODE::.ASCIZ  ?%N%AENDCODE: %O3?
1630 023755    045     116     045    STATUS::.ASCIZ  ?%N%ASTATUS: %O6?
1631 023775    045     116     045    PRGVER::.ASCIZ  ?%N%APROGRAM VERSION: %O6?
1632 024026    045     116     045    TIMOUT::.ASCIZ  ?%N%ATIMEOUT: %O3?
1633 024047    045     116     045    FLAGS:: .ASCIZ  ?%N%AFLAGS: %O3?
1634 024066    045     116     045    FAULTC::.ASCIZ  ?%N%AFAULT CODE:       SUB-FAULT CODE:    ?
1635                                            .EVEN
1636
1637                                    ;;*****************************************************************
1638                                    ;
1639                                    ;ERROR MESSAGES
1640                                    ;
1641                                    ;;*****************************************************************
1642
1643
1644 024140    116     130     115    EMSG5:: .ASCIZ  ?NXM ON READ TUIP?
1645 024161    124     125     111    EMSG6:: .ASCIZ  ?TUIP NOT 0 ON FIRST READ?
1646 024212    116     130     115    EMSG7:: .ASCIZ  ?NXM ON READ TUSA?
1647 024233    123     101     040    EMSG8:: .ASCIZ  ?SA REG IN ERROR ON FIRST READ?
1648 024271    123     101     040    EMSG9:: .ASCIZ  ?SA CONTENTS IN ERROR?
1649 024316    123     101     040    EMSG10::.ASCIZ  ?SA WRONG IN DATA WRAP?
1650 024344    105     130     120    EMSG11::.ASCIZ  ?EXPECTED INTERRUPT DID NOT OCCUR?
1651 024405    111     116     124    EMSG12::.ASCIZ  ?INTRRPT OCCURRED WITH CPU PRIORITY = 7?
1652 024454    123     101     040    EMSG13::.ASCIZ  ?SA NOT 0 IN PURGE/POLL?
1653 024503    120     125     122    EMSG14::.ASCIZ  ?PURGE/POLL TEST FAILED?
1654 024532    105     130     124    EMSG15::.ASCIZ  ?EXTENDED ADDRESS TEST FAILED?
1655 024567    042     105     130    EMSG16::.ASCIZ  ?"EXECUTE LOCAL PROGRAM" COMMAND TIMEOUT?
1656 024637    042     107     105    EMSG17::.ASCIZ  ?"GET DUST STATUS" COMMAND TIMEOUT?
1657 024701    042     107     105    EMSG18::.ASCIZ  ?"GET DUST STATUS" COMMAND FAILURE?
1658 024743    042     105     130    EMSG19::.ASCIZ  ?"EXECUTE LOCAL PROGRAM" COMMAND FAILURE?
```

GLOBAL TEXT SECTION

```
1659 025013 042 122 105 EMSG20::.ASCIZ  ?"RECEIVE DATA" COMMAND FAILURE?
1660 025052 101 102 117 EMSG21::.ASCIZ  ?ABORT COMMANDS DON'T WORK?
1661 025104 111 116 124 EMSG22::.ASCIZ  ?INTERNAL DRIVE TEST HUNG?
1662 025135 111 116 126 EMSG23::.ASCIZ  ?INVALID MESSAGE NUMBER FROM INTERNAL DRIVE TEST?
1663 025215 111 116 124 EMSG24::.ASCIZ  ?INTERNAL DRIVE TEST FAILED?
1664                                    .EVEN
1665
1666 025250 124 111 115 WRER1:: .ASCIZ  ?TIME OUT DURING PORT INIT?
1667 025302 120 117 122 WRER2:: .ASCIZ  ?PORT INIT FAILED?
1668 025323 124 115 123 WRER3:: .ASCIZ  ?TMSCP COMMAND FAILURE?
1669 025351 120 117 122 WRER4:: .ASCIZ  ?PORT DETECTED ERROR?
1670 025375 111 116 103 WRER5:: .ASCIZ  ?INCORRECT COMMAND REFERENCE NUMBER RECEIVED.?
1671 025452 045 116 045 WRER6:: .ASCIZ  ?%N%AREFER TO PATHFINDER FOR EXPLANATION OF CODES.?
1672 025534 045 116 045 WRER7:: .ASCIZ  ?%N%ARECEIVED INVALID MESSAGE NUMBER FROM INTERNAL DRIVE TEST.?
1673                                    .EVEN
1674
1675                                    ;;*****************************************************************
1676                                    ;
1677                                    ;MISCELLANEOUS ERROR MESSAGES
1678                                    ;
1679                                    ;;*****************************************************************
1680
1681 025632 114 105 123 LESI::   .ASCIZ  ?LESI ADAPTER?
1682 025647 103 117 116 CTRL::   .ASCIZ  ?CONTROLLER/CABLE?
1683 025670 114 105 123 LSCT::   .ASCIZ  ?LESI/CONTROLLER/CABLE?
1684 025716 104 122 111 DRVE::   .ASCIZ  ?DRIVE?
1685                                    .EVEN
1686
1687                                    ;;*****************************************************************
1688                                    ;
1689                                    ;MANUAL TEST MESSAGES
1690                                    ;
1691                                    ;;*****************************************************************
1692
1693 025724 045 116 045 T10MS1::.ASCIZ  \%N%ATest 10: FUNCTIONAL FAULT DETECTION TEST (Drive Resident Test #1)\
1694 026032 045 116 062 T10MS2::.ASCIZ  \%N2%A*** CAUTION ***\
1695 026057 045 116 045 T10MS3::.ASCIZ  \%N%AThis test will destroy the data on tape.\
1696 026134 045 116 045 T10MS4::.ASCIZ  \%N%AMount a scratch tape UNTENSIONED but THREADED.%N\
1697 026221 045 116 045 T11MS1::.ASCIZ  \%N%ATest 11: TENSION FAULT ISOLATION TEST (Drive Resident Test #2)\
1698 026324 045 116 045 T12MS1::.ASCIZ  \%N%ATest 12: VELOCITY FAULT ISOLATION TEST (Drive Resident Test #3)\
1699 026430 045 116 045 T13MS1::.ASCIZ  \%N%ATest 13: SELECT A DRIVE RESIDENT TEST (Drive Resident Tests 1-99)\
1700 026536 045 116 062 MMSG::.ASCIZ    \%N2%A*** REFER TO PATHFINDER FOR TEST REQUIREMENTS BEFORE PROCEDING ***\
1701 026646 105 156 164 SELTST::.ASCIZ  \Enter drive resident test number (1-99)\
1702 026716 111 163 040 QUESTN::.ASCIZ  \Is the drive ready (To bypass this test hit return)\
1703                                    .EVEN
```

```
 1705                                   .SBTTL  GLOBAL ERROR REPORT SECTION
 1709                                   ;;*****************************************************************
 1710                                   ;;*****************************************************************
 1711                                   ;
 1712                                   ;GLOBAL ERROR REPORTS
 1713                                   ;       THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB
 1714                                   ;       AND PRINTX CALLS THAT ARE USED IN MORE THAN ONE TEST.
 1715                                   ;       IT ALSO INCLUDES THE ASCII MESSAGES THAT ARE USED BY
 1716                                   ;       THE PRINTB AND PRINTX CALLS.
 1717                                   ;
 1718                                   ;;*****************************************************************
 1719                                   ;;*****************************************************************
 1723
 1724 027002                           BGNMSG
 1725
 1726 027002                           PRIINI::
 1727 027002                                   PRINTX  #LINE1,INISTP
      027002  013746  002336                   MOV     INISTP,-(SP)
      027006  012746  022774                   MOV     #LINE1,-(SP)
      027012  012746  000002                   MOV     #2,-(SP)
      027016  010600                           MOV     SP,R0
      027020  104415                           TRAP    C$PNTX
      027022  062706  000006                   ADD     #6,SP
 1728
 1729 027026                           PRISA::
 1730 027026                                   PRINTX  #LINE2,TUSA(R4),SAEXP,TUSASV(R4)
      027026  016446  000012                   MOV     TUSASV(R4),-(SP)
      027032  013746  002334                   MOV     SAEXP,-(SP)
      027036  016446  000002                   MOV     TUSA(R4),-(SP)
      027042  012746  023030                   MOV     #LINE2,-(SP)
      027046  012746  000004                   MOV     #4,-(SP)
      027052  010600                           MOV     SP,R0
      027054  104415                           TRAP    C$PNTX
      027056  062706  000012                   ADD     #12,SP
 1731 027062  000137  030624                   JMP     FRUERR
 1732
 1733 027066                           PRIPAD::
 1734 027066                                   PRINTX  #LINE7,R2
      027066  010246                           MOV     R2,-(SP)
      027070  012746  023323                   MOV     #LINE7,-(SP)
      027074  012746  000002                   MOV     #2,-(SP)
      027100  010600                           MOV     SP,R0
      027102  104415                           TRAP    C$PNTX
      027104  062706  000006                   ADD     #6,SP
 1735 027110  000137  027142                   JMP     PRIDAT
 1736
 1737 027114                           PRIVAD::
 1738 027114                                   PRINTX  #LINE5,KPAR3,R2
      027114  010246                           MOV     R2,-(SP)
      027116  013746  172346                   MOV     KPAR3,-(SP)
      027122  012746  023203                   MOV     #LINE5,-(SP)
      027126  012746  000003                   MOV     #3,-(SP)
      027132  010600                           MOV     SP,R0
      027134  104415                           TRAP    C$PNTX
      027136  062706  000010                   ADD     #10,SP
 1739
 1740 027142                           PRIDAT::
```

```
 1741 027142                                    PRINTX  #LINE6,R1,(R2)
      027142  011246                           MOV     (R2),-(SP)
      027144  010146                           MOV     R1,-(SP)
      027146  012746  023260                   MOV     #LINE6,-(SP)
      027152  012746  000003                   MOV     #3,-(SP)
      027156  010600                           MOV     SP,R0
      027160  104415                           TRAP    C$PNTX
      027162  062706  000010                   ADD     #10,SP
 1742 027166  000137  030624                   JMP     FRUERR
 1743
 1744 027172                                PRIIP::
 1745 027172                                    PRINTX  #LINE3,TUIP(R4)
      027172  016446  000000                   MOV     TUIP(R4),-(SP)
      027176  012746  023110                   MOV     #LINE3,-(SP)
      027202  012746  000002                   MOV     #2,-(SP)
      027206  010600                           MOV     SP,R0
      027210  104415                           TRAP    C$PNTX
      027212  062706  000006                   ADD     #6,SP
 1746 027216  000137  030624                   JMP     FRUERR
 1747
 1748 027222                                PRIERR::
 1749 027222  000137  030624                   JMP     FRUERR
 1750
 1751
 1752 027226                                WRINTO::
 1753 027226                                    PRINTX  #LINE1,INISTP
      027226  013746  002336                   MOV     INISTP,-(SP)
      027232  012746  022774                   MOV     #LINE1,-(SP)
      027236  012746  000002                   MOV     #2,-(SP)
      027242  010600                           MOV     SP,R0
      027244  104415                           TRAP    C$PNTX
      027246  062706  000006                   ADD     #6,SP
 1754
 1755 027252                                WRPRTE::
 1756 027252                                    PRINTX  #WR1,TUSA(R4),TUSASV(R4)
      027252  016446  000012                   MOV     TUSASV(R4),-(SP)
      027256  016446  000002                   MOV     TUSA(R4),-(SP)
      027262  012746  023350                   MOV     #WR1,-(SP)
      027266  012746  000003                   MOV     #3,-(SP)
      027272  010600                           MOV     SP,R0
      027274  104415                           TRAP    C$PNTX
      027276  062706  000010                   ADD     #10,SP
 1757 027302  000137  030624                   JMP     FRUERR
 1758
 1759 027306                                ELPERR::                        ;COMMAND/RESPONSE PACKET PRINTOUT
 1760 027306                                    PRINTB  #PKSENT
      027306  012746  023414                   MOV     #PKSENT,-(SP)
      027312  012746  000001                   MOV     #1,-(SP)
      027316  010600                           MOV     SP,R0
      027320  104414                           TRAP    C$PNTB
      027322  062706  000004                   ADD     #4,SP
 1761 027326                                    PRINTB  #CREFNO,(R5)
      027326  011546                           MOV     (R5),-(SP)
      027330  012746  023436                   MOV     #CREFNO,-(SP)
      027334  012746  000002                   MOV     #2,-(SP)
      027340  010600                           MOV     SP,R0
      027342  104414                           TRAP    C$PNTB
```

```
      027344  062706  000006                  ADD     #6,SP
1762  027350                                  PRINTB  #OPCODE,<B,10(R5)>
      027350  005046                          CLR     -(SP)
      027352  156516  000010                  BISB    10(R5),(SP)
      027356  012746  023500                  MOV     #OPCODE,-(SP)
      027362  012746  000002                  MOV     #2,-(SP)
      027366  010600                          MOV     SP,R0
      027370  104414                          TRAP    C$PNTB
      027372  062706  000006                  ADD     #6,SP
1763  027376                                  PRINTB  #MODIFY,12(R5)
      027376  016546  000012                  MOV     12(R5),-(SP)
      027402  012746  023520                  MOV     #MODIFY,-(SP)
      027406  012746  000002                  MOV     #2,-(SP)
      027412  010600                          MOV     SP,R0
      027414  104414                          TRAP    C$PNTB
      027416  062706  000006                  ADD     #6,SP
1764  027422                                  PRINTB  #PRGNAM,<B,14(R5)>,<B,15(R5)>,<B,16(R5)>,<B,17(R5)>,<B,20(R5)>,<B,21(R5)>
      027422  005046                          CLR     -(SP)
      027424  156516  000021                  BISB    21(R5),(SP)
      027430  005046                          CLR     -(SP)
      027432  156516  000020                  BISB    20(R5),(SP)
      027436  005046                          CLR     -(SP)
      027440  156516  000017                  BISB    17(R5),(SP)
      027444  005046                          CLR     -(SP)
      027446  156516  000016                  BISB    16(R5),(SP)
      027452  005046                          CLR     -(SP)
      027454  156516  000015                  BISB    15(R5),(SP)
      027460  005046                          CLR     -(SP)
      027462  156516  000014                  BISB    14(R5),(SP)
      027466  012746  023543                  MOV     #PRGNAM,-(SP)
      027472  012746  000007                  MOV     #7,-(SP)
      027476  010600                          MOV     SP,R0
      027500  104414                          TRAP    C$PNTB
      027502  062706  000020                  ADD     #20,SP
1765  027506                                  PRINTB  #PKRECV
      027506  012746  023706                  MOV     #PKRECV,-(SP)
      027512  012746  000001                  MOV     #1,-(SP)
      027516  010600                          MOV     SP,R0
      027520  104414                          TRAP    C$PNTB
      027522  062706  000004                  ADD     #4,SP
1766  027526                                  PRINTB  #CREFNO,(R3)
      027526  011346                          MOV     (R3),-(SP)
      027530  012746  023436                  MOV     #CREFNO,-(SP)
      027534  012746  000002                  MOV     #2,-(SP)
      027540  010600                          MOV     SP,R0
      027542  104414                          TRAP    C$PNTB
      027544  062706  000006                  ADD     #6,SP
1767  027550                                  PRINTB  #ENCODE,<B,10(R3)>
      027550  005046                          CLR     -(SP)
      027552  156316  000010                  BISB    10(R3),(SP)
      027556  012746  023734                  MOV     #ENCODE,-(SP)
      027562  012746  000002                  MOV     #2,-(SP)
      027566  010600                          MOV     SP,R0
      027570  104414                          TRAP    C$PNTB
      027572  062706  000006                  ADD     #6,SP
1768  027576                                  PRINTB  #STATUS,12(R3)
      027576  016346  000012                  MOV     12(R3),-(SP)
```

```
        027602  012746  023755                  MOV     #STATUS,-(SP)
        027606  012746  000002                  MOV     #2,-(SP)
        027612  010600                          MOV     SP,R0
        027614  104414                          TRAP    C$PNTB
        027616  062706  000006                  ADD     #6,SP
  1769  027622                                  PRINTB  #PRGVER,14(R3)
        027622  016346  000014                  MOV     14(R3),-(SP)
        027626  012746  023775                  MOV     #PRGVER,-(SP)
        027632  012746  000002                  MOV     #2,-(SP)
        027636  010600                          MOV     SP,R0
        027640  104414                          TRAP    C$PNTB
        027642  062706  000006                  ADD     #6,SP
  1770  027646                                  PRINTB  #TIMOUT,<B,15(R3)>
        027646  005046                          CLR     -(SP)
        027650  156316  000015                  BISB    15(R3),(SP)
        027654  012746  024026                  MOV     #TIMOUT,-(SP)
        027660  012746  000002                  MOV     #2,-(SP)
        027664  010600                          MOV     SP,R0
        027666  104414                          TRAP    C$PNTB
        027670  062706  000006                  ADD     #6,SP
  1771  027674                                  PRINTB  #FLAGS,<B,16(R3)>
        027674  005046                          CLR     -(SP)
        027676  156316  000016                  BISB    16(R3),(SP)
        027702  012746  024047                  MOV     #FLAGS,-(SP)
        027706  012746  000002                  MOV     #2,-(SP)
        027712  010600                          MOV     SP,R0
        027714  104414                          TRAP    C$PNTB
        027716  062706  000006                  ADD     #6,SP
  1772  027722  000137  030624                  JMP     FRUERR
  1773
  1774  027726                          RCVERR::                        ;COMMAND/RESPONSE PACKET PRINTOUT
  1775  027726                                  PRINTB  #PKSENT
        027726  012746  023414                  MOV     #PKSENT,-(SP)
        027732  012746  000001                  MOV     #1,-(SP)
        027736  010600                          MOV     SP,R0
        027740  104414                          TRAP    C$PNTB
        027742  062706  000004                  ADD     #4,SP
  1776  027746                                  PRINTB  #CREFNO,(R5)
        027746  011546                          MOV     (R5),-(SP)
        027750  012746  023436                  MOV     #CREFNO,-(SP)
        027754  012746  000002                  MOV     #2,-(SP)
        027760  010600                          MOV     SP,R0
        027762  104414                          TRAP    C$PNTB
        027764  062706  000006                  ADD     #6,SP
  1777  027770                                  PRINTB  #OPCODE,<B,10(R5)>
        027770  005046                          CLR     -(SP)
        027772  156516  000010                  BISB    10(R5),(SP)
        027776  012746  023500                  MOV     #OPCODE,-(SP)
        030002  012746  000002                  MOV     #2,-(SP)
        030006  010600                          MOV     SP,R0
        030010  104414                          TRAP    C$PNTB
        030012  062706  000006                  ADD     #6,SP
  1778  030016                                  PRINTB  #MODIFY,12(R5)
        030016  016546  000012                  MOV     12(R5),-(SP)
        030022  012746  023520                  MOV     #MODIFY,-(SP)
        030026  012746  000002                  MOV     #2,-(SP)
        030032  010600                          MOV     SP,R0
```

```
        030034  104414                          TRAP    C$PNTB
        030036  062706  000006                  ADD     #6,SP
   1779 030042                                  PRINTB  #BYTCNT,14(R5)
        030042  016546  000014                  MOV     14(R5),-(SP)
        030046  012746  023627                  MOV     #BYTCNT,-(SP)
        030052  012746  000002                  MOV     #2,-(SP)
        030056  010600                          MOV     SP,R0
        030060  104414                          TRAP    C$PNTB
        030062  062706  000006                  ADD     #6,SP
   1780 030066                                  PRINTB  #BUFDES,20(R5)
        030066  016546  000020                  MOV     20(R5),-(SP)
        030072  012746  023653                  MOV     #BUFDES,-(SP)
        030076  012746  000002                  MOV     #2,-(SP)
        030102  010600                          MOV     SP,R0
        030104  104414                          TRAP    C$PNTB
        030106  062706  000006                  ADD     #6,SP
   1781 030112                                  PRINTB  #PKRECV
        030112  012746  023706                  MOV     #PKRECV,-(SP)
        030116  012746  000001                  MOV     #1,-(SP)
        030122  010600                          MOV     SP,R0
        030124  104414                          TRAP    C$PNTB
        030126  062706  000004                  ADD     #4,SP
   1782 030132                                  PRINTB  #CREFNO,(R3)
        030132  011346                          MOV     (R3),-(SP)
        030134  012746  023436                  MOV     #CREFNO,-(SP)
        030140  012746  000002                  MOV     #2,-(SP)
        030144  010600                          MOV     SP,R0
        030146  104414                          TRAP    C$PNTB
        030150  062706  000006                  ADD     #6,SP
   1783 030154                                  PRINTB  #ENCODE,<B,10(R3)>
        030154  005046                          CLR     -(SP)
        030156  156316  000010                  BISB    10(R3),(SP)
        030162  012746  023734                  MOV     #ENCODE,-(SP)
        030166  012746  000002                  MOV     #2,-(SP)
        030172  010600                          MOV     SP,R0
        030174  104414                          TRAP    C$PNTB
        030176  062706  000006                  ADD     #6,SP
   1784 030202                                  PRINTB  #STATUS,12(R3)
        030202  016346  000012                  MOV     12(R3),-(SP)
        030206  012746  023755                  MOV     #STATUS,-(SP)
        030212  012746  000002                  MOV     #2,-(SP)
        030216  010600                          MOV     SP,R0
        030220  104414                          TRAP    C$PNTB
        030222  062706  000006                  ADD     #6,SP
   1785 030226                                  PRINTB  #BYTCNT,14(R3)
        030226  016346  000014                  MOV     14(R3),-(SP)
        030232  012746  023627                  MOV     #BYTCNT,-(SP)
        030236  012746  000002                  MOV     #2,-(SP)
        030242  010600                          MOV     SP,R0
        030244  104414                          TRAP    C$PNTB
        030246  062706  000006                  ADD     #6,SP
   1786 030252  000137  030624                  JMP     FRUERR
   1787
   1788 030256                          GDSERR::                        ;COMMAND/RESPONSE PACKET PRINTOUT
   1789 030256                                  PRINTB  #PKSENT
        030256  012746  023414                  MOV     #PKSENT,-(SP)
        030262  012746  000001                  MOV     #1,-(SP)
```

```
                030266  010600                  MOV     SP,R0
                030270  104414                  TRAP    C$PNTB
                030272  062706  000004          ADD     #4,SP
          1790  030276                          PRINTB  #CREFNO,(R5)
                030276  01_546                  MOV     (R5),-(SP)
                030300  012746  023436          MOV     #CREFNO,-(SP)
                030304  012746  000002          MOV     #2,-(SP)
                030310  010600                  MOV     SP,R0
                030312  104414                  TRAP    C$PNTB
                030314  062706  000006          ADD     #6,SP
          1791  030320                          PRINTB  #OPCODE,<8,10(R5)>
                030320  005046                  CLR     -(SP)
                030322  156516  000010          BISB    10(R5),(SP)
                030326  012746  023500          MOV     #OPCODE,-(SP)
                030332  012746  000002          MOV     #2,-(SP)
                030336  010600                  MOV     SP,R0
                030340  104414                  TRAP    C$PNTB
                030342  062706  000006          ADD     #6,SP
          1792  030346                          PRINTB  #MODIFY,12(R5)
                030346  C16546  000012          MOV     12(R5),-(SP)
                030352  012746  023520          MOV     #MODIFY,-(SP)
                030356  012746  000002          MOV     #2,-(SP)
                030362  010600                  MOV     SP,R0
                030364  104414                  TRAP    C$PNTB
                030366  062706  000006          ADD     #6,SP
          1793  030372                          PRINTB  #PKRECV
                030372  012746  023706          MOV     #PKRECV,-(SP)
                030376  012746  000001          MOV     #1,-(SP)
                030402  010600                  MOV     SP,R0
                030404  104414                  TRAP    C$PNTB
                030406  062706  000004          ADD     #4,SP
          1794  030412                          PRINTB  #CREFNO,(R3)
                030412  011346                  MOV     (R3),-(SP)
                030414  012746  023436          MOV     #CREFNO,-(SP)
                030420  012746  000002          MOV     #2,-(SP)
                030424  010600                  MOV     SP,R0
                030426  104414                  TRAP    C$PNTB
                030430  062706  000006          ADD     #6,SP
          1795  030434                          PRINTB  #ENCODE,<8,10(R3)>
                030434  005046                  CLR     -(SP)
                030436  156316  000010          BISB    10(R3),(SP)
                030442  012746  023734          MOV     #ENCODE,-(SP)
                030446  012746  000002          MOV     #2,-(SP)
                030452  010600                  MOV     SP,R0
                030454  104414                  TRAP    C$PNTB
                030456  062706  000006          ADD     #6,SP
          1796  030462                          PRINTB  #STATUS,12(R3)
                030462  016346  000012          MOV     12(R3),-(SP)
                030466  012746  023755          MOV     #STATUS,-(SP)
                030472  012746  000002          MOV     #2,-(SP)
                030476  010600                  MOV     SP,R0
                030500  104414                  TRAP    C$PNTB
                030502  062706  000006          ADD     #6,SP
          1797  030506                          PRINTB  #FLAGS,<8,17(R3)>
                030506  005046                  CLR     -(SP)
                030510  156316  000017          BISB    17(R3),(SP)
                030514  012746  024047          MOV     #FLAGS,-(SP)
```

```
                030520  012746  000002                  MOV     #2,-(SP)
                030524  010600                          MOV     SP,R0
                030526  104414                          TRAP    C$PNTB
                030530  062706  000006                  ADD     #6,SP
     1798       030534  000137  030624                  JMP     FRUERR
     1799
     1800       030540                          INTMSG::
     1801       030540                                  PRINTB  #FAULTC
                030540  012746  024066                  MOV     #FAULTC,-(SP)
                030544  012746  000001                  MOV     #1,-(SP)
                030550  010600                          MOV     SP,R0
                030552  104414                          TRAP    C$PNTB
                030554  062706  000004                  ADD     #4,SP
     1802       030560                                  PRINTB  #MRER6
                030560  012746  025452                  MOV     #MRER6,-(SP)
                030564  012746  000001                  MOV     #1,-(SP)
                030570  010600                          MOV     SP,R0
                030572  104414                          TRAP    C$PNTB
                030574  062706  000004                  ADD     #4,SP
     1803       030600  000137  030624                  JMP     FRUERR
     1804
     1805       030604                          INVMSG::
     1806       030604                                  PRINTB  #MRER7
                030604  012746  025534                  MOV     #MRER7,-(SP)
                030610  012746  000001                  MOV     #1,-(SP)
                030614  010600                          MOV     SP,R0
                030616  104414                          TRAP    C$PNTB
                030620  062706  000004                  ADD     #4,SP
     1807
     1808
     1809       030624                          FRUERR::
     1810       030624                                  PRINTB  #LINE4,FRUIS
                030624  013746  002350                  MOV     FRUIS,-(SP)
                030630  012746  023140                  MOV     #LINE4,-(SP)
                030634  012746  000002                  MOV     #2,-(SP)
                030640  010600                          MOV     SP,R0
                030642  104414                          TRAP    C$PNTB
                030644  062706  000006                  ADD     #6,SP
     1811
     1812       030650                          PRIEX:  EXIT    MSG
                030650  000167                          .WORD   J$JMP
                030652  000000                          .WORD   L10003-2-.
     1813
     1814       030654                                  ENDMSG
                030654                          L10003:
                030654  104423                          TRAP    C$MSG
     1815
```

```
 1817                                   .SBTTL  GLOBAL SUBROUTINES SECTION
 1821
 1822                                   ;;*****************************************************************
 1823                                   ;;*****************************************************************
 1824                                   ;
 1825                                   ;GLOBAL SUBROUTINES SECTION
 1826                                   ;       THIS SECTION CONTAINS ALL SUBROUTINES AND
 1827                                   ;       INTERRUPT SERVICE ROUTINES THAT ARE AC-
 1828                                   ;       CESSED FROM ANYWHERE IN THE PROGRAM.
 1829                                   ;
 1830                                   ;;*****************************************************************
 1831                                   ;;*****************************************************************
 1832
 1833
 1834                                   ;;*****************************************************************
 1835                                   ;;*****************************************************************
 1836                                   ;
 1837                                   ;TRAP4
 1838                                   ;       THE ADDRESS OF THIS ROUTINE IS LOADED
 1839                                   ;       INTO VECTOR 4 WHENEVER THE PROGRAM IS
 1840                                   ;       ATTEMPTING TO ACCESS A PIECE OF HARDWARE
 1841                                   ;       FOR THE FIRST TIME.  IT IS INTENDED TO
 1842                                   ;       CATCH NON-EXISTENT MEMORY TIMEOUTS IN
 1843                                   ;       THE EVENT THE HARDWARE IS NOT REALLY PRE-
 1844                                   ;       SENT OR IS MALFUNCTIONING.  IT SIMPLY
 1845                                   ;       SETS A FLAG, INDICATING THE TRAP OCCURRED.
 1846                                   ;
 1847                                   ;;*****************************************************************
 1848                                   ;;*****************************************************************
 1852
 1853 030656                           BGNSRV  TRAP4
      030656                           TRAP4::
 1854
 1855 030656  005237  002316                   INC     TRP4FG          ;SET THE FLAG - TRAP OCCURRED
 1856
 1857 030662                           ENDSRV
      030662                           L10004:
      030662  000002                   RTI
 1858
```

```
1863
1864
1865                                    ;;*****************************************************************
1866                                    ;;*****************************************************************
1867                                    ;
1868                                    ;INTRCV
1869                                    ;       THIS IS THE TU81 INTERRUPT HANDLER USED BY THE PRO-
1870                                    ;       GRAM WHEN INTERRUPTS HAVE BEEN ENABLED.  IF THE
1871                                    ;       BRFALG IS CLEAR, THE ROUTINE SETS A FLAG INDICATING
1872                                    ;       THE EXPECTED INTERRUPT OCCURRED.  IF BRFLAG IS SET,
1873                                    ;       IT INDICATES THAT PROCESOR PRIORITY WAS SET TO A
1874                                    ;       LEVEL THAT SHOULD HAVE INHIBITED THE INTERRUPT, SO
1875                                    ;       THE ROUTINE SETS AN ERROR INDICATOR.
1876                                    ;
1877                                    ;;*****************************************************************
1878                                    ;;*****************************************************************
1879
1880 030664                                     BGNSRV  INTRCV
     030664                             INTRCV::
1884
1885                                    ;       BIT     #BRFLAG,LUNFLG(R4)      ;IF NOT PRIORITY LEVEL TESTING
1886                                    ;       BEQ     5$                      ;  THEN SKIP AROUND
1887                                    ;       MOV     #DRPFLG,LUNFLG(R4)      ;  ELSE SET FAILED BIT
1888                                    ;       BR      EXTINT                  ;RETURN
1889
1890 030664  052764  000002  000014   5$:     BIS     #INTFLG,LUNFLG(R4)      ;SET THE FLAG
1891
1892 030672                             EXTINT:
1893 030672                                     ENDSRV
     030672                             L10005:
     030672  000002                             RTI
1894
```

```
1899
1900                                    ;;**********************************************************************
1901                                    ;;**********************************************************************
1902                                    ;
1903                                    ;ILLINT
1904                                    ;       THIS HANDLER ROUTINE'S ADDRESS IS LOADED INTO THE
1905                                    ;       CURRENT UUT'S VECTOR FOR ALL TESTS THAT DO NOT EN-
1906                                    ;       ABLE DEVICE INTERRUPTS.
1907                                    ;
1908                                    ;;**********************************************************************
1909                                    ;;**********************************************************************
1913
1914 030674                                        BGNSRV  ILLINT
     030674                             ILLINT::
1915
1916 030674  052764  000001  000014               BIS     #DRPFLG,LUNFLG(R4)
1917
1918
1919 030702                                        ENDSRV
     030702                             L10006:
     030702  000002                               RTI
1920
```

```
 1925
 1926                                   ;;**********************************************************************
 1927                                   ;;**********************************************************************
 1928                                   ;
 1929                                   ;CHKCAC
 1930                                   ;       THIS ROUTINE IS USED IN THE DATA WRAP TEST TO CHECK IF
 1931                                   ;       CACHE MEMORY IS PRESENT AND ENABLED ON THE SYSTEM BEING
 1932                                   ;       TESTED.  IF SO, CACHE IS DISABLED BEFORE PROCEEDING
 1933                                   ;       TO PREVENT THE TEST FROM INCORRECTLY REPORTING AN ERROR.
 1934                                   ;
 1935                                   ;;**********************************************************************
 1936                                   ;;**********************************************************************
 1940
 1941 030704                            CHKCAC::
 1942 030704                                    SETVEC  #VEC4,#TRAP4,#PRIO7     ;SET UP FOR POSSIBLE ILLEGAL INT
      030704  012746  000000G                   MOV     #PRIO7,-(SP)
      030710  012746  030656                    MOV     #TRAP4, (SP)
      030714  012746  000004                    MOV     #VEC4,-(SP)
      030720  012746  000003                    MOV     #3,-(SP)
      030724  104437                            TRAP    C$SVEC
      030726  062706  000010                    ADD     #10,SP
 1943 030732  005037  002362                    CLR     CPFLAG                  ;CLEAR "CACHE PRESENT" FLAG
 1944 030736  005737  177746                    TST     CCR                     ;READ CACHE CONTROL REGISTER
 1945 030742  005737  002316                    TST     TRP4FG                  ;CACHE PRESENT ?
 1946 030746  001005                            BNE     10$                     ;NO, BRANCH
 1947 030750  052737  000014  177746           BIS     #DISCAC,CCR             ;DISABLE CACHE
 1948 030756  005237  002362                    INC     CPFLAG                  ;SET "CACHE PRESENT" FLAG
 1949 030762                            10$:    CLRVEC  #VEC4                   ;RESTORE VECTOR
      030762  012700  000004                    MOV     #VEC4,R0
      030766  104436                            TRAP    C$CVEC
 1950 030770  005037  002316                    CLR     TRP4FG                  ;MORE HOUSEKEEPING
 1951 030774  000207                            RTS     PC
 1952
```

```
                        ;;**********************************************************************
                        ;;**********************************************************************
                        ;
                        ;KTTEST
                        ;       THIS SUBROUTINE IS USED BY THE INIT CODE TO
                        ;       DETERMINE IF THE MEMORY MANAGEMENT UNIT IS
                        ;       PRESENT.  IF SO, IT RETURNS A FLAG IN THE
                        ;       SET STATE.  OTHERWISE THE FALG IS CLEAR IN
                        ;       WHICH CASE TEST SEVEN IS BYPASSED.
                        ;
                        ;;**********************************************************************
                        ;;**********************************************************************

030776                  KTTEST::
030776                          SETVEC  @VEC4,@TRAP4,@PRIO7     ;SET UP FOR POSSIBLE NXM
030776  012746  000340          MOV     @PRIO7,-(SP)
031002  012746  030656          MOV     @TRAP4,-(SP)
031006  012746  000004          MOV     @VEC4,-(SP)
031012  012746  000003          MOV     #3,-(SP)
031016  104437                  TRAP    C$SVEC
031020  062706  000010          ADD     #10,SP
031024  005737  177572          TST     MMUSR0          ;ARE YOU THERE, MMU?
031030                          DELAY   1               ;GIVE NXM TIMEOUT A CHANCE
031030  012727  000001          MOV     #1,(PC)+
031034  000000                  .WORD   0
031036  013727  002116          MOV     L$DLY,(PC)+
031042  000000                  .WORD   0
031044  005367  177772          DEC     -6(PC)
031050  001375                  BNE     .-4
031052  005367  177756          DEC     -22(PC)
031056  001367                  BNE     .-20

031060  005737  002316          TST     TRP4FG          ;IF NXM OCCURRED
031064  001026                  BNE     NOKT            ;  THEN NO MMU IS PRESENT
031066  005237  002314          INC     KTFLAG          ;  ELSE SAY WE FOUND 18 BIT SO FAR

031072  005737  172516          TST     MMUSR3          ;NOW LOOK FOR 22 BIT MAPPING
031076                          DELAY   1               ;GIVE NXM A CHANCE
031076  012727  000001          MOV     #1,(PC)+
031102  000000                  .WORD   0
031104  013727  002116          MOV     L$DLY,(PC)+
031110  000000                  .WORD   0
031112  005367  177772          DEC     -6(PC)
031116  001375                  BNE     .-4
031120  005367  177756          DEC     -22(PC)
031124  001367                  BNE     .-20

031126  005737  002316          TST     TRP4FG          ;IF NXM OCCURRED
031132  001005                  BNE     KTEXT           ;  THEN 18 BIT IS ALL WE'VE GOT
031134  005237  002314          INC     KTFLAG          ;  ELSE SAY WE'VE GOT 22 BIT
031140  000402                  BR      KTEXT           ;   AND BRANCH AROUND NEXT

031142  005037  002314  NOKT:   CLR     KTFLAG          ;NO MMU - CLEAR FLAG

031146                  KTEXT:  CLRVEC  @VEC4           ;RESTORE VECTOR
031146  012700  000004          MOV     @VEC4,R0
```

GLOBAL SUBROUTINES SECTION

```
        031152  104436                          TRAP    C$CVEC
  1994  031154  005037  002316                  CLR     TRP4FG          ;MORE HOUSEKEEPING
  1995  031160  000207                          RTS     PC
  1996
  1997
```

GLOBAL SUBROUTINES SECTION

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

031162                                  RSTVEC::
031162                                          SETVEC  TUVEC(R4),@ILLINT,@PRI00
031162  012746  000000                          MOV     @PRI00,-(SP)
031166  012746  030674                          MOV     @ILLINT,-(SP)
031172  016446  000004                          MOV     TUVEC(R4),-(SP)
031176  012746  000003                          MOV     @3,-(SP)
031202  104437                                  TRAP    C$SVEC
031204  062706  000010                          ADD     @10,SP

031210  000207                                  RTS     PC
```

```
 2030
 2031
 2032                                    ;;*****************************************************************
 2033                                    ;;*****************************************************************
 2034                                    ;
 2035                                    ;VECTOR
 2036                                    ;       THIS ROUTINE IS CALLED FROM VARIOUS PLACES
 2037                                    ;       IN THE PROGRAM TO SET THE UUT'S VECTOR WITH
 2038                                    ;       THE ADDRESS OF A HANDLER ROUTINE WHEN DEVICE
 2039                                    ;       INTERRUPTS HAVE BEEN ENABLED.  THE ROUTINE HAS
 2040                                    ;       TWO MODES OF OPERATION: WHEN BRFLAG IS CLEAR,
 2041                                    ;       PROCESSOR PRIORITY IS SET TO ZERO, ALLOWING
 2042                                    ;       DEVICE INTERRUPTS.  IF BRFLAG IS SET, PRIORITY
 2043                                    ;       IS SET TO 7.  IF AN INTERRUPT OCCURS IN THIS
 2044                                    ;       CASE, AN ERROR IS RETURNED BY THE HANDLER
 2045                                    ;       ROUTINE, "INTRCV".
 2046                                    ;
 2047                                    ;;*****************************************************************
 2048                                    ;;*****************************************************************
 2049
 2053 031212                            VECTOR::
 2054 031212  032764  000004  000014            BIT     #BRFLAG,LUNFLG(R4)      ;IF FLAG IS SET
 2055 031220  001014                            BNE     5$                      ;  THEN SKIP TO SECOND HALF
 2056 031222                                    SETVEC  TUVEC(R4),#INTRCV,#PRI00        ;ELSE LOW PRIORITY
      031222  012746  000000                    MOV     #PRI00,-(SP)
      031226  012746  030664                    MOV     #INTRCV,-(SP)
      031232  016446  000004                    MOV     TUVEC(R4),-(SP)
      031236  012746  000003                    MOV     #3,-(SP)
      031242  104437                            TRAP    C$SVEC
      031244  062706  000010                    ADD     #10,SP
 2057 031250  000413                            BR      EXTVEC                  ;RETURN
 2058
 2059 031252                            5$:     SETVEC  TUVEC(R4),#INTRCV,#PRI07        ;HIGH PRIORITY
      031252  012746  000340                    MOV     #PRI07,-(SP)
      031256  012746  030664                    MOV     #INTRCV,-(SP)
      031262  016446  000004                    MOV     TUVEC(R4),-(SP)
      031266  012746  000003                    MOV     #3,-(SP)
      031272  104437                            TRAP    C$SVEC
      031274  062706  000010                    ADD     #10,SP
 2060
 2061 031300  000207                    EXTVEC: RTS     PC
 2062
 2063
```

```
2068
2069
2070                                    ;;**********************************************************************
2071                                    ;;**********************************************************************
2072                                    ;
2073                                    ;PDELAY
2074                                    ;       THIS ROUTINE IS USED THROUGHOUT THE PROGRAM TO PROVIDE
2075                                    ;       A VARIABLE AMOUNT OF DELAY TIME.  THE DELAY WILL BE
2076                                    ;       INSTRUCTION EXECUTION TIME DEPENDENT.  TWO VALUES MUST
2077                                    ;       BE LOADED BY MAINLINE CODE PRIOR TO CALLING PDELAY:
2078                                    ;       "INNER" AND "OUTER".  IF SUFFICIENT CALLS TO PDELAY ARE
2079                                    ;       MADE SUCH THAT THE OUTER COUNT IS EXHAUSTED, THE ROUTINE
2080                                    ;       RETURNS "TOUT" EQUAL TO 1, INDICATING TIMEOUT HAS OCCURRED.
2081                                    ;       "INNER" SHOULD BE RE-LOADED BY MAINLINE CODE, PRIOR TO
2082                                    ;       CALL TO PDELAY WITHIN A TIMING LOOP.
2083                                    ;
2084                                    ;;**********************************************************************
2085                                    ;;**********************************************************************
2089
2090
2091 031302                            PDELAY::
2092 031302  005037  002350                     CLR     TOUT            ;CLEAR TIMEOUT INDICATOR
2093 031306  005337  002344                     DEC     INNER           ;IF COUNT NOT EXHAUSTED
2094 031312  001373                             BNE     PDELAY          ;  THEN KEEP LOOPING
2095 031314  005337  002346                     DEC     OUTER           ;IF MAJOR COUNT NOT 0
2096 031320  001002                             BNE     PDLYEX          ;  THEN LEAVE WITH STATUS = OK
2097 031322  005237  002350                     INC     TOUT            ;  ELSE SET TIMEOUT
2098 031326  000207                    PDLYEX: RTS     PC
2099
2100
```

```
2105
2106                                    ;;*****************************************************************
2107                                    ;;*****************************************************************
2108                                    ;
2109                                    ;STEP1
2110                                    ;       THIS SUBROUTINE IS RESPONSIBLE FOR PERFORMING
2111                                    ;       STEP 1 OF THE UQ-PORT INIT SEQUENCE.  SPECIFI-
2112                                    ;       CALLY, IT WILL INITIALIZE THE UUT BY WRITING
2113                                    ;       TO ITS IP REGISTER.  AFTER A BRIEF DELAY, IT
2114                                    ;       WILL READ THE SA REGISTER TO ENSURE THAT THE
2115                                    ;       STEP 1 BIT IS SET AND THE ERROR BIT IS CLEAR.
2116                                    ;       IT WILL THEN WRITE THE FIRST LOCATION OF THE
2117                                    ;       STEP TABLE (SET UP BY MAINLINE CODE) TO THE
2118                                    ;       UUT'S SA REG.  IF ALL STEPS COMPLETE SUCCESS-
2119                                    ;       FULLY THE ROUTINE RETURNS "STEPST" CLEARED;
2120                                    ;       OTHERWISE "STEPST" IS RETURNED INDICATING A
2121                                    .       FAILURE OCCURRED.
2122                                    ;
2123                                    ;;*****************************************************************
2124                                    ;;*****************************************************************
2125
2126 031330                            STEP1::
2127 031330  005037  002340                      CLR     STEPST          ;CLEAR THE STATUS INDICATOR
2128 031334  012774  000000  000000              MOV     #0,@TUIP(R4)    ;INIT THE UUT
2129 031342  012727  000001                      MOV     #1,(PC)
     031346  000000                              .WORD   0
     031350  013727  002116                      MOV     L$DLY,(PC)+
     031354  000000                              .WORD   0
     031356  005367  177772                      DEC      6(PC)
     031362  001375                              BNE     .-4
     031364  005367  177756                      DEC     -22(PC)
     0313..  001367                              BNE     .-20
2130 031372  017464  000002  000012              MOV     @TUSA(R4),TUSASV(R4)    ;GET THE SA REG CONTENTS
21.. 031400  022764  004600  000012              CMP     #B.S1!B.DI!B.OD,TUSASV(R4)
2132
2133                                                                    ;IF ALL THE RIGHT BITS AREN'T SET
2134 031406  001004                              BNE     STP1ER          ;  THEN TAKE ERROR EXIT
2135 031410  013774  00227.  000002              MOV     STPTBL,@TUSA(R4);  ELSE WRITE HOST'S STEP 1 RESPONSE
2136 031416  000402                              BR      STP1EX          ;    AND LEAVE SHOWING SUCCESS
2137
2138 031420  005237  002340          STP1ER: INC     STEPST          ;SET ERROR INDICATOR
2139
2140 031424  000207                  STP1EX: RTS     PC
2141
```

```
2146                                    ;;**********************************************************************
2147                                    ;;**********************************************************************
2148                                    ;
2149                                    ;BAKPAT
2150                                    ;       THIS SUBROUTINE WILL FILL THE COMMUNICATION WITH AN
2151                                    ;       ALL 1'S DATA PATTERN.  THE LENGTH OF THE AREA IN USE
2152                                    ;       BY THE CURRENT TEST IS CONTAINED IN "CMARLG".
2153                                    ;
2154                                    ;;**********************************************************************
2155                                    ;;**********************************************************************
2156
2160 031426                            BAKPAT::
2161 031426  012702  060000                       MOV     #COMMBF,R2              ;STARTING ADDRESS OF COMM AREA
2162                                                                               ;-20 WORDS
2163 031432  012703  000024                       MOV     #20.,R3                 ;BUFFER LENGTH IN FRONT OF AREA
2164 031436  006303                               ASL     R3                      ;MULTIPLIED BY 2
2165 031440  063703  002326                       ADD     CMARLG,R3               ;ADD COMM AREA LENGTH USED
2166 031444  012722  177777             1$:       MOV     #-1,(R2)+               ;WRITE THE DATA
2167 031450  005303                               DEC     R3                      ;IF NOT DONE YET
2168 031452  C01374                               BNE     1$                      ;  THEN DO IT AGAIN
2169
2170 031454  000207                               RTS     PC
2171
```

```
2176
2177
2178                                    ;;**********************************************************************
2179                                    ;;**********************************************************************
2180                                    ;
2181                                    ;CHKCOM
2182                                    ;       THIS ROUTINE IS CALLED BY TESTS DOING THE PURGE/POLL
2183                                    ;       CHECK.  IT IS USED TO VERIFY THAT THE PORT LEFT THE
2184                                    ;       COMMUNICATIONS AREA CLEARED.  ADDITIONALLY, IT CHECKS
2185                                    ;       THE 20 WORDS PRECEDING AND SUCCEEDING THE COMM AREA
2186                                    ;       TO MAKE SURE THE PORT DIDN'T GO OUTSIDE THE COMM AREA.
2187                                    ;
2188                                    ;;**********************************************************************
2189                                    ;;**********************************************************************
2193
2194 031456                            CHKCOM::
2195 031456  012701  177777                    MOV     #-1,R1                  ;TEST DATA
2196 031462  012702  060000                    MOV     #COMMBF,R2              ;STARTING ADDRESS
2197 031466  012703  000022                    MOV     #18.,R3                 ;FIRST COUNT
2198 031472  C20122                    1$:     CMP     R1,(R2)+                ;IF NOT ALL 1'S
2199 031474  001022                            BNE     15$                     ;  THEN GO REPORT ERROR
2200 031476  005303                            DEC     R3                      ;IF NOT ALL DONE
2201 031500  001374                            BNE     1$                      ;  THEN GO CHECK ANOTHER
2202
2203 031502  005001                            CLR     R1                      ;TEST DATA FOR PRINTOUT
2204 031504  013703  002326                    MOV     CMARLG,R3               ;SET UP COUNTER FOR COMM APEA
2205 031510  005722                    5$:     TST     (R2)+                   ;IF NOT 0
2206 031512  001013                            BNE     15$                     ;  THEN GO REPORT ERROR
2207 031514  005303                            DEC     R3                      ;IF NOT ALL DONE
2208 031516  001374                            BNE     5$                      ;  THEN GO CHECK ANOTHER
2209
2210 031520  012701  177777                    MOV     #-1,R1                  ;TEST DATA FOR PRINTOUT
2211 031524  012703  000024                    MOV     #20.,R3                 ;SET UP COUNTER FOR POST COMM AREA
2212 031530  020122                    10$:    CMP     R1,(R2)+                ;IF NOT ALL 1'S
2213 031532  001003                            BNE     15$                     ;  THEN GO REPORT ERROR
2214 031534  005303                            DEC     R3                      ;IF NOT ALL DONE
2215 031536  001374                            BNE     10$                     ;  THEN GO CHECK ANOTHER
2216 031540  000425                            BR      CKCMEX                  ;  ELSE RETURN
2217
2218 031542  162702  000002            15$:    SUB     #2,R2                   ;ADJUST ADDRESS FOR PRINTOUT
2219 031546  012737  025632  002330            MOV     #LESI,FRUIS             ;LOAD FAILING FRU
2220 031554  022737  000010  002114            CMP     #8.,L$TEST              ;IF IN TEST 8
2221 031562  001405                            BEQ     20$                     ;  THEN DO ALTERNATE PRINTOUT
2222 031564                                    ERRDF   1,EMSG14,PRIPAD         ;"PURGE/POLL TEST FAILED"
     031564  104455                            TRAP    C$ERDF
     031566  000001                            .WORD   1
     031570  024503                            .WORD   EMSG14
     031572  027066                            .WORD   PRIPAD
2223 031574  000404                            BR      25$                     ;COMMON EXIT
2224
2225 031576                            20$:    ERRDF   2,EMSG15,PRIVAD         ;"EXTENDED ADDRESS TEST FAILED"
     031576  104455                            TRAP    C$ERDF
     031600  000002                            .WORD   2
     031602  024532                            .WORD   EMSG15
     031604  027114                            .WORD   PRIVAD
2226
2227 031606                            25$:    DODU    LOGUNT
```

```
        031606  013700  002332                  MOV     LOGUNT,R0
        031612  104451                          TRAP    C$DODU
   2228
   2229 031614  000207                  CKCMEX: RTS     PC
   2230
```

```
                        ;;*****************************************************************
                        ; ******************************************************************
                        ;
                        ;I'  MU
                        ;       THIS SUBROUTINE IS CALLED FROM TEST 8 TO INITIALIZE
                        ;       MEMORY MANAGEMENT REGISTERS.  ALL PAR'S EXCEPT ONE
                        ;       ARE SET UP TO MAP VIRTUAL ADDRESSES INTO THE LOWEST
                        ;       32K OF PHYSICAL MEMORY.  KPAR7 IS SET UP TO MAP TO
                        ;       THE I/O PAGE.  THE PAR REGISTER THAT CORRESPONDS TO
                        ;       THE VIRTUAL ADDRESS OF THE COMMUNICATION AREA IS SET
                        ;       UP TO POINT TO THE SECOND 32K OF PHYSICAL MEMORY.
                        ;       ALL PDR'S ARE INITIALIZED TO THE SAME VALUE; NAMELY,
                        ;       UPWARD EXPANDABLE, READ/WRITE ACCESS ENABLED, AND THE
                        ;       FULL 8KBYTE PAGE IS ACCESSIBLE.
                        ;
                        ;;*****************************************************************
                        ;;*****************************************************************
031616                  INTMMU::
031616  012703  172300          MOV     #KPDR0,R3               ;START OF PDR ADDRESS RANGE
031622  012702  172340          MOV     #KPAR0,R2               ;START OF PAR ADDRESS RANGE
031626  005001                  CLR     R1                      ;STARTING RELOCATION VALUE
                        
031630  010122          1$:     MOV     R1,(R2)+                ;LOAD RELOCATION VALUE
031632  012723  077406          MOV     #77406,(R3)+            ;LOAD PDR
031636  062701  000200          ADD     #200,R1                 ;ADJUST RELOCATION VALUE
031642  022701  002000          CMP     #2000,R1                ;IF NOT AT THE END
031646  001370                  BNE     1$                      ;  THEN DO ANOTHER ONE
                        
031650  010137  172346          MOV     R1,KPAR3                ;  ELSE SET THIS REG TO NEXT 32K
031654  012737  007600  172356  MOV     #7600,KPAR7             ;18 BIT I/O PAGE
031662  032737  000002  002314  BIT     #BIT1,KTFLAG            ;IF 22-BIT BUS NOT AVAILABLE
031670  001406                  BEQ     2$                      ;  THEN GO TURN MMU ON
031672  012737  177600  172356  MOV     #177600,KPAR7           ;  ELSE SET 22 BIT I/O PAGE
031700  012737  000020  172516  MOV     #MM22ON,MMUSR3          ;  AND ENABLE 22 BIT MAPPING
                        
031706  012737  000001  177572  2$:     MOV     #MMON,MMUSR0    ;TURN ON THE WHOLE THING
031714  000207                  RTS     PC
                        
                        
031716                  PRTINT::
031716  010174  000000          MOV     R1,@TUIP(R4)            ;INITIALIZE THE DRIVE
031722  012703  032140          MOV     #INTTBL,R3              ;PUT THE TABLE ADDRESS INTO R3
031726  012701  004000          MOV     #S1,R1                  ;SET UP TO BEGIN AT STEP 1
031732  005037  002336          CLR     INISTP                  ;CLEAR THE STEP TRACKER
031736  012737  000030  002736  LOOP:   MOV     #24.,CNTHI      ;SET UP THE TIME OUT COUNTER
031744  005002                  CLR     R2                      ;CLEAR R2
031746  005202          ILOOP:  INC     R2                      ;INCREMENT HI TIME OUT VALUE ?
031750  001016                  BNE     2$                      ;IF NOT, BRANCH
031752  005337  002736          DEC     CNTHI                   ;ELSE, DECREMENT LO TIMEOUT
031756  001013                  BNE     2$                      ;BRANCH IF NO TIME OUT
031760  017464  000002  000012  MOV     @TUSA(R4),TUSASV(R4)    ;SAVE THE SA FOR THE ERROR PRINTOUT
031766                          ERRDF   51.,WRER1,WRINTO        ;PRINT PORT INIT FAILURE
031766  104455                  TRAP    C$ERDF
031770  000063                  .WORD   51
```

GLOBAL SUBROUTINES SECTION

```
        031772  025250                          .WORD   WRER1
        031774  027226                          .WORD   WRINTO
   2293 031776                                  DODU    LOGUNT                  ;DROP THE UNIT
        031776  013700  002332                  MOV     LOGUNT,R0
        032002  104451                          TRAP    C$DODU
   2294 032004  000454                          BR      100$                    ;EXIT ROUTINE
   2295 032006  037401  000002          2$:     BIT     @TUSA(R4),R1            ;TEST FOR STEP BIT FROM DRIVE
   2296 032012  001755                          BEQ     LOOP                    ;LOOP UNTIL SOMETHING SETS
   2297 032014  032774  100000  000002          BIT     #ERR,@TUSA(R4)          ;CHECK FOR ERROR
   2298 032022  001413                          BEQ     3$                      ;NO ERROR, KEEP GOING
   2299 032024  017464  000002  000012          MOV     @TUSA(R4),TUSASV(R4)    ;SAVE THE SA CONTENTS
   2300 032032                                  ERRDF   52.,WRER2,WRPRTE        ;PRINT ERROR
        032032  104455                          TRAP    C$ERDF
        032034  000064                          .WORD   52
        032036  025302                          .WORD   WRER2
        032040  027252                          .WORD   WRPRTE
   2301 032042                                  DODU    LOGUNT                  ;DROP THE UNIT
        032042  013700  002332                  MOV     LOGUNT,R0
        032046  104451                          TRAP    C$DODU
   2302 032050  000432                          BR      100$                    ;EXIT ROUTINE
   2303 032052  005237  002336          3$:     INC     INISTP                  ;INCREMENT THE STEP TRACKER
   2304 032056  012374  000002                  MOV     (R3)+,@TUSA(R4)         ;WRITE WORD FROM TABLE TO CONTROLLER
   2305 032062  006301                          ASL     R1                      ;SHIFT TO NEXT STEP
   2306 032064  100324                          BPL     LOOP                    ;IF NOT AT LAST STEP LOOP
   2307 032066  012702  002716                  MOV     #RSPRNG,R2              ;PUT THE RESPONSE DESCRIPTOR ADD IN R2
   2308 032072  012703  002506                  MOV     #RSPBUF,R3              ;PUT THE RESPONSE BUFFER ADDRESS IN R3
   2309 032076  010322                  5$:     MOV     R3,(R2)+                ;PUT THE BUFF ADD IN THE DESCRIPTOR
   2310 032100  012722  100000                  MOV     #OWN,(R2)+              ;SET THE DESCRIPTOR TO THE CONTROLLER
   2311 032104  062703  000104                  ADD     #RSPSTP,R3              ;STEP TO THE NEXT BUFFER SLOT
   2312 032110  022703  002716                  CMP     #RSPEND,R3              ;ARE WE AT THE END OF THE BUFFER ?
   2313 032114  001370                          BNE     5$                      ;NO, KEEP GOING
   2314 032116  012737  002716  022752          MOV     #RSPRNG,RSPSAV          ;SET UP TO USE FIRST RESPONSE BUFFER
   2315 032124  012737  002726  022750          MOV     #CMDRNG,CMDSAV          ;SET UP TO USE FIRST COMMAND BUFFER
   2316 032132  005037  002744                  CLR     CMDREF                  ;SET THE COMMAND REFERENCE # TO 0
   2317 032136  000207                  100$:   RTS     PC                      ;RETURN
   2318
   2319                                         ;INIT DATA TABLE
   2320 032140  104400                  INTTBL:         .WORD   104400
   2321 032142  002716                                  .WORD   RSPRNG
   2322 032144  000000                                  .WORD   0
   2323 032146  000001                                  .WORD   GO
```

```
2325 032150  005064  000014          DRVTST: CLR     LUNFLG(R4)              ;CLEAR ALL FLAGS
2326 032154  005037  002356                  CLR     PROGRL                  ;CLEAR LOW WORD OF PROGRESS INDICATOR
2327 032160  005037  002360                  CLR     PROGRH                  ;CLEAR HIGH WORD OF PROGRESS INDICATOR
2328 032164  012737  025647  002330          MOV     @CTRL,FRUIS             ;DEFAULT FRU IS CONTROLLER
2329 032172  004737  031716                  JSR     PC,PRTINT               ;GO DO A PORT INIT
2330 032176  032764  000001  000014          BIT     @DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
2331 032204  001060                          BNE     100$                    ;NO, BRANCH TO EXIT
2332 032206  012705  002410                  MOV     @EXELOC,R5              ;SET UP FOR "EXECUTE LOCAL PROGRAM"
2333 032212  004737  032350                  JSR     PC,CLSDRV               ;GO ISSUE THE COMMAND
2334 032216  032764  000001  000014          BIT     @DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
2335 032224  001050                          BNE     100$                    ;NO, BRANCH TO EXIT
2336 032226  012705  002436                  MOV     @RCVDAT,R5              ;SET UP FOR "RECEIVE DATA"
2337 032232  004737  032350                  JSR     PC,CLSDRV               ;GO ISSUE THE COMMAND
2338 032236  005001                  10$:    CLR     R1                      ;CLEAR LOW DELAY COUNTER
2339 032240  012702  000024                  MOV     #20.,R2                 ;SET UP HIGH DELAY COUNTER
2340 032244  032737  000200  177560  30$:    BIT     #BIT7,RCSR              ;"CONTROL C" INPUT ?
2341 032252  001021                          BNE     50$                     ;YES, BRANCH
2342 032254  005201                          INC     R1                      ;DELAY BETWEEN "GET DUST STATUS" COMMANDS
2343 032256  001372                          BNE     30$
2344 032260  005302                          DEC     R2
2345 032262  001370                          BNE     30$
2346 032264  012705  002370                  MOV     @GDUST,R5               ;SET UP FOR "GET DUST STATUS"
2347 032270  004737  032350                  JSR     PC,CLSDRV               ;GO ISSUE THE COMMAND
2348 032274  032764  000001  000014          BIT     @DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
2349 032302  001021                          BNE     100$                    ;NO, BRANCH TO EXIT
2350 032304  032764  000020  000014          BIT     @DONEFL,LUNFLG(R4)      ;INTERNAL TEST DONE ?
2351 032312  001015                          BNE     100$                    ;YES, BRANCH TO EXIT
2352 032314  000750                          BR      10$                     ;LOOP
2353 032316  013705  177562          50$:    MOV     RBUF,R5                 ;GET DATA INPUT FROM KEYBOARD
2354 032322  042705  000200                  BIC     #BIT7,R5                ;STRIP PARITY
2355 032326  022705  000003                  CMP     #CNTRLC,R5              ;"CONTROL C" INPUT ?
2356 032332  001344                          BNE     30$                     ;NO, BRANCH
2357 032334  012705  002466          40$:    MOV     #ABORT,R5               ;SET UP FOR "ABORT"
2358 032340  004737  032350                  JSR     PC,CLSDRV               ;GO ISSUE THE COMMAND
2359 032344                                  BREAK
     032344  104422                          TRAP    C$BRK
2360 032346  000207                  100$:   RTS     PC                      ;RETURN
2361
2362
2363
2364
2365 032350                          CLSDRV::
2366 032350  004737  032456          1$:     JSR     PC,PRTDRV               ;GO SEND THE COMMAND
2367 032354  032764  000001  000014          BIT     @DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
2368 032362  001034                          BNE     100$                    ;GET OUT IF NOT AVAILABLE
2369 032364  020527  002436                  CMP     R5,@RCVDAT              ;"RECEIVE DATA" COMMAND JUST ISSUED ?
2370 032370  001431                          BEQ     100$                    ;YES, BRANCH TO EXIT
2371 032372  004737  032556                  JSR     PC,CDRECV               ;GO CHECK FOR ANY NEW RESPONSES
2372 032376  032764  000001  000014          BIT     @DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
2373 032404  001023                          BNE     100$                    ;GET OUT IF NOT AVAILABLE
2374 032406  004737  033042                  JSR     PC,CHKRSP               ;GO CHECK CONTENTS OF RESPONSE
2375 032412  032764  000001  000014          BIT     @DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
2376 032420  001015                          BNE     100$                    ;GET OUT IF NOT AVAILABLE
2377 032422  022705  002436                  CMP     @RCVDAT,R5              ;WAS IT A "RECEIVE DATA" COMMAND ?
2378 032426  001012                          BNE     100$                    ;NO, BRANCH TO EXIT
2379 032430  004737  033326                  JSR     PC,CHKMSG               ;GO CHECK MESSAGE FROM INTERNAL TEST
2380 032434  032764  000001  000014          BIT     @DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
```

```
2381 032442  001004                        BNE     100$                    ;GET OUT IF NOT AVAILABLE
2382 032444  012705  002370                MOV     #GDUST,R5               ;"GET DUST STAUS" PACKET ADDRESS
2383 032450  004737  032556                JSR     PC,CDRECV               ;GO GET LAST RESPONSE
2384 032454  000207                100$:   RTS     PC                      ;RETURN
2385
2386
2387
2388
2389  32456                         PRTDRV::
2390  32456  013701  022750                MOV     CMDSAV,R1               ;SET UP COMMAND RING POINTER
2391 032462  010511                        MOV     R5,(R1)                 ;PUT THE PACKET ADDRESS INTO THE DESCRIPTOR
2392 032464  012761  100000  000002        MOV     #OWN,HIADDR(R1)         ;SET THE OWNERSHIP BIT OF THE DESCRIPTOR
2393 032472  005774  000000                TST     @TUIP(R4)               ;READ THE IP REGISTER
2394 032476  005774  000002                TST     @TUSA(R4)               ;READ THE SA REGISTER
2395 032502  001413                        BEQ     10$                     ;BRANCH IF NO ERRORS
2396 032504  017464  000002  000012        MOV     @TUSA(R4),TUSASV(R4)    ;SAVE THE SA FOR THE ERROR PRINTOUT
2397 032512                                ERRDF   53.,WRER4,WRPRTE        ;PRINT PORT DETECTED ERROR
     032512  104455                        TRAP    C$ERDF
     032514  000065                        .WORD   53
     032516  025351                        .WORD   WRER4
     032520  027252                        .WORD   WRPRTE
2398 032522                                DODU    LOGUNT                  ;DROP THE UNIT
     032522  013700  002332                MOV     LOGUNT,R0
     032526  104451                        TRAP    C$DODU
2399 032530  000411                        BR      100$                    ;GET OUT
2400 032532  062701  000004        10$:    ADD     #RNGSTP,R1              ;ADJUST RESPONCE POINTER FOR NEXT TIME
2401 032536  022701  002736                CMP     #DSCEND,R1              ;ARE WE AT THE END ?
2402 032542  001002                        BNE     15$                     ;NO, GET OUT
2403 032544  012701  002726                MOV     #CMDRNG,R1              ;SET R1 TO TOP BUFFER
2404 032550  010137  022750        15$:    MOV     R1,CMDSAV               ;SAVE THE COMMAND RING LOCATION
2405 032554  000207                100$:   RTS     PC                      ;RETURN
2406
2407
2408
2409
2410 032556                         CDRECV::
2411 032556  004737  032670        1$:     JSR     PC,PDRECV               ;CALL PORT DRIVER RECEIVE
2412 032562  032764  000001  000014        BIT     #DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
2413 032570  001036                        BNE     100$                    ;GET OUT IF NOT AVAILABLE
2414 032572  032764  000020  000014        BIT     #DONEFL,LUNFLG(R4)      ;INTERNAL TEST DONE ?
2415 032600  001016                        BNE     10$                     ;YES, BRANCH
2416 032602  011103                        MOV     (R1),R3                 ;SET UP RESPONCE BUFFER POINTER
2417 032604  026365  000000  000000        CMP     P.CRF(R3),P.CRF(R5)     ;IS THIS THE RESPONSE THAT IS EXPECTED ?
2418 032612  001411                        BEQ     10$                     ;YES, BRANCH
2419 032614  022705  002370                CMP     #GDUST,R5               ;WAS IT A "GET DUST STATUS" COMMAND ?
2420 032620  001022                        BNE     100$                    ;NO, BRANCH TO EXIT
2421 032622  012705  002436                MOV     #RCVDAT,R5              ;GET START OF "RECEIVE DATA" PACKET
2422 032626  026365  000000  000000        CMP     P.CRF(R3),P.CRF(R5)     ;IS IT A "RECEIVE DATA" RESPONSE ?
2423 032634  001014                        BNE     100$                    ;NO, BRANCH TO EXIT
2424 032636  012761  100000  000002  10$:  MOV     #OWN,HIADDR(R1)         ;GIVE THE CONTROLLER THE RING BACK
2425 032644  062701  000004                ADD     #RNGSTP,R1              ;ADJUST RESPONCE POINTER FOR NEXT TIME
2426 032650  022701  002726                CMP     #CMDRNG,R1              ;ARE WE AT THE END ?
2427 032654  001002                        BNE     15$                     ;NO, GET OUT
2428 032656  012701  002716                MOV     #RSPRNG,R1              ;SET R1 TO TOP BUFFER
2429 032662  010137  022752        15$:    MOV     R1,RSPSAV               ;SAVE THE POINTER FOR NEXT TIME
2430 032666  000207                100$:   RTS     PC                      ;RETURN
2431
```

```
2432
2433
2434 032670                                PDRECV::
2435 032670  013701  022752                        MOV     RSPSAV,R1               ;PUT THE RESPONSE RING SAVE IN R1
2436 032674  012737  000005  002736  1$:     MOV     #5,CNTHI                ;SET UP THE TIME OUT COUNTER
2437 032702  005002                        CLR     R2                      ;CLEAR R2
2438 032704  005202                5$:     INC     R2                      ;INCREMENT HI TIME OUT VALUE ?
2439 032706  001026                        BNE     10$                     ;NO OVERFLOW YET, BRANCH
2440 032710  005337  002736                DEC     CNTHI                   ;ELSE, INCREMENT HI TIMEOUT
2441 032714  001023                        BNE     10$                     ;KEEP GOING ,NO TIME OUT YET
2442 032716  022705  002370                CMP     #GDUST,R5               ;WAS IT A "GET DUST STATUS" COMMAND ?
2443 032722  001410                        BEQ     6$                      ;YES, PRINT ERROR
2444 032724                                ERRDF   54.,EMSG16,FRUERR       ;"EXECUTE LOCAL PROGRAM' COMMAND TIMEOUT
     032724  104455                        TRAP    C$ERDF
     032726  000066                        .WORD   54
     032730  024567                        .WORD   EMSG16
     032732  030624                        .WORD   FRUERR
2445 032734                                DODU    LOGUNT                  ;GO DROP THE UNIT
     032734  013700  002332                MOV     LOGUNT,R0
     032740  104451                        TRAP    C$DODU
2446 032742  000436                        BR      100$                    ;GET OUT ON ERROR
2447 032744                        6$:     ERRDF   55.,EMSG17,FRUERR       ;"GET DUST STATUS" COMMAND TIMEOUT
     032744  104455                        TRAP    C$ERDF
     032746  000067                        .WORD   55
     032750  024637                        .WORD   EMSG17
     032752  030624                        .WORD   FRUERR
2448 032754                                DODU    LOGUNT                  ;GO DROP THE UNIT
     032754  013700  002332                MOV     LOGUNT,R0
     032760  104451                        TRAP    C$DODU
2449 032762  000426                        BR      100$                    ;GET OUT ON ERROR
2450 032764  017464  000002  000012  10$:     MOV     @TUSA(R4),TUSASV(R4)    ;GET SA CONTENTS
2451 032772  032764  100000  000012          BIT     #BIT15,TUSASV(R4)       ;CHECK FOR SA ERROR
2452 033000  001413                        BEQ     20$                     ;NO ERROR, BRANCH
2453 033002  012737  025670  002330        MOV     #LSCT,FRUIS             ;LOAD FAILING FRU
2454 033010                                ERRDF   56.,EMSG9,WRPRTE        ;PRINT "SA CONTENTS IN ERROR" MESSAGE
     033010  104455                        TRAP    C$ERDF
     033012  000070                        .WORD   56
     033014  024271                        .WORD   EMSG9
     033016  027252                        .WORD   WRPRTE
2455 033020                                DODU    LOGUNT                  ;DROP THE UNIT
     033020  013700  002332                MOV     LOGUNT,R0
     033024  104451                        TRAP    C$DODU
2456 033026  000404                        BR      100$                    ;GET OUT ON ERROR
2457 033030  032761  100000  000002  20$:     BIT     #OWN,HIADDR(R1)         ;IS THE SLOT SET TO US ?
2458 033036  001322                        BNE     5$                      ;KEEP GOING TILL TIMEOUT OR SUCCESS
2459 033040  000207                100$:   RTS     PC                      ;RETURN
2460
2461
2462
2463
2464 033042  026365  000000  000000  CHKRSP: CMP     P.CRF(R3),P.CRF(R5)     ;DID COMMAND REFERENCE NUMBERS MATCH ?
2465 033050  001003                        BNE     5$                      ;NO, BRANCH
2466 033052  005763  000012                TST     P.STS(R3)               ;WAS STATUS "NORMAL"?
2467 033056  001451                        BEQ     15$                     ;YES, BRANCH
2468 033060  022705  002410          5$:     CMP     #EXELOC,R5              ;WAS IT AN "EXEC LOC PROG" COMMAND ?
2469 033064  001416                        BEQ     7$                      ;YES, BRANCH
2470 033066  022705  002436                CMP     #RCVDAT,R5              ;WAS IT A "RECEIVE DATA" COMMAND ?
```

```
2471 033072  001423                        BEQ     8$                      ;YES, BRANCH
2472 033074  022705  002466                CMP     @ABORT,R5               ;WAS IT AN "ABORT" COMMAND ?
2473 033100  001430                        BEQ     9$                      ;YES, BRANCH
2474 033102                          6$:   ERRDF   57.,EMSG18,GDSERR       ;PRINT "GET DUST STATUS" COMMAND FAILURE
     033102  104455                        TRAP    C$ERDF
     033104  000071                        .WORD   57
     033106  024701                        .WORD   EMSG18
     033110  030256                        .WORD   GDSERR
2475 033112                                DODU    LOGUNT                  ;DROP THE UNIT
     033112  013700  002332                MOV     LOGUNT,R0
     033116  104451                        TRAP    C$DODU
2476 033120  000501                        BR      100$                    ;GET OUT ON ERROR
2477 033122                          7$:   ERRDF   58.,EMSG19,ELPERR       ;PRINT "EXECUTE LOCAL PROGRAM" COMMAND FAILURE
     033122  104455                        TRAP    C$ERDF
     033124  000072                        .WORD   58
     033126  024743                        .WORD   EMSG19
     033130  027306                        .WORD   ELPERR
2478 033132                                DODU    LOGUNT                  ;DROP THE UNIT
     033132  013700  002332                MOV     LOGUNT,R0
     033136  104451                        TRAP    C$DODU
2479 033140  000471                        BR      100$                    ;GET OUT ON ERROR
2480 033142                          8$:   ERRDF   59.,EMSG20,RCVERR       ;PRINT "RECEIVE DATA" COMMAND FAILURE
     033142  104455                        TRAP    C$ERDF
     033144  000073                        .WORD   59
     033146  025013                        .WORD   EMSG20
     033150  027726                        .WORD   RCVERR
2481 033152                                DODU    LOGUNT                  ;DROP THE UNIT
     033152  013700  002332                MOV     LOGUNT,R0
     033156  104451                        TRAP    C$DODU
2482 033160  000461                        BR      100$                    ;GET OUT ON ERROR
2483 033162                          9$:   ERRDF   60.,EMSG21,FRUERR       ;PRINT "ABORT" COMMAND FAILURE
     033162  104455                        TRAP    C$ERDF
     033164  000074                        .WORD   60
     033166  025052                        .WORD   EMSG21
     033170  030624                        .WORD   FRUERR
2484 033172                                DODU    LOGUNT                  ;DROP THE UNIT
     033172  013700  002332                MOV     LOGUNT,R0
     033176  104451                        TRAP    C$DODU
2485 033200  000451                        BR      100$                    ;GET OUT ON ERROR
2486 033202  022705  002370          15$:  CMP     @GDUST,R5               ;WAS IT A GET DUST STATUS COMMAND ?
2487 033206  001046                        BNE     100$                    ;NO, BRANCH TO EXIT
2488 033210  032764  000010  000014        BIT     @TEST.9,LUNFLG(R4)      ;ARE WE IN TEST 9 ?
2489 033216  001411                        BEQ     20$                     ;NO, GO CHECK PROGRESS INDICATOR
2490 033220  126327  000010  000201        CMPB    P.FNDC(R3),@201         ;CORRECT ENDCODE ?
2491 033226  001325                        BNE     6$                      ;NO, ERROR
2492 033230  126327  000017  000007        CMPB    P.FLGS(R3),@7           ;CORRECT FLAGS ?
2493 033236  001321                        BNE     6$                      ;NO, ERROR
2494 033240  000431                        BR      100$                    ;SUCCESS, RETURN
2495 033242  026337  000020  002356  20$:  CMP     P.IND1(R3),PROGRL       ;CHECK LOW WORD OF PROGRESS INDICATOR
2496 033250  003017                        BGT     50$                     ;PROGRESS BEING MADE, BRANCH
2497 033252  026337  000022  002360        CMP     P.IND2(R3),PROGRH       ;CHECK HIGH WORD OF PROGRESS INDICATOR
2498 033260  003013                        BGT     50$                     ;PROGRESS BEING MADE, BRANCH
2499 033262  012737  025716  002330  30$:  MOV     @DRVE,FRUIS             ;LOAD FAILING FRU
2500 033270                                ERRDF   61.,EMSG22,FRUERR       ;PRINT "INTERNAL TEST HUNG" ERROR
     033270  104455                        TRAP    C$ERDF
     033272  000075                        .WORD   61
     033274  025104                        .WORD   EMSG22
```

```
                033276  030624                          .WORD   FRUERR
     2501       033300                                  DODU    LOGUNT                  ;DROP THE UNIT
                033300  013700  002332                  MOV     LOGUNT,R0
                033304  104451                          TRAP    C$DODU
     2502       033306  000406                          BR      100$                    ;GET OUT ON ERROR
     2503       033310  016337  000020  002356  50$:    MOV     P.IND1(R3),PROGRL       ;UPDATE LOW WORD OF PROGRESS INDICATOR
     2504       033316  016337  000022  002360          MOV     P.IND2(R3),PROGRH       ;UPDATE HIGH WORD OF PROGRESS INDICATOR
     2505       033324  000207                  100$:   RTS     PC
     2506
     2507
     2508
     2509
     2510       033326  012701  060000          CHKMSG: MOV     #RDBUF,R1               ;GET START ADDRESS OF MESSAGE BUFFER
     2511       033332  121127  000001                  CMPB    (R1),#1                 ;NORMAL COMPLETION MESSAGE ?
     2512       033336  001446                          BEQ     100$                    ;YES, BRANCH TO EXIT
     2513       033340  121127  000002                  CMPB    (R1),#2                 ;ERROR COMPLETION MESSAGE ?
     2514       033344  001413                          BEQ     1$                      ;YES, BRANCH
     2515       033346  121127  000003                  CMPB    (R1),#3                 ;NORMAL COMPLETION WITH INFO. MESSAGE ?
     2516       033352  001440                          BEQ     100$                    ;YES, BRANCH TO EXIT
     2517       033354                                  ERRDF   62.,EMSG23,INVMSG       ;INVALID MESSAGE FROM INTERNAL TEST
                033354  104455                          TRAP    C$ERDF
                033356  000076                          .WORD   62
                033360  025135                          .WORD   EMSG23
                033362  030604                          .WORD   INVMSG
     2518       033364                                  DODU    LOGUNT                  ;DROP THE UNIT
                033364  013700  002332                  MOV     LOGUNT,R0
                033370  104451                          TRAP    C$DODU
     2519       033372  000430                          BR      100$                    ;GET OUT ON ERROR
     2520       033374  012737  025716  002330  1$:     MOV     #DRVE,FRUIS             ;GET FAILING FRU
     2521       033402  012702  024066                  MOV     #FAULTC,R2              ;GET ADDRESS OF ERROR MESSAGE
     2522       033406  116162  000002  000020          MOVB    2(R1),20(R2)            ;1ST ASCII BYTE OF FAULT CODE INTO MESSAGE
     2523       033414  116162  000003  000021          MOVB    3(R1),21(R2)            ;2ND ASCII BYTE OF FAULT CODE INTO MESSAGE
     2524       033422  116162  000004  000046          MOVB    4(R1),46(R2)            ;1ST ASCII BYTE OF SUB-FAULT CODE INTO MESSAGE
     2525       033430  116162  000005  000047          MOVB    5(R1),47(R2)            ;2ND ASCII BYTE OF SUB-FAULT CODE INTO MESSAGE
     2526       033436                                  ERRDF   63.,EMSG24,INTMSG       ;PRINT ERROR MESSAGE
                033436  104455                          TRAP    C$ERDF
                033440  000077                          .WORD   63
                033442  025215                          .WORD   EMSG24
                033444  030540                          .WORD   INTMSG
     2527       033446                                  DODU    LOGUNT                  ;DROP THE UNIT
                033446  013700  002332                  MOV     LOGUNT,R0
                033452  104451                          TRAP    C$DODU
     2528       033454  052764  000020  000014  100$:   BIS     #DONEFL,LUNFLG(R4)      ;SET DONE FLAG
     2529       033462  000207                          RTS     PC                      ;RETURN
     2530
     2531       033464                                  ENDMOD
     2532
     2543                                       .TITLE MISCELLANEOUS SECTIONS
     2544                                       .SBTTL  REPORT CODING SECTION
     2572
     2573       033464                                  BGNMOD
     2574                                       .SBTTL  INITIALIZE SECTION
     2575
     2576                                       ;++
     2577                                       ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
     2578                                       ; AT THE BEGINNING OF EACH PASS.
     2579                                       ;--
```

```
2580
2581 033464                                    BGNINIT
     033464                          L$INIT::
2582
2583
2584 033464                                    READEF  @EF.START               ;IF THIS IS A FRESH START
     033464  012700  000040                    MOV     @EF.START,R0
     033470  104447                            TRAP    C$REFG
2585 033472                                    BCOMPLETE       START           ;  THEN GO TO START
     033472  103421                            BCS     START
2586
2587 033474                                    READEF  @EF.RESTART             ;IF THIS IS A RESTART
     033474  012700  000037                    MOV     @EF.RESTART,R0
     033500  104447                            TRAP    C$REFG
2588 033502                                    BCOMPLETE       START           ;  THEN GO TO START
     033502  103415                            BCS     START
2589
2590 033504                                    READEF  @EF.PWR                 ;IF POWER-FAIL OCCURRED
     033504  012700  000034                    MOV     @EF.PWR,R0
     033510  104447                            TRAP    C$REFG
2591 033512                                    BCOMPLETE       START           ;  THEN START FROM THE BEGINNING
     033512  103411                            BCS     START
2592
2593 033514                                    READEF  @EF.NEW                 ;IF THIS IS A NEW PASS
     033514  012700  000035                    MOV     @EF.NEW,R0
     033520  104447                            TRAP    C$REFG
2594 033522                                    BCOMPLETE       NUPASS          ;  THEN SKIP START UP CODE
     033522  103422                            BCS     NUPASS
2595
2596 033524                                    READEF  @EF.CONTINUE            ;IF THIS IS A CONTINUE
     033524  012700  000036                    MOV     @EF.CONTINUE,R0
     033530  104447                            TRAP    C$REFG
2597 033532                                    BCOMPLETE       END             ;  THEN SKIP ALL INIT CODE
     033532  103465                            BCS     END
2598
2599 033534  000423                            BR      NEXT                    ;JUST HERE FOR NEXT UUT
2600
2601 033536                          START:
2602 033536  012737  000000  002312            MOV     @0,PASCNT               ;INITIALIZE PASS COUNT
2603 033544  005037  002314                    CLR     KTFLAG                  ;IN CASE WE'RE STARTED > THAN ONCE
2604 033550  012704  002234                    MOV     @LUNBLK,R4              ;R4 WILL ALWAYS POINT TO LUNBLK
2605 033554  022737  001400  002120            CMP     @1400,L$HIME            ;IF <= 28KWORDS OF MEMORY PRESENT
2606 033562  103002                            BHIS    NUPASS                  ;  THEN SKIP NEXT
2607 033564  004737  030776                    JSR     PC,KTTEST               ;  ELSE SEE IF MMU IS PRESENT
2608
2609 033570                          NUPASS: BRESET                          ;CLEAR THE WORLD
     033570  104433                            TRAP    C$RESET
2610 033572  005237  002312                    INC     PASCNT                  ;UPDATE THE PASS COUNT
2611 033576  012737  177777  002332            MOV     @-1,LOGUNT              ;INITIALIZE LOGICAL UNIT COUNT
2612
2613 033604  005237  002332          NEXT:     INC     LOGUNT                  ;POINT TO NEXT UUT
2614 033610  023737  002332  002012            CMP     LOGUNT,L$UNIT           ;IF WE'VE PASSED MAXIMUM UUT'S
2615 033616  001433                            BEQ     END                     ;  THEN LEAVE INIT
2616
2617 033620                                    GPHARD  LOGUNT,R0               ;GET P-TABLE FOR THIS UNIT
     033620  013700  002332                    MOV     LOGUNT,R0
     033624  104442                            TRAP    C$GPHRD
```

```
2618 033626                                    BNCOMPLETE      NEXT            ;TRY AGAIN
     033626  103366                            BCC     NEXT
2619
2620 033630  011064  000000                    MOV     (R0),TUIP(R4)           ;PUT IP REG ADDRESS IN LUNBLK
2621 033634  012064  000002                    MOV     (R0)+,TUSA(R4)          ; AND ANOTHER COPY IN LUNBLK
2622 033640  062764  000002  000002            ADD     #2,TUSA(R4)             ;MAKE IT THE SA REG ADDRESS
2623 033646  012064  000004                    MOV     (R0)+,TUVEC(R4)         ;GET THE VECTOR INTO THE LUNBLK
2624 033652  011064  000006                    MOV     (R0),MSCPUN(R4)         ;PUT THE T/MSCP UNIT # IN LUNBLK
2625 033656  004737  031162                    JSR     PC,RSTVEC               ;SET UUT VECTOR FOR ILLEGAL INTRPTS.
2626 033662                                    PRINTF  #IMSG,LOGUNT            ;"TESTING UNIT N"
     033662  013746  002332                    MOV     LOGUNT,-(SP)
     033666  012746  033712                    MOV     #IMSG,-(SP)
     033672  012746  000002                    MOV     #2,-(SP)
     033676  010600                            MOV     SP,R0
     033700  104417                            TRAP    C$PNTF
     033702  062706  000006                    ADD     #6,SP
2627
2628 033706                            END:
2629 033706                                    EXIT    INI1
     033706  104432                            TRAP    C$EXIT
     033710  000032                            .WORD   L10007-.
2630
2642 033712     045     116     045    IMSG:   .ASCIZ  ?%N%ATESTING UNIT %D1%N?
2643                                           .EVEN
2644
2645 033742                                    ENDINIT
     033742                            L10007:
     033742  104411                            TRAP    C$INIT
```

```
                                .SBTTL  CLEANUP CODING SECTION

                                ;++
                                ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
                                ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
                                ;--

033744                                  BGNCLN
033744                          L$CLEAN::
033744  032764  000000G 002234          BIT     #T9FLAG,LUNBLK(R4)      ;IF NOT HERE FROM TEST 9
033752  001400                          BEQ     ENDCLE                  ;  THEN SKIP THE REST

                                ;EVENTUALLY MORE CODE WILL BE PLACED HERE TO GUARANTEE THAT AN ABORT
                                ;COMMAND IS ISSUED TO THE UUT TO STOP EXECUTION OF THE LOCAL PROGRAM.

033754  005064  000014          ENDCLE: CLR     LUNFLG(R4)              ;CLEAR OUT THE LUN FLAGS

                                ;NOTE: THIS LINE OF CODE MAY HAVE TO BE REMOVED TO HANDLE ^C FOLLOWED
                                ;BY A PROCEED COMMAND CORRECTLY.
033760                                  CLRVEC  TUVEC(R4)               ;PUT "TRAP CATCHER" INTO VECTOR
033760  016400  000004                    MOV     TUVEC(R4),R0
033764  104436                          TRAP    C$CVEC

033766                                  EXIT    CLN
033766  104432                          TRAP    C$EXIT
033770  000002                          .WORD   L10010-.

                                .EVEN

                                ENDCLN
033772                          L10010:
033772  104412                          TRAP    C$CLEAN
```

```
                                .SBTTL  DROP UNIT SECTION

                                ;++
                                ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                ; TO NO LONGER BE TESTED.
                                ;--

033774                                  BGNDU
033774                          L$DU::
033774  012764  000001  000014          MOV     #DRPFLG,LUNFLG(R4)      ;LETS PROGRAM KNOW IT'S DEAD
034002                                  EXIT    DU
034002  000167                          .WORD   J$JMP
034004  000000                          .WORD   L10011-2-.

                                        .EVEN

034006                                  ENDDU
034006                          L10011:
034006  104453                          TRAP    C$DU
```

```
                                .SBTTL  ADD UNIT SECTION

                                ;**
                                ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
                                ;  TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
                                ;  TO THE TEST CYCLE.
                                ;--

034010                                  BGNAU
034010                          L$AU::
034010                                  EXIT    AU
034C10  000167                          .WORD   J$JMP
034012  000000                          .WORD   L10012-2-.

                                        .EVEN

                                        ENDAU
034014                          L10012:
034014  104452                          TRAP    C$AU

034016                                  ENDMOD

                                .TITLE HARDWARE TEST
                        000000  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 TU81.  VECTOR 4 IS SET UP WITH
                                ;       A TRAP HANDLING ROUTINE IN CASE OF A NON-EXISTENT
                                ;       MEMORY TIMEOUT.
                                ;
                                ;;*****************************************************************
                                ;;*****************************************************************

034016                                  BGNTST
034016                          T1::
034016  000240                          NOP
034020  012737  000001  000000G         MOV     #1,ITRCNT               ;SET UP FOR ONE TEST ITERATION
034026  005737  002312                  TST     PASCNT                  ;IF PASS 0
034032  001404                          BEQ     T1.1                    ;  THEN START TEST
034034  012737  000010  000000G         MOV     #1C,ITRCNT              ;  ELSE DO MULTIPLE ITERATIONS
```

```
                                        NOP
                                        P^NSUB
                                T1.1:
                                        TRAP    C$BSUB
2819 034046  005037  002316       1$:    CLR     TRP4FG                  ;CLEAR NXM TRAP FLAG
2820
2821 034052                             SETVEC  #VEC4,^TRAP4,#PRIO7     ;SET UP VECTOR 4 FOR NXM TRAP
     034052  012746  000340             MOV     #PRIO7,-(SP)
     034056  012746  030656             MOV     #TRAP,-(SP)
     034062  012746  000004             MOV     #VEC4,-(SP)
     034066  012746  000003             MOV     #3,-(SP)
     034072  104437                     TRAP    C$SVEC
     034074  062706  000010             ADD     #10,SP
2822 034100  000240                     NOP
2823 034102  005074  000000             CLR     @TUIP(R4)               ;WRITE THE IP REGISTER
2824 034106  000240                     NOP
2825 034110                             DELAY   1                       ;MAKE SURE TIMEOUT CAN OCCUR
     034110  012727  000001             MOV     #1,(PC)+
     034114  000000                     .WORD   0
     034116  013727  002116             MOV     L$DLY,(PC)+
     034122  000000                     .WORD   0
     034124  005367  177772             DEC     -6(PC)
     034130  001375                     BNE     .-4
     034132  005367  177756             DEC     -22(PC)
     034136  001367                     BNE     .-20
2826
2827 034140  005737  002316             TST     TRP4FG                  ;IF NO TRAP OCCURRED
2828 034144  001416                     BEQ     5$                      ;  THEN CONTINUE TEST
2829 034146  000240                     NOP
2830 034150  012737  025647  002330     MOV     #CTRL,FRUIS             ;IDENTIFY FAILING FRU FOR PRINTOUT
2831 034156                             ERRDF   5,EMSG5,PRIERR          ;"NXM ON READ TUIP"
     034156  104455                     TRAP    C$ERDF
     034160  000005                     .WORD   5
     034162  024140                     .WORD   EMSG5
     034164  027222                     .WORD   PRIERR
2832 034166                             CKLOOP                          ;LOOP ON ERROR?
     034166  104406                     TRAP    C$CLP1
2833 034170                             DODU    LOGUNT                  ;DROP UNIT
     034170  013700  002332             MOV     LOGUNT,R0
     034174  104451                     TRAP    C$DODU
2834 034176                             ESCAPE  SUB                     ;CAN'T CONTINUE
     034176  104410                     TRAP    C$ESCAPE
     034200  000002                     .WORD   L10014-.
2835
2836 034202                             ENDSUB
     034202                     L10014:
     034202  104403                     TRAP    C$ESUB
2837 034204  000240                     NOP
2838 034206                             CLRVEC  #VEC4                   ;RESTORE VECTOR 4
     034206  012700  000004             MOV     #VEC4,R0
     034212  104436                     TRAP    C$CVEC
2839 034214  032764  000001  000014     BIT     #ORPFLG,LUNFLG(R4)      ;IF UNIT WAS NOT DROPPED
2840 034222  001402                     BEQ     T1.2                    ;  THEN CONTINUE TESTING
2841 034224                             ESCAPE  TST                     ;  ELSE LEAVE TEST
     034224  104410                     TRAP    C$ESCAPE
     034226  000264                     .WORD   L10013-.
2842
```

```
2843 034230                                    BGNSUB
     034230                             T1.2:
     034230  104402                            TRAP    C$BSUB
2844 034232  005037  002316             10$:    CLR     TRP4FG                  ;CLEAR NXM ERROR FLAG
2845
2846 034236                                    SETVEC  #VEC4,#TRAP4,#PRI07     ;SET VECTOR 4 FOR NXM TRAPS
     034236  012746  000340                    MOV     #PRI07,-(SP)
     034242  012746  030656                    MOV     #TRAP4,-(SP)
     034246  012746  000004                    MOV     #VEC4,-(SP)
     034252  012746  000003                    MOV     #3,-(SP)
     034256  104437                            TRAP    C$SVEC
     034260  062706  000010                    ADD     #10,SP
2847 034264  000240                            NOP
2848 034266  005774  000002                    TST     @TUSA(R4)               ;READ THE SA REGISTER
2849 034272  000240                            NOP
2850 034274                                    DELAY   25.                     ;WAIT TO ALLOW NXM TRAP
     034274  012727  000031                    MOV     #25.,(PC)+
     034300  000000                            .WORD   0
     034302  013727  002116                    MOV     L$DLY,(PC)+
     034306  000000                            .WORD   0
     034310  005367  177772                    DEC     -6(PC)
     034314  001375                            BNE     .-4
     034316  005367  177756                    DEC     -22(PC)
     034322  001367                            BNE     .-20
2851
2852 034324  005737  002316                    TST     TRP4FG                  ;IF NXM DID NOT OCCUR
2853 034330  001416                            BEQ     15$                     ;  THEN CONTINUE TEST
2854 034332  000240                            NOP
2855 034334  012737  025647  002330            MOV     #CTRL,FRUIS             ;IDENTIFY FAILING FRU FOR PRINTOUT
2856 034342                                    ERRDF   7,EMSG7,PRIERR          ;"NXM ON FIRST READ OF SA"
     034342  104455                            TRAP    C$ERDF
     034344  000007                            .WORD   7
     034346  024212                            .WORD   EMSG7
     034350  027222                            .WORD   PRIERR
2857 034352                                    CKLOOP                          ;LOOP ON ERROR?
     034352  104406                            TRAP    C$CLP1
2858 034354                                    DODU    LOGUNT                  ;DROP UNIT IF NOT
     034354  013700  002332                    MOV     LOGUNT,R0
     034360  104451                            TRAP    C$DODU
2859 034362                                    ESCAPE  SUB                     ;LEAVE TEST
     034362  104410                            TRAP    C$ESCAPE
     034364  000062                            .WORD   L10015-.
2860
2861 034366  017464  000002  000012  15$:     MOV     @TUSA(R4),TUSASV(R4)    ;GET A COPY OF SA IN MEMORY
2862 034374  032764  004000  000012            BIT     #B.S1,TUSASV(R4)        ;IF STEP 1 BIT IS SET
2863 034402  001021                            BNE     16$                     ;  THEN TEST 1 IS COMPLETE
2864 034404  000240                            NOP
2865 034406  012737  004000  002334            MOV     #B.S1,SAEXP             ;LOAD "EXPECTED FOR PRINTOUT
2866 034414  012737  025670  002330            MOV     #LSCT,FRUIS             ;IDENTIFY FAILING FRU FOR PRINTOUT
2867 034422                                    ERRDF   8.,EMSG8,PRISA          ;"SA REG IN ERROR ON FIRST READ"
     034422  104455                            TRAP    C$ERDF
     034424  000010                            .WORD   8
     034426  024233                            .WORD   EMSG8
     034430  027026                            .WORD   PRISA
2868 034432                                    CKLOOP                          ;LOOP ON ERROR?
     034432  104406                            TRAP    C$CLP1
2869 034434                                    DODU    LOGUNT                  ;DROP UNIT IF NOT
```

```
        034434  013700  002332                  MOV     LOGUNT,R0
        034440  104451                          TRAP    C$DODU
  2870  034442                                  ESCAPE  SUB                     ;LEAVE TEST
        034442  104410                          TRAP    C$ESCAPE
        034444  000002                          .WORD   L10015 .
  2871  034446                          16$:    ENDSUB
        034446                          L10015:
        034446  104403                          TRAP    C$ESUB
  2872
  2873  034450  005037  002334          20$:    CLR     SAEXP                   ;CLEAR ERROR INDICATOR
  2874  034454                                  CLRVEC  #VEC4                   ;RESTORE VECTOR 4
        034454  012700  000004                  MOV     #VEC4,R0
        034460  104436                          TRAP    C$CVEC
  2875  034462  032764  000001  000014          BIT     #DRPFLG,LUNFLG(R4)      ;IF UNIT DROPPED
  2876  034470  001006                          BNE     25$                     ;  THEN LEAVE NOW
  2877  034472  005337  000000G                 DEC     ITRCNT                  ;IF ITERATIONS EQUAL 0
  2878  034476  000240                          NOP
  2879  034500  001402                          BEQ     25$                     ;  THEN LEAVE TEST
  2880  034502  000137  034044                  JMP     T1.1                    ;  ELSE GO BACK FOR MORE
  2881
  2882  034506                          25$:    EXIT    TST
        034506  104432                          TRAP    C$EXIT
        034510  000002                          .WORD   L10013-.
  2883
  2884
  2885                                          .EVEN
  2886
  2887  034512                                  ENDTST
        034512                          L10013:
        034512  104401                          TRAP    C$ETST
  2888
```

```
                                        .SBTTL TEST 2: INITIALIZATION TEST (POWER UP MICRODIAGNOSTICS)

                                        ;;*****************************************************************
                                        ;;*****************************************************************
                                        ;
                                        ;TEST 2 - INITIALIZATION TEST (POWER UP MICRODIAGNOSTICS)
                                        ;       THIS TEST COMMENCES STEP 1 OF THE UQ-PORT INITIALIZATION
                                        ;       SEQUENCE WITH INTERRUPTS DISABLED.  AS A RESULT, THE ROM
                                        ;       RESIDENT MICRODIAGNOSTICS WILL BE RUN TO COMPLETION AND
                                        ;       CHECKED FOR ANY ERRORS.
                                        ;
                                        ;;*****************************************************************
                                        ;;*****************************************************************

034514                                          BGNTST
034514                                  T2::
034514  032764  000001  000014                  BIT     #DRPFLG,LUNFLG(R4)      ;IF UUT NOT DROPPED
034522  001402                                  BEQ     1$                      ;  THEN DO TEST
034524                                          EXIT    TST                     ;  ELSE GET OUT
034524  104432                                  TRAP    C$EXIT
034526  000214                                  .WORD   L10016-.
034530  012737  025632  002330  1$:             MOV     #LESI,FRUIS             ;FAILING FRU IN CASE OF ERROR
034536  012737  000001  000000G                 MOV     #1,ITRCNT               ;SET UP FOR ONE TEST ITERATION
034544  022737  000001  002312                  CMP     #1,PASCNT               ;IF FIRST PASS
034552  001403                                  BEQ     2$                      ;  THEN START TEST
034554  012737  000012  000000G                 MOV     #10.,ITRCNT             ;  ELSE DO 10 ITERATIONS

034562  012705  000000          2$:             MOV     #0,R5                   ;SET UP R5 AS INDEX TO STEP TABLES
034566  012737  000001  002336                  MOV     #1,INISTP               ;STEP 1 FOR ERROR PRINTOUT
034574  016437  000004  002272                  MOV     TUVEC(R4),STPTBL        ;PUT VECTOR IN STEP 1
034602  006237  002272                          ASR     STPTBL                  ;DIVIDE BY TWO
034606  006237  002272                          ASR     STPTBL                  ;DIVIDE BY FOUR
034612  013737  002272  002306                  MOV     STPTBL,CMPTBL+4         ;PUT VECTOR IN STEP 3 COMPARE
034620  052737  104400  002272                  BIS     #104400,STPTBL          ;REST OF STEP ONE
034626  012737  005700  002302                  MOV     #B.S1!B.QB!B.DI!B.OD!B.MP,CMPTBL
                                                                                ;STEP 1 COMPARE VALUE
034634  012737  060050  002274                  MOV     #COMMAR,STPTBL+2        ;STEP 2   COMM AREA ADDRESS
034642  012737  010211  002304                  MOV     #010211,CMPTBL+2        ;STEP 2 COMPARE
034650  012737  000000  002276                  MOV     #0,STPTBL+4             ;STEP 3 - HIGH ADDRESS
034656  112737  000040  002307                  MOVB    #40,CMPTBL+5            ;REST OF STEP 3 COMPARE
034664  012737  000000  002300                  MOV     #0,STPTBL+6             ;STEP 4
034672  012737  040000  002310                  MOV     #040000,CMPTBL+6        ;STEP 4 COMPARE

034700  004737  031330                          JSR     PC,STEP1                ;GO DO IT
034704  005737  002340                          TST     STEPST                  ;IF STATUS OKAY
034710  001412                                  BEQ     T2EXT                   ;  THEN DO NEXT TEST

034712                                          ERRDF   9.,EMSG9,PRIINI         ;"SA CONTENTS IN ERROR"
034712  104455                                  TRAP    C$ERDF
034714  000011                                  .WORD   9
034716  024271                                  .WORD   EMSG9
034720  027002                                  .WORD   PRIINI
034722                                          CKLOOP                          ;LOOP ON ERROR?
034722  104406                                  TRAP    C$CLP1
034724                                          DODU    LOGUNT                  ;DROP UUT
```

```
        034724  013700  002332                  MOV     LOGUNT,R0
        034730  104451                          TRAP    C$DODU
   2946 034732                                  ESCAPE  TST                     ;LEAVE TST
        034732  104410                          TRAP    C$ESCAPE
        034734  000006                          .WORD   L10016 .
   2947
   2948 034736                          TSEXT:  EXIT    TST
        034736  104432                          TRAP    C$EXIT
        034740  000002                          .WORD   L10016-.
   2949
   2950 034742                                  ENDTST
        034742                          L10016:
        034742  104401                          TRAP    C$ETST
   2951
```

```
                                        .SBTTL TEST 3: INITIALIZATION TEST

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

                                                BGNTST
034744                                  T3::
034744  032764  000001  000014                  BIT     #DRPFLG,LUNFLG(R4)      ;IF UUT NOT DROPPED
034752  001402                                  BEQ     1$                      ;  THEN DO TEST
034754                                          EXIT    TST                     ;  ELSE GET OUT
034754  104432                                  TRAP    C$EXIT
034756  000402                                  .WORD   L10017-.
034760  012737  000001  000000G 1$:             MOV     #1,ITRCNT               ;SET UP FOR ONE TEST ITERATION
034766  022737  000001  002312                  CMP     #1,PASCNT               ;IF FIRST PASS
034774  001403                                  BEQ     2$                      ;  THEN START TEST
034776  012737  000012  000000G                 MOV     #10.,ITRCNT             ;  ELSE DO 10 ITERATIONS

035004  012705  000000          2$:             MOV     #0,R5                   ;SET UP R5 AS INDEX TO STEP TABLES
035010  012737  000001  002336                  MOV     #1,INISTP               ;STEP 1 FOR ERROR PRINTOUT
035016  016437  000004  002272                  MOV     TUVEC(R4),STPTBL        ;PUT VECTOR IN STEP 1
035024  006237  002272                          ASR     STPTBL                  ;DIVIDE BY TWO
035030  006237  002272                          ASR     STPTBL                  ;DIVIDE BY FOUR
035034  013737  002272  002306                  MOV     STPTBL,CMPTBL+4         ;PUT VECTOR IN STEP 3 COMPARE
035042  052737  104400  002272                  BIS     #104400,STPTBL          ;REST OF STEP ONE
035050  012737  005700  002302                  MOV     #B.S1!B.QB!B.DI!B.OD!B.MP,CMPTBL
                                                                                ;STEP 1 COMPARE VALUE
035056  012737  060050  002274                  MOV     #COMMAR,STPTBL+2        ;STEP 2 - COMM AREA ADDRESS
035064  012737  010211  002304                  MOV     #010211,CMPTBL+2        ;STEP 2 COMPARE
035072  012737  000000  002276                  MOV     #0,STPTBL+4             ;STEP 3 - HIGH ADDRESS
035100  112737  000040  002307                  MOVB    #40,CMPTBL+5            ;REST OF STEP 3 COMPARE
035106  012737  000000  002300                  MOV     #0,STPTBL+6             ;STEP 4
035114  012737  040000  002310                  MOV     #040000,CMPTBL+6        ;STEP 4 COMPARE

035122  004737  031330                          JSR     PC,STEP1                ;GO DO IT
035126  005737  002340                          TST     STEPST                  ;IF STATUS OKAY
035132  001415                                  BEQ     5$                      ;  THEN CONTINUE TEST

035134  012737  025632  002330                  MOV     #LESI,FRUIS             ;FAILING FRU IN CASE OF ERROR
035142                                          ERRDF   9.,EMSG9,PRIINI         ;"SA CONTENTS IN ERROR"
035142  104455                                  TRAP    C$ERDF
035144  000011                                  .WORD   9
035146  024271                                  .WORD   EMSG9
035150  027002                                  .WORD   PRIINI
035152                                          CKLOOP                          ;LOOP ON ERROR?
```

```
                035152  104406                          TRAP    C$CLP1
        3010    035154                                  DODU    LOGUNT                  ;DROP UUT
                035154  013700  002332                  MOV     LOGUNT,R0
                035160  104451                          TRAP    C$DODU
        3011    035162                                  ESCAPE  TST                     ;LEAVE TST
                035162  104410                          TRAP    C$ESCAPE
                035164  000174                          .WORD   L10017-.
        3012
        3013    035166  005237  002336          5$:     INC     INISTP                  ;ADJUST STEP COUNTER
        3014    035172  062705  000002                  ADD     #2,R5                   ;ADJUST TABLE INDEX
        3015    035176  012737  000100  002346  6$:     MOV     #100,OUTER              ;SET UP FOR DELAY ROUTINE
        3016    035204  016537  002302  002334          MOV     CMPTBL(R5),SAEXP        ;SET UP FOR COMPARE
        3017    035212  012737  037200  002344  7$:     MOV     #16000.,INNER           ;SET UP INNER
        3018    035220  017464  000002  000012          MOV     @TUSA(R4),TUSASV(R4)    ;GET SA CONTENTS
        3019    035226  022705  000006                  CMP     #6,R5                   ;ARE WE IN STEP 4?
        3020    035232  001005                          BNE     8$                      ;BRANCH IF NOT
        3021    035234  033764  002334  000012          BIT     SAEXP,TUSASV(R4)        ;JUST LOOK FOR STEP 4 BIT
        3022    035242  001027                          BNE     10$                     ;IT'S SET SO LET'S GO
        3023    035244  000404                          BR      9$                      ;STAY IN LOOP OTHERWISE
        3024    035246  C23764  002334  000012  8$:     CMP     SAEXP,TUSASV(R4)        ;IF SA IS WHAT WE EXPECT
        3025    035254  001422                          BEQ     10$                     ;   THEN MOVE ALONG
        3026    035256  004737  031302          9$:     JSR     PC,PDELAY               ;   ELSE GIVE UUT SOME TIME
        3027    035262  005737  002350                  TST     TOUT                    ;IF NO TIMEOUT YET
        3028    035266  001751                          BEQ     7$                      ;   THEN GO TAKE ANOTHER LOOK
        3029
        3030    035270  012737  025670  002330          MOV     #LSCT,FRUIS             ;FAILING FRU IN CASE OF ERROR
        3031    035276                                  ERRDF   13.,EMSG9,PRIINI        ;"SA CONTENTS IN ERROR"
                035276  104455                          TRAP    C$ERDF
                035300  000015                          .WORD   13
                035302  024271                          .WORD   EMSG9
                035304  027002                          .WORD   PRIINI
        3032    035306                                  CKLOOP
                035306  104406                          TRAP    C$CLP1
        3033    035310                                  DODU    LOGUNT
                035310  013700  002332                  MOV     LOGUNT,R0
                035314  104451                          TRAP    C$DODU
        3034    035316                                  ESCAPE  TST
                035316  104410                          TRAP    C$ESCAPE
                035320  000040                          .WORD   L10017-.
        3035
        3036    035322  016574  002272  000002  10$:    MOV     STPTBL(R5),@TUSA(R4)    ;WRITE NEXT STEP TO UUT
        3037    035330  022705  000006                  CMP     #6,R5                   ;IF NOT IN STEP 4
        3038    035334  001314                          BNE     5$                      ;GO BACK TO MAIN LOOP
        3039
        3040    035336  032764  000001  000014          BIT     #DRPFLG,LUNFLG(R4)      ;HAS UUT BEEN DROPPED
        3041    035344  001003                          BNE     T3EXT                   ;LEAVE NOW IF SO
        3042    035346  005337  000000G                 DEC     ITRCNT                  ;IF MORE ITERATIONS LEFT
        3043    035352  001214                          BNE     2$                      ;   THEN GO DO IT AGAIN
        3044
        3045    035354                          T3EXT:  EXIT    TST
                035354  104432                          TRAP    C$EXIT
                035356  000002                          .WORD   L10017-.
        3046
        3047    035360                                  ENDTST
                035360                          L10017:
                035360  104401                          TRAP    C$ETST
```

```
3050                                    .SBTTL TEST 4: SA REGISTER WRAP TEST
3054
3055                                    ;;*****************************************************************
3056                                    ;;*****************************************************************
3057                                    ;
3058                                    ;TEST 4 - SA REGISTER WRAP TEST
3059                                    ;       THIS TEST WILL INITIALIZE THE UUT BY WRITING TO ITS
3060                                    ;       IP REGISTER.  IT WILL FORCE THE UUT INTO DIAGNOSTIC
3061                                    ;       WRAP MODE, AND WRITE FIRST A FLOATING 0 DATA PATTERN,
3062                                    ;       FOLLOWED BY A FLOATING 1 DATA PATTERN TO THE SA REG.
3063                                    ;       EACH WRITE WILL BE FOLLOWED BY A READ AND COMPARE
3064                                    ;       OPERATION.
3065                                    ;
3066                                    ;;*****************************************************************
3067                                    ;;*****************************************************************
3071 035362                                    BGNTST
     035362                             T4::
3072
3073 035362  004737  030734                    JSR     PC,CHKCAC
3074 035366  032764  000001  000014            BIT     #DRPFLG,LUNFLG(R4)      ;IF UUT NOT DROPPED
3075 035374  001402                            BEQ     1$                      ;  THEN DO TEST
3076 035376                                    EXIT    TST                     ;  ELSE GET OUT
     035376  104432                            TRAP    C$EXIT
     035400  000522                            .WORD   L10020-.
3077 035402  012737  000001  002336  1$:      MOV     #1,INISTP               ;STEP 1 FOR ERROR PRINTOUT
3078 035410  012737  000001  000000G           MOV     #1,ITRCNT               ;SET UP FOR ONE TEST ITERATION
3079 035416  022737  000001  002312            CMP     #1,PASCNT               ;IF FIRST PASS
3080 035424  001403                            BEQ     2$                      ;  THEN START TEST
3081 035426  012737  000002  000000G           MOV     #2,ITRCNT               ;  ELSE DO 2 ITERATIONS
3082
3083 035434  012737  140000  002334  2$:      MOV     #BIT15!B.WR,SAEXP       ;SET UP STEP 1 FOR DIAG. WRAP MODE
3084 035442  013737  002334  002272            MOV     SAEXP,STPTBL            ;PUT IT IN STEP 1 OF TABLE
3085 035450  004737  031330                    JSR     PC,STEP1                ;GO DO IT
3086
3087 035454  005737  002340                    TST     STEPST                  ;IF STATUS OKAY
3088 035460  001415                            BEQ     5$                      ;  THEN CONTINUE TEST
3089
3090 035462  012737  025647  002330            MOV     #CTRL,FRUIS             ;FAILING FRU FOR PRINTOUT
3091 035470                                    ERRDF   9.,EMSG9,PRIINI         ;"SA CONTENTS IN ERROR"
     035470  104455                            TRAP    C$ERDF
     035472  000011                            .WORD   9
     035474  024271                            .WORD   EMSG9
     035476  027002                            .WORD   PRIINI
3092 035500                                    CKLOOP                          ;LOOP ON ERROR?
     035500  104406                            TRAP    C$CLP1
3093 035502                                    DODU    LOGUNT                  ;DROP UUT
     035502  013700  002332                    MOV     LOGUNT,R0
     035506  104451                            TRAP    C$DODU
3094 035510                                    ESCAPE  TST                     ;LEAVE TST
     035510  104410                            TRAP    C$ESCAPE
     035512  000410                            .WORD   L10020-.
3095
3096 035514  012737  000100  002346  5$:      MOV     #100,OUTER              ;SET UP FOR DELAY ROUTINE
3097 035522  012737  006000  002344  6$:      MOV     #6000,INNER             ;SET UP INNER
3098 035530  017464  000002  000012            MOV     @TUSA(R4),TUSASV(R4)    ;GET SA CONTENTS
3099 035536  023764  002334  000012            CMP     SAEXP,TUSASV(R4)        ;IF SA IS WHAT WE EXPECT
3100 035544  001422                            BEQ     10$                     ;  THEN MOVE ALONG
```

```
3101 035546  004737  031302                    JSR     PC,PDELAY               ;  ELSE GIVE UUT SOME TIME
3102 035552  005737  002350                    TST     TOUT                    ;IF NO TIMEOUT YET
3103 035556  001761                            BEQ     6$                      ;  THEN GO TAKE ANOTHER LOOK
3104
3105 035560  012737  025647  002330            MOV     #CTRL,FRUIS             ;FAILING FRU FOR PRINTOUT
3106 035566                                    ERRDF   10.,EMSG9,PRIINI        ;"SA CONTENTS IN ERROR"
     035566  104455                            TRAP    C$ERDF
     035570  000012                             .WORD   10
     035572  024271                            .WORD   EMSG9
     035574  027002                            .WORD   PRIINI
3107 035576                                    CKLOOP
     035576  104406                            TRAP    C$CLP1
3108 035600                                    DODU    LOGUNT
     035600  013700  002332                    MOV     LOGUNT,R0
     035604  104451                            TRAP    C$DODU
3109 035606                                    ESCAPE  TST
     035606  104410                            TRAP    C$ESCAPE
     035610  000312                            .WORD   L10020-.
3110
3111 035612  000261                    10$:    SEC                             ;SET CARRY BIT
3112 035614  012737  177776  002342            MOV     #177776,WRDATA          ;SET UP FLOATING "0" PATTERN
3113 035622  013774  002342  000002    11$:    MOV     WRDATA,@TUSA(R4)        ;SEND DATA TO UUT
3114 035630  013737  002342  002334            MOV     WRDATA,SAEXP            ;SAVE A COPY FOR COMPARE
3115 035636  012737  000100  002346            MOV     #100,OUTER              ;SET UP FOR DELAY ROUTINE
3116
3117 035644  012737  006000  002344    15$:    MOV     #6000,INNER             ;INNER TOO
3118 035652  017464  000002  000012            MOV     @TUSA(R4),TUSASV(R4)    ;READ SA
3119 035660  023764  002334  000012            CMP     SAEXP,TUSASV(R4)        ;IF DATA MATCHES
3120 035666  001422                            BEQ     20$                     ;  THEN CHANGE DATA
3121 035670  004737  031302                    JSR     PC,PDELAY               ;  ELSE GIVE UUT SOME TIME
3122 035674  005737  002350                    TST     TOUT                    ;IF NO TIMEOUT YET
3123 035700  001761                            BEQ     15$                     ;  THEN GO TAKE ANOTHER LOOK
3124
3125 035702  012737  025647  002330            MOV     #CTRL,FRUIS             ;FAILING FRU FOR PRINTOUT
3126 035710                                    ERRDF   11.,EMSG10,PRIINI       ;"SA WRONG IN DATA WRAP"
     035710  104455                            TRAP    C$ERDF
     035712  000013                            .WORD   11
     035714  024316                            .WORD   EMSG10
     035716  027002                            .WORD   PRIINI
3127 035720                                    CKLOOP
     035720  104406                            TRAP    C$CLP1
3128 035722                                    DODU    LOGUNT
     035722  013700  002332                    MOV     LOGUNT,R0
     035726  104451                            TRAP    C$DODU
3129 035730                                    ESCAPE  TST                     ;GET OUT IF NOT LOOPING
     035730  104410                            TRAP    C$ESCAPE
     035732  000170                            .WORD   L10020-.
3130
3131 035734  006137  002342            20$:    ROL     WRDATA                  ;SHIFT TEST PATTERN
3132 035740  103730                            BCS     11$                     ;WE'RE NOT DONE YET
3133
3134 035742  012737  000001  002342            MOV     #1,WRDATA               ;SET UP FOR FLOATING 1 PATTERN
3135 035750  013774  002342  000002    24$:    MOV     WRDATA,@TUSA(R4)        ;SEND DATA TO UUT
3136 035756  013737  002342  002334            MOV     WRDATA,SAEXP            ;KEEP A COPY FOR COMPARE
3137 035764  012737  000100  002346            MOV     #100,OUTER              ;SET UP FOR DELAY ROUTINE
3138
3139 035772  012737  006000  002344    25$:    MOV     #6000,INNER             ;DELAY ROUTINE TOO
```

TEST 4: SA REGISTER WRAP TEST

```
3140 036000  017464  000002  000012          MOV     @TUSA(R4),TUSASV(R4)    ;READ THE SA
3141 036006  023764  002334  000012          CMP     SAEXP,TUSASV(R4)        ;IF IT MATCHES
3142 036014  001422                          BEQ     30$                     ;  THEN SEE IF WE'RE DONE
3143 036016  004737  031302                  JSR     PC,PDELAY               ;  ELSE GIVE UUT SOME MORE TIME
3144 036022  005737  002350                  TST     TOUT                    ;IF NO TIMEOUT YET
3145 036026  001761                          BEQ     25$                     ;  THEN TAKE ANOTHER LOOK
3146
3147 036030  012737  025647  002330          MOV     #CTRL,FRUIS             ;FAILING FRU FOR PRINTOUT
3148 036036                                  ERRDF   12.,EMSG10,PRIINI       ;"SA WRONG IN DATA WRAP"
     036036  104455                          TRAP    C$ERDF
     036040  000014                          .WORD   12
     036042  024316                          .WORD   EMSG10
     036044  027002                          .WORD   PRIINI
3149 036046                                  CKLOOP
     036046  104406                          TRAP    C$CLP1
3150 036050                                  DODU    LOGUNT
     036050  013700  002332                  MOV     LOGUNT,R0
     036054  104451                          TRAP    C$DODU
3151 036056                                  ESCAPE  TST                     ;LEAVE TEST IF NOT LOOPING
     036056  104410                          TRAP    C$ESCAPE
     036060  000042                          .WORD   L10020-.
3152
3153 036062  006137  002342          30$:    ROL     WRDATA                  ;SHIFT DATA PATTERN
3154 036066  103330                          BCC     24$                     ;WE'RE NOT DONE YET
3155 036070  005337  000000G                 DEC     ITRCNT                  ;IF ITERATIONS  = 0
3156 036074  001402                          BEQ     T4EXT                   ;  THEN LEAVE TEST
3157 036076  000137  035434                  JMP     2$                      ;  ELSE DO ANOTHER ONE
3158
3159 036102  005737  000000G         T4EXT:  TST     CPFLG                   ;CHECK IF CACHE WAS DISABLED
3160 036106  001403                          BEQ     EXT                     ;NO, BRANCH
3161 036110  042737  000014  177746          BIC     #DISCAC,CCR             ;RE-ENABLE CACHE
3162 036116                          EXT:    EXIT    TST                     ;GET OUTTA HERE
     036116  104432                          TRAP    C$EXIT
     036120  000002                          .WORD   L10020-.
3163
3164 036122                                  ENDTST
     036122                          L10020:
     036122  104401                          TRAP    C$ETST
```

```
                                .SBTTL  TEST 5:
                                .SBTTL  SUBTEST 1: VECTOR AND INTERRUPT TEST

                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;
                                ;TEST 5
                                ;SUBTEST 1 -
                                ;       VECTOR AND INTERRUPT TEST
                                ;       TEST 3 IS REPEATED, BUT WITH INTERRUPTS ENABLED.
                                ;       THE PROGRAM VERIFIES THAT AN INTERRUPT DCCURS AT
                                ;       THE END OF STEPS 1 - 3.
                                ;
                                ;;**********************************************************************
                                ;;**********************************************************************

036124                                  BGNTST
036124                          T5::
036124                                  BGNSUB
036124                          T5.1:
036124  104402                          TRAP    C$BSUB

036126  032764  000001  000014          BIT     @DRPFLG,LUNFLG(R4)      ;IF UUT NOT DROPPED
036134  001402                          BEQ     1$                      ;  THEN DO TEST
036136                                  EXIT    TST                     ;  ELSE GET OUT
036136  104432                          TRAP    C$EXIT
036140  001114                          .WORD   L10021-.
036142  042764  000004  000014  1$:     BIC     @BRFLAG,LUNFLG(R4)      ;DO TEST WITH PRIORITY SET TO 0
036150  012737  025647  002330          MOV     @CTRL,FRUIS             ;FAILING FRU IN CASE OF ERROR
036156  012737  000001  000000G         MOV     #1,ITRCNT               ;SET UP FOR ONE TEST ITERATION
036164  022737  000001  002312          CMP     #1,PASCNT               ;IF FIRST PASS
036172  001403                          BEQ     2$                      ;  THEN START TEST
036174  012737  000012  000000G         MOV     #10.,ITRCNT             ;  ELSE DO 10 ITERATIONS

036202  004737  031212          2$:     JSR     PC,VECTOR               ;SET UP VECTOR WITH INTERRUPT HANDLER
036206  012705  000000                  MOV     #0,R5                   ;SET UP R5 AS INDEX TO STEP TABLES
036212  012737  000001  002336          MOV     #1,INISTP               ;STEP 1 FOR ERROR PRINTOUT
036220  016437  000004  002272          MOV     TUVEC(R4),STPTBL        ;PUT VECTOR IN STEP 1
036226  006237  002272                  ASR     STPTBL                  ;DIVIDE BY TWO
036232  006237  002272                  ASR     STPTBL                  ;DIVIDE BY FOUR
036236  013737  002272  002306          MOV     STPTBL,CMPTBL+4         ;PUT VECTOR IN STEP 3 COMPARE
036244  052737  104600  002272          BIS     #104600,STPTBL          ;REST OF STEP ONE
036252  012737  005700  002302          MOV     #B.S1!B.QB!B.DI!B.OD!B.MP,CMPTBL
                                                                        ;STEP 1 COMPARE VALUE
036260  012737  060050  002274          MOV     #COMMAR,STPTBL+2        ;STEP 2 - COMM AREA ADDRESS
036266  012737  010211  002304          MOV     #010211,CMPTBL+2        ;STEP 2 COMPARE
036274  012737  000000  002276          MOV     #0,STPTBL+4             ;STEP 3 - HIGH ADDRESS
036302  052737  000200  002306          BIS     #B.IE,CMPTBL+4          ;SET THE INTERRUPT ENABLE BIT
036310  112737  000040  002307          MOVB    #40,CMPTBL+5            ;REST OF STEP 3 COMPARE
036316  012737  000000  002300          MOV     #0,STPTBL+6             ;STEP 4
036324  012737  040000  002310          MOV     #040000 CMPTBL+6        ;STEP 4 COMPARE

036332  004737  031330                  JSR     PC,STEP1                ;GO DO IT
036336  005737  002340                  TST     STEPST                  ;IF STATUS OKAY
036342  001412                          BEQ     5$                      ;  THEN CONTINUE TEST

036344                                  ERRDF   14.,EMSG9,PRIINI        ;"SA CONTENTS IN ERROR"
```

```
                036344  104455                          TRAP    C$ERDF
                036346  000016                          .WORD   14
                036350  024271                          .WORD   EMSG9
                036352  027002                          .WORD   PRIINI
          3225  036354                                  CKLOOP                          ;LOOP ON ERROR?
                036354  104406                          TRAP    C$CLP1
          3226  036356                                  DODU    LOGUNT                  ;DROP UUT
                036356  013700  002332                  MOV     LOGUNT,R0
                036362  104451                          TRAP    C$DODU
          3227  036364                                  ESCAPE  TST                     ;LEAVE TST
                036364  104410                          TRAP    C$ESCAPE
                036366  000666                          .WORD   L10021-.
          3228
          3229  036370  012737  000100  002346  5$:     MOV     #100,OUTER              ;SET UP FOR DELAY ROUTINE
          3230  036376  016537  002302  002334          MOV     CMPTBL(R5),SAEXP        ;SET UP FOR COMPARE
          3231  036404  012737  037200  002344  7$:     MOV     #16000.,INNER           ;SET UP INNER
          3232  036412  032764  000002  000014          BIT     #INTFLG,LUNFLG(R4)      ;IF INTERRUPT OCCURRED
          3233  036420  001022                          BNE     10$                     ;  THEN SEE IF SA IS CORRECT
          3234  036422  004737  031302          9$:     JSR     PC,PDELAY               ;  ELSE GIVE UUT SOME TIME
          3235  036426  005737  002350                  TST     TOUT                    ;IF NO TIMEOUT YET
          3236  036432  001764                          BEQ     7$                      ;  THEN GO TAKE ANOTHER LOOK
          3237
          3238  036434  012737  025632  002330          MOV     #LESI,FRUIS             ;FAILING FRU
          3239  036442                                  ERRDF   15.,EMSG11,PRIERR       ;"EXPECTED INTERRUPT DID NOT OCCUR"
                036442  104455                          TRAP    C$ERDF
                036444  000017                          .WORD   15
                036446  024344                          .WORD   EMSG11
                036450  027222                          .WORD   PRIERR
          3240  036452                                  CKLOOP
                036452  104406                          TRAP    C$CLP1
          3241  036454                                  DODU    LOGUNT
                036454  013700  002332                  MOV     LOGUNT,R0
                036460  104451                          TRAP    C$DODU
          3242  036462                                  ESCAPE  TST
                036462  104410                          TRAP    C$ESCAPE
                036464  000570                          .WORD   L10021-.
          3243
          3244  036466  042764  000002  000014  10$:    BIC     #INTFLG,LUNFLG(R4)      ;CLEAR THE INTERRUPT FLAG
          3245  036474  005237  002336                  INC     INISTP                  ;ADJUST THE STEP COUNTER
          3246  036500  062705  000002                  ADD     #2,R5                   ;ADJUST TABLE INDEX
          3247  036504  016537  002302  002334          MOV     CMPTBL(R5),SAEXP        ;GET THE COMPARISON VALUE
          3248  036512  017464  000002  000012          MOV     @TUSA(R4),TUSASV(R4)    ;GET SA CONTENTS
          3249  036520  022705  000006                  CMP     #6,R5                   ;ARE WE IN STEP 4?
          3250  036524  001005                          BNE     15$                     ;BRANCH IF NOT
          3251  036526  033764  002334  000012          BIT     SAEXP,TUSASV(R4)        ;JUST LOOK FOR STEP 4 BIT
          3252  036534  001022                          BNE     20$                     ;IT'S SET SO LET'S GO
          3253  036536  000407                          BR      16$                     ;ERROR
          3254  036540  023764  002334  000012  15$:    CMP     SAEXP,TUSASV(R4)        ;IF SA IS WHAT WE EXPECT
          3255  036546  001415                          BEQ     20$                     ;  THEN MOVE ALONG
          3256
          3257  036550  012737  025632  002330          MOV     #LESI,FRUIS             ;FAILING FRU
          3258  036556                          16$:    ERRDF   16.,EMSG9,PRIINI                ;"SA CONTENTS IN ERROR"
                036556  104455                          TRAP    C$ERDF
                036560  000020                          .WORD   16
                036562  024271                          .WORD   EMSG9
                036564  027002                          .WORD   PRIINI
          3259  036566                                  CKLOOP
```

```
                036566  104406                          TRAP    C$CLP1
        3260    036570                                  DODU    LOGUNT
                036570  013700  002332                  MOV     LOGUNT,R0
                036574  104451                          TRAP    C$DODU
        3261    036576                                  ESCAPE  TST
                036576  104410                          TRAP    C$ESCAPE
                036600  000454                          .WORD   L10021-.
        3262
        3263    036602  016574  002272  000002  20$:    MOV     STPTBL(R5),@TUSA(R4)    ;WRITE NEXT STEP TO UUT
        3264    036610  022705  000006                  CMP     #6,R5                   ;IF NOT IN STEP 4
        3265    036614  001265                          BNE     5$                      ;GO BACK TO MAIN LOOP
        3266
        3267    036616  032764  000001  000014          BIT     #DRPFLG,LUNFLG(R4)      ;HAS UUT BEEN DROPPED
        3268    036624  001005                          BNE     T5EXT                   ;LEAVE NOW IF SO
        3269    036626  005337  000000G                 DEC     ITRCNT                  ;IF NO MORE ITERATIONS LEFT
        3270    036632  001402                          BEQ     T5EXT                   ;   THEN EXIT
        3271    036634  000137  036202                  JMP     2$                      ;  ELSE DO IT AGAIN
        3272
        3273    036640  004737  031162          T5EXT:  JSR     PC,RSTVEC               ;CATCH ILLEGAL INTERRUPTS
        3274    036644                                  EXIT    TST
                036644  104432                          TRAP    C$EXIT
                036646  000406                          .WORD   L10021-.
        3275    036650                                  ENDSUB
                036650                          L10022:
                036650  104403                          TRAP    C$ESUB
```

```
                                        .SBTTL SUBTEST 2: BR LEVEL TEST
                                        ;;**********************************************************************
                                        ;;**********************************************************************
                                        ;
                                        ;SUBTEST 2 -
                                        ;       BR LEVEL TEST
                                        ;       THIS TEST INSURES THAT THE TU81 CAN NOT INTERRUPT
                                        ;       WHEN THE CPU PRIORITY IS SET TO 7.  THE TEST GOES
                                        ;       ONLY THROUGH THE FIRST STEP OF THE INIT SEQUENCE
                                        ;       SINCE THE CONTROLLER WILL "HANG" WAITING FOR THE
                                        ;       INTERRUPT ACKNOWLEDGE.
                                        ;
                                        ;;**********************************************************************
                                        ;;**********************************************************************

                                                BGNSUB
                                        T5.2:
036652  104402                                  TRAP    C$BSUB

036654  032764  000001  000014                  BIT     #DRPFLG,LUNFLG(R4)      ;IF UUT NOT DROPPED
036662  001402                                  BEQ     1$                      ;   THEN DO TEST
036664                                          EXIT    TST                     ;   ELSE GET OUT
036664  104432                                  TRAP    C$EXIT
036666  000366                                  .WORD   L10021-.
036670                                  1$:
036670  052764  000004  000014                  BIS     #BRFLAG,LUNFLG(R4)      ;DO TEST WITH HIGH PRIORITY
036676  012737  025647  002330                  MOV     #CTRL,FRUIS             ;FAILING FRU IN CASE OF ERROR
036704  012737  000001  000000G                 MOV     #1,ITRCNT               ;SET UP FOR ONE TEST ITERATION
036712  022737  000001  002312                  CMP     #1,PASCNT               ;IF FIRST PASS
036720  001403                                  BEQ     2$                      ;   THEN START TEST
036722  012737  000002  000000G                 MOV     #2,ITRCNT               ;   ELSE DO 10 ITERATIONS

036730  106427  000340                  2$:     MTPS    #PRIO7                  ;CPU PRIORITY = 7
036734  004737  031212                          JSR     PC,VECTOR               ;SET UP VECTOR WITH INTERRUPT HANDLER
036740  012705  000000                          MOV     #0,R5                   ;SET UP R5 AS INDEX TO STEP TABLES
036744  012737  000001  002336                  MOV     #1,INISTP               ;STEP 1 FOR ERROR PRINTOUT
036752  016437  000004  002272                  MOV     TUVEC(R4),STPTBL        ;PUT VECTOR IN STEP 1
036760  006237  002272                          ASR     STPTBL                  ;DIVIDE BY TWO
036764  006237  002272                          ASR     STPTBL                  ;DIVIDE BY FOUR
036770  052737  104600  002272                  BIS     #104600,STPTBL          ;REST OF STEP ONE
036776  016437  000004  002302                  MOV     TUVEC(R4),CMPTBL        ;STEP 1 COMPARE VALUE

037004  004737  031330                          JSR     PC,STEP1                ;GO DO IT
037010  005737  002340                          TST     STEPST                  ;IF STATUS OKAY
037014  001412                                  BEQ     5$                      ;   THEN CONTINUE TEST

037016                                          ERRDF   14.,EMSG9,PRIINI        ;"SA CONTENTS IN ERROR"
037016  104455                                  TRAP    C$ERDF
037020  000016                                  .WORD   14
037022  024271                                  .WORD   EMSG9
037024  027002                                  .WORD   PRIINI
037026                                          CKLOOP                          ;LOOP ON ERROR?
037026  104406                                  TRAP    C$CLP1
037030                                          DODU    LOGUNT                  ;DROP UUT
037030  013700  002332                          MOV     LOGUNT,R0
037034  104451                                  TRAP    C$DODU
```

```
3330 037036                                        ESCAPE  TST                     ;LEAVE TST
     037036  104410                                TRAP    C$ESCAPE
     037040  000214                                .WORD   L10021-.
3331
3332 037042  012737  000100  002346  5$:          MOV     #100,OUTER              ;SET UP FOR DELAY ROUTINE
3333 037050  016537  002302  002334                MOV     CMPTBL(R5),SAEXP        ;SET UP FOR COMPARE
3334 037056  012737  037200  002344  7$:          MOV     #16000.,INNER           ;SET UP INNER
3335 037064  004737  031302          9$:          JSR     PC,PDELAY               ;  ELSE GIVE UUT SOME TIME
3336 037070  005737  002350                      TST     TOUT                    ;IF NO TIMEOUT YET
3337 037074  001770                              BEQ     7$                      ;  THEN GO TAKE ANOTHER LOOK
3338
3339 037076  017464  000002  000012              MOV     @TUSA(R4),TUSASV(R4)    ;GET SA CONTENTS
3340 037104  023764  002334  000012              CMP     SAEXP,TUSASV(R4)        ;IF CONTENTS OKAY
3341 037112  001412                              BEQ     10$                     ;  THEN CHECK FOR INTERRUPT
3342
3343 037114                                        ERRDF   17.,EMSG9,PRIINI        ;"SA CONTENTS IN ERROR'
     037114  104455                                TRAP    C$ERDF
     037116  000021                                .WORD   17
     037120  024271                                .WORD   EMSG9
     037122  027002                                .WORD   PRIINI
3344 037124                                        CKLOOP
     037124  104406                                TRAP    C$CLP1
3345 037126                                        DODU    LOGUNT
     037126  013700  002332                        MOV     LOGUNT,R0
     037132  104451                                TRAP    C$DODU
3346 037134                                        ESCAPE  TST
     037134  104410                                TRAP    C$ESCAPE
     037136  000116                                .WORD   L10021-.
3347
3348 037140  032764  000002  000014  10$:          BIT     #INTFLG,LUNFLG(R4)      ;IF NO INTERRUPT OCCURRED
3349 037146  001415                              BEQ     20$                     ;  THEN CARRY ON WITH TEST
3350 037150  042764  000002  000014              BIC     #INTFLG,LUNFLG(R4)      ;CLEAR FLAG IN CASE WE'RE LOOPING
3351 037156                                        ERRDF   18.,EMSG12,PRIINI       ;"INTRRPT WITH CPU PRIORITY =7"
     037156  104455                                TRAP    C$ERDF
     037160  000022                                .WORD   18
     037162  024405                                .WORD   EMSG12
     037164  027002                                .WORD   PRIINI
3352 037166                                        CKLOOP
     037166  104406                                TRAP    C$CLP1
3353 037170                                        DODU    LOGUNT
     037170  013700  002332                        MOV     LOGUNT,R0
     037174  104451                                TRAP    C$DODU
3354 037176                                        ESCAPE  TST
     037176  104410                                TRAP    C$ESCAPE
     037200  000054                                .WORD   L10021-.
3355
3356 037202  106427  000000          20$:          MTPS    #PRI00                  ;CPU PRIORITY = 0
3357 037206  000240                              NOP
3358 037210  000240                              NOP                             ;DELAY FOR PENDING INTERRUPT
3359 037212  042764  000002  000014              BIC     #INTFLG,LUNFLG(R4)      ;CLEAR THE FLAG NOW
3360
3361 037220  032764  000001  000014              BIT     #DRPFLG,LUNFLG(R4)      ;HAS UUT BEEN DROPPED
3362 037226  001005                              BNE     ST5EXT                  ;LEAVE NOW IF SO
3363 037230  005337  000000G                     DEC     ITRCNT                  ;IF NO MORE ITERATIONS LEFT
3364 037234  001402                              BEQ     ST5EXT                  ;  THEN EXIT
3365 037236  000137  036730                      JMP     2$                      ;  ELSE DO IT AGAIN
3366
```

```
 3367 037242  004737  031162          ST5EXT: JSR     PC,RSTVEC               ;CATCH ILLEGAL INTERRUPTS
 3368 037246                                  EXIT    TST
      037246  104432                          TRAP    C$EXIT
      037250  000004                          .WORD   L10021-.
 3369
 3370 037252                                  ENDSUB
      037252                          L10023:
      037252  104403                          TRAP    C$ESUB
 3371
 3372 037254                                  ENDTST
      037254                          L10021:
      037254  104401                          TRAP    C$ETST
```

```
3375                                    .SBTTL TEST 6:
3376                                    .SBTTL SUBTEST 1: PURGE AND POLL TEST
3380
3381                                    ;;*****************************************************************
3382                                    ;;*****************************************************************
3383                                    ;
3384                                    ;SUBTEST 6 - PURGE AND POLL TEST
3385                                    ;       THIS TEST WILL AGAIN RUN THROUGH THE INIT SEQUENCE. THIS
3386                                    ;       TIME SETTING THE "PURGE AND POLL" BIT IN STEP 3.  THIS
3387                                    ;       SHOULD CAUSE THE PORT TO DMA VARIOUS DATA PATTERNS TO
3388                                    ;       AND FROM THE COMMUNICATIONS AREA AND FINALLY LEAVE IT
3389                                    ;       CLEARED BEFORE TRANSITIONING TO STEP 4.  THE PROGRAM WILL
3390                                    ;       HAVE FILLED THIS AREA WITH A BACKGROUND PATTERN OF ALL
3391                                    ;       1'S DATA PRIOR TO STARTING THE INIT.  WHEN STEP 4 IS
3392                                    ;       REACHED, THE PROGRAM WILL VERIFY THAT THE COMM AREA IS
3393                                    ;       ALL 0'S, AND THAT THE 20 WORDS PRECEDING AND SUCCEEDING
3394                                    ;       THE COMM AREA ARE UNTOUCHED.
3395                                    ;
3396                                    ;;*****************************************************************
3397                                    ;;*****************************************************************
3401
3402 037256                            BGNTST
      037256                    T6::
3403 037256                            BGNSUB
      037256                    T6.1:
      037256  104402                   TRAP    C$BSUB
3404
3405 037260  032764  000001  000014    BIT     #ORPFLG,LUNFLG(R4)      ;IF UUT NOT DROPPED
3406 037266  001402                    BEQ     1$                      ;  THEN DO TEST
3407 037270                            EXIT    TST                     ;  ELSE GET OUT
      037270  104432                   TRAP    C$EXIT
      037272  001406                   .WORD   L10024-.
3408 037274  012737  025647  002330  1$:     MOV     #CTRL,FRUIS             ;FAILING FRU IN CASE OF ERROR
3409 037302  012737  000001  000000G       MOV     #1,ITRCNT               ;SET UP FOR ONE TEST ITERATION
3410 037310  022737  000001  002312        CMP     #1,PASCNT               ;IF FIRST PASS
3411 037316  001403                    BEQ     2$                      ;   THEN START TEST
3412 037320  012737  000012  000000G       MOV     #10.,ITRCNT             ;  ELSE DO 10 ITERATIONS
3413
3414 037326  012705  000000            2$:     MOV     #0,R5                   ;SET UP R5 AS INDEX TO STEP TABLES
3415 037332  012737  000001  002336        MOV     #1,INISTP               ;STEP 1 FOR ERROR PRINTOUT
3416 037340  016437  000004  002272        MOV     TUVEC(R4),STPTBL        ;PUT VECTOR IN STEP 1
3417 037346  006237  002272                ASR     STPTBL                  ;DIVIDE BY TWO
3418 037352  006237  002272                ASR     STPTBL                  ;DIVIDE BY FOUR
3419 037356  013737  002272  002306        MOV     STPTBL,CMPTBL+4         ;PUT VECTOR IN STEP 3 COMPARE
3420 037364  052737  111000  002272        BIS     #111000,STPTBL          ;RES1 OF STEP ONE
3421 037372  012737  005700  002302        MOV     #B.S1!B.QB!B.DI!B.OD!B.MP,CMPTBL
3422                                                                    ;STEP 1 COMPARE VALUE
3423 037400  012737  060050  002274        MOV     #COMMAR,STPTBL+2        ;STEP 2 - COMM AREA ADDRESS
3424 037406  012737  010222  002304        MOV     #010222,CMPTBL+2        ;STEP 2 COMPARE
3425 037414  012737  100000  002276        MOV     #B.PP,STPTBL+4          ;STEP 3   HIGH ADDRESS AND PRGE/POLL
3426 037422  112737  000040  002307        MOVB    #40,CMPTBL+5            ;REST OF STEP 3 COMPARE
3427 037430  012737  000000  002300        MOV     #0,STPTBL+6             ;STEP 4
3428 037436  012737  040000  002310        MOV     #040000,CMPTBL+6        ;STEP 4 COMPARE
3429
3430 037444  012737  000022  002326        MOV     #18.,CMARLG             ;LENGTH OF COMM AREA FOR THIS TEST
3431 037452  004737  031426                JSR     PC,BAKPAT               ;FILL COMM AREA WITH ALL 1'S DATA
3432
```

```
3433 037456  004737  031330              JSR     PC,STEP1                ;GO DO IT
3434 037462  005737  002340              TST     STEPST                  ;IF STATUS OKAY
3435 037466  001412                      BEQ     5$                      ;  THEN CONTINUE TEST
3436
3437 037470                              ERRDF   19.,EMSG9,PRIINI        ;"SA CONTENTS IN ERROR"
     037470  104455                      TRAP    C$ERDF
     037472  000023                      .WORD   19
     037474  024271                      .WORD   EMSG9
     037476  027002                      .WORD   PRIINI
3438 037500                              CKLOOP                          ;LOOP ON ERROR?
     037500  104406                      TRAP    C$CLP1
3439 037502                              DODU    LOGUNT                  ;DROP UUT
     037502  013700  002332              MOV     LOGUNT,R0
     037506  104451                      TRAP    C$DODU
3440 037510                              ESCAPE  TST                     ;LEAVE TST
     037510  104410                      TRAP    C$ESCAPE
     037512  001166                      .WORD   L10024-.
3441
3442 037514  005237  002336      5$:     INC     INISTP                  ;ADJUST STEP COUNTER
3443 037520  062705  000002              ADD     #2,R5                   ;ADJUST TABLE INDEX
3444 037524  012737  000100  002346  6$: MOV     #100,OUTER              ;SET UP FOR DELAY ROUTINE
3445 037532  016537  002302  002334      MOV     CMPTBL(R5),SAEXP        ;SET UP FOR COMPARE
3446 037540  012737  037200  002344  7$: MOV     #16000.,INNER           ;SET UP INNER
3447 037546  017464  000002  000012      MOV     @TUSA(R4),TUSASV(R4)    ;GET SA CONTENTS
3448 037554  022705  000006              CMP     #6,R5                   ;ARE WE IN STEP 4?
3449 037560  001005                      BNE     8$                      ;BRANCH IF NOT
3450 037562  033764  002334  000012      BIT     SAEXP,TUSASV(R4)        ;JUST LOOK FOR STEP 4 BIT
3451 037570  001027                      BNE     10$                     ;IT'S SET SO LET'S GO
3452 037572  000404                      BR      9$                      ;STAY IN LOOP OTHERWISE
3453 037574  023764  002334  000012  8$: CMP     SAEXP,TUSASV(R4)        ;IF SA IS WHAT WE EXPECT
3454 037602  001422                      BEQ     10$                     ;  THEN MOVE ALONG
3455 037604  004737  031302      9$:     JSR     PC,PDELAY               ;  ELSE GIVE UUT SOME TIME
3456 037610  005737  002350              TST     TOUT                    ;IF NO TIMEOUT YET
3457 037614  001751                      BEQ     7$                      ;  THEN GO TAKE ANOTHER LOOK
3458
3459 037616  012737  025632  002330      MOV     #LESI,FRUIS             ;FAILING FRU
3460 037624                              ERRDF   20.,EMSG9,PRIINI        ;"SA CONTENTS IN ERROR"
     037624  104455                      TRAP    C$ERDF
     037626  000024                      .WORD   20
     037630  024271                      .WORD   EMSG9
     037632  027002                      .WORD   PRIINI
3461 037634                              CKLOOP
     037634  104406                      TRAP    C$CLP1
3462 037636                              DODU    LOGUNT
     037636  013700  002332              MOV     LOGUNT,R0
     037642  104451                      TRAP    C$DODU
3463 037644                              ESCAPE  TST
     037644  104410                      TRAP    C$ESCAPE
     037646  001032                      .WORD   L10024-.
3464
3465 037650  016574  002272  000002 10$: MOV     STPTBL(R5),@TUSA(R4)    ;WRITE NEXT STEP TO UUT
3466 037656  022705  000004              CMP     #4,R5                   ;IF STEP 3
3467 037662  001404                      BEQ     15$                     ;  THEN DO PURGE/POLL STUFF
3468 037664  022705  000006              CMP     #6,R5                   ;IF NOT IN STEP 4
3469 037670  001311                      BNE     5$                      ;  THEN GO BACK TO MAIN LOOP
3470 037672  000440                      BR      20$                     ;  ELSE GO CHECK RESULTS
3471
```

```
3472 037674                                15$:    DELAY   1                       ;GIVE PORT SOME TIME
     037674  012727  000001                        MOV     #1,(PC)+
     037700  000000                                .WORD   0
     037702  013727  002116                        MOV     L$DLY,(PC)+
     037706  000000                                .WORD   0
     037710  005367  177772                        DEC     -6(PC)
     037714  001375                                BNE     .-4
     037716  005367  177756                        DEC     -22(PC)
     037722  001367                                BNE     .-20
3473 037724  017464  000002  000012                MOV     @TUSA(R4),TUSASV(R4)    ;GET SA CONTENTS
3474 037732  001412                                BEQ     16$                     ;BRANCH IF OKAY
3475
3476 037734                                        ERRDF   21.,EMSG13,PRIINI       ;SA NOT 0 IN PURGE/POLL
     037734  104455                                TRAP    C$ERDF
     037736  000025                                .WORD   21
     037740  024454                                .WORD   EMSG13
     037742  027002                                .WORD   PRIINI
3477 037744                                        CKLOOP
     037744  104406                                TRAP    C$CLP1
3478 037746                                        DODU    LOGUNT
     037746  013700  002332                        MOV     LOGUNT,R0
     037752  104451                                TRAP    C$DODU
3479 037754                                        ESCAPE  TST
     037754  104410                                TRAP    C$ESCAPE
     037756  000722                                .WORD   L10024-.
3480
3481 037760  012774  000000  000002  16$:    MOV     #0,@TUSA(R4)            ;WRITE 0'S TO SA
3482 037766  005774  000000                        TST     @TUIP(R4)               ;AND READ IP
3483 037772  000650                                BR      5$                      ;GO WAIT FOR NEXT TRANSITION
3484
3485 037774  004737  031456            20$:    JSR     PC,CHKCOM               ;GO CHECK COMM AREA
3486 040000  032764  000001  000014              BIT     #DRPFLG,LUNFLG(R4)      ;HAS UUT BEEN DROPPED
3487 040006  001005                                BNE     T6EXT                   ;LEAVE NOW IF SO
3488 040010  005337  000000G                       DEC     ITRCNT                  ;IF NO MORE ITERATIONS LEFT
3489 040014  001402                                BEQ     T6EXT                   ;   THEN LEAVE TEST
3490 040016  000137  037326                        JMP     2$                      ;  ELSE DO IT AGAIN
3491
3492 040022                                T6EXT:  EXIT    TST
     040022  104432                                TRAP    C$EXIT
     040024  000654                                .WORD   L10024-.
3493 040026                                        ENDSUB
     040026                             L10025:
     040026  104403                                TRAP    C$ESUB
```

```
3496                                        .SBTTL  SUBTEST 2: EXTENDED ADDRESS TEST
3497
3498 040030                                        BGNSUB
     040030                                 T6.2:
     040030  104402                               TRAP    C$BSUB
3499
3500 040032  032764  000001  000014               BIT     #DRPFLG,LUNFLG(R4)      ;IF UUT NOT DROPPED
3501 040040  001407                               BEQ     1$                      ;  THEN DO TEST
3502 040042                                       EXIT    TST                     ;  ELSE GET OUT
     040042  104432                               TRAP    C$EXIT
     040044  000634                               .WORD   L10024-.
3503 040046  005737  002314                       TST     KTFLAG                  ;IF MEMORY MANAGEMENT AVAILABLE
3504 040052  001002                               BNE     1$                      ;  THEN DO TEST
3505 040054                                       EXIT    TST                     ;  ELSE GET OUT
     040054  104432                               TRAP    C$EXIT
     040056  000622                               .WORD   L10024-.
3506 040060  012737  025647  002330  1$:          MOV     #CTRL,FRUIS             ;FAILING FRU IN CASE OF ERROR
3507 040066  012737  000001  000000G              MOV     #1,ITRCNT               ;SET UP FOR ONE TEST ITERATION
3508 040074  022737  000001  002312               CMP     #1,PASCNT               ;IF FIRST PASS
3509 040102  001403                               BEQ     2$                      ;  THEN START TEST
3510 040104  012737  000012  000000G              MOV     #10.,ITRCNT             ;  ELSE DO 10 ITERATIONS
3511
3512 040112  004737  031616          2$:          JSR     PC,INTMMU               ;INITIALIZE MMU REGISTERS
3513 040116  012705  000000          3$:          MOV     #0,R5                   ;SET UP R5 AS INDEX TO STEP TABLES
3514 040122  012737  000001  002336               MOV     #1,INISTP               ;STEP 1 FOR ERROR PRINTOUT
3515 040130  016437  000004  002272               MOV     TUVEC(R4),STPTBL        ;PUT VECTOR IN STEP 1
3516 040136  006237  002272                       ASR     STPTBL                  ;DIVIDE BY TWO
3517 040142  006237  002272                       ASR     STPTBL                  ;DIVIDE BY FOUR
3518 040146  013737  002272  002306               MOV     STPTBL,CMPTBL+4         ;PUT VECTOR IN STEP 3 COMPARE
3519 040154  052737  111000  002272               BIS     #111000,STPTBL          ;REST OF STEP ONE
3520 040162  012737  005700  002302               MOV     #B.S1!B.QB!B.DI!B.OD!B.MP,CMPTBL
3521                                                                               ;STEP 1 COMPARE VALUE
3522 040170  012737  060050  002274               MOV     #COMMAR,STPTBL+2        ;STEP 2 - COMM AREA ADDRESS
3523 040176  042737  160000  002274               BIC     #BIT15!BIT14!BIT13,STPTBL+2
3524                                                                               ;CLEAR THE ACTIVE PAGE FIELD
3525 040204  012737  010222  002304               MOV     #010222,CMPTBL+2        ;STEP 2 COMPARE
3526 040212  013737  172346  002352               MOV     KPAR3,TEMP              ;GET RELOCATION VALUE
3527 040220  113737  002353  002276               MOVB    TEMP+1,STPTBL+4         ;JUST THE HGH BYTE
3528 040226  006237  002276                       ASR     STPTBL+4                ;MAKE IT THE EXTENDED
3529 040232  006237  002276                       ASR     STPTBL+4                ;  ADDRESS OF THE COMM AREA
3530 040236  052737  100000  002276               BIS     #B.PP,STPTBL+4          ;NOW SET PURGE/POLL BIT
3531 040244  112737  000040  002307               MOVB    #40,CMPTBL+5            ;REST OF STEP 3 COMPARE
3532 040252  012737  000000  002300               MOV     #0,STPTBL+6             ;STEP 4
3533 040260  012737  040000  002310               MOV     #040000,CMPTBL+6        ;STEP 4 COMPARE
3534
3535 040266  012737  000022  002326               MOV     #18.,CMARLG             ;LENGTH OF COMM AREA FOR THIS TEST
3536 040274  004737  031426                       JSR     PC,BAKPAT               ;FILL COMM AREA WITH ALL 1'S DATA
3537
3538 040300  004737  031330                       JSR     PC,STEP1                ;GO DO IT
3539 040304  005737  002340                       TST     STEPST                  ;IF STATUS OKAY
3540 040310  001412                               BEQ     5$                      ;  THEN CONTINUE TEST
3541
3542 040312                                       ERRDF   25.,EMSG9,PRIINI        ;"SA CONTENTS IN ERROR"
     040312  104455                               TRAP    C$ERDF
     040314  000031                               .WORD   25
     040316  024271                               .WORD   EMSG9
     040320  027002                               .WORD   PRIINI
```

```
3543 040322                                        CKLOOP                          ;LOOP ON ERROR?
     040322  104406                                TRAP    C$CLP1
3544 040324                                        DODU    LOGUNT                  ;DROP UUT
     040324  013700  002332                        MOV     LOGUNT,R0
     040330  104451                                TRAP    C$DODU
3545 040332                                        ESCAPE  TST                     ;LEAVE TST
     040332  104410                                TRAP    C$ESCAPE
     040334  000344                                .WORD   L10024-.
3546
3547 040336  005237  002336          5$:           INC     INISTP                  ;ADJUST STEP COUNTER
3548 040342  062705  000002                        ADD     #2,R5                   ;ADJUST TABLE INDEX
3549 040346  012737  000100  002346  6$:           MOV     #100,OUTER              ;SET UP FOR DELAY ROUTINE
3550 040354  016537  002302  002334                MOV     CMPTBL(R5),SAEXP        ;SET UP FOR COMPARE
3551 040362  012737  037200  002344  7$:           MOV     #16000.,INNER           ;SET UP INNER
3552 040370  017464  000002  000012                MOV     @TUSA(R4),TUSASV(R4)    ;GET SA CONTENTS
3553 040376  022705  000006                        CMP     #6,R5                   ;ARE WE IN STEP 4?
3554 040402  001005                                BNE     8$                      ;BRANCH IF NOT
3555 040404  033764  002334  000012                BIT     SAEXP,TUSASV(R4)        ;JUST LOOK FOR STEP 4 BIT
3556 040412  001024                                BNE     10$                     ;IT'S SET SO LET'S GO
3557 040414  000404                                BR      9$                      ;STAY IN LOOP OTHERWISE
3558 040416  023764  002334  000012  8$:           CMP     SAEXP,TUSASV(R4)        ;IF SA IS WHAT WE EXPECT
3559 040424  001417                                BEQ     10$                     ;  THEN MOVE ALONG
3560 040426  004737  031302          9$:           JSR     PC,PDELAY               ;  ELSE GIVE UUT SOME TIME
3561 040432  005737  002350                        TST     TOUT                    ;IF NO TIMEOUT YET
3562 040436  001751                                BEQ     7$                      ;  THEN GO TAKE ANOTHER LOOK
3563
3564 040440                                        ERRDF   26.,EMSG9,PRIINI        ;"SA CONTENTS IN ERROR"
     040440  104455                                TRAP    C$ERDF
     040442  000032                                .WORD   26
     040444  024271                                .WORD   EMSG9
     040446  027002                                .WORD   PRIINI
3565 040450                                        CKLOOP
     040450  104406                                TRAP    C$CLP1
3566 040452                                        DODU    LOGUNT
     040452  013700  002332                        MOV     LOGUNT,R0
     040456  104451                                TRAP    C$DODU
3567 040460                                        ESCAPE  TST
     040460  104410                                TRAP    C$ESCAPE
     040462  000216                                .WORD   L10024-.
3568
3569 040464  016574  002272  000002  10$:          MOV     STPTBL(R5),@TUSA(R4)    ;WRITE NEXT STEP TO UUT
3570 040472  022705  000004                        CMP     #4,R5                   ;IF STEP 3
3571 040476  001404                                BEQ     15$                     ;  THEN DO PURGE/POLL STUFF
3572 040500  022705  000006                        CMP     #6,R5                   ;IF NOT IN STEP 4
3573 040504  001314                                BNE     5$                      ;  THEN GO BACK TO MAIN LOOP
3574 040506  000440                                BR      20$                     ;  ELSE GO CHECK RESULTS
3575
3576 040510                          15$:          DELAY   1                       ;GIVE PORT SOME TIME
     040510  012727  000001                        MOV     #1,(PC)+
     040514  000000                                .WORD   0
     040516  013727  002116                        MOV     L$DLY,(PC)+
     040522  000000                                .WORD   0
     040524  005367  177772                        DEC     -6(PC)
     040530  001375                                BNE     .-4
     040532  005367  177756                        DEC     -22(PC)
     040536  001367                                BNE     .-20
3577 040540  017464  000002  000012                MOV     @TUSA(R4),TUSASV(R4)    ;GET SA CONTENTS
```

```
                                                BEQ     16$                     ;BRANCH IF OKAY
                                                ERRDF   27.,EMSG13,PRIINI       ;SA NOT 0 IN PURGE/POLL
                                                TRAP    C$ERDF
                                                .WORD   27
                                                .WORD   EMSG13
                                                .WORD   PRIINI
                                                CKLOOP
                                                TRAP    C$CLP1
                                                DODU    LOGUNT
                                                MOV     LOGUNT,R0
                                                TRAP    C$DODU
                                                ESCAPE  TST
                                                TRAP    C$ESCAPE
                                                .WORD   L10024-.

                                        16$:    MOV     #0,@TUSA(R4)            ;WRITE 0'S TO SA
                                                TST     @TUIP(R4)               ;AND READ IP
                                                BR      5$                      ;GO WAIT FOR NEXT TRANSITION

                                        20$:    JSR     PC,CHKCOM               ;GO CHECK COMM AREA
                                                BIT     #DRPFLG,LUNFLG(R4)      ;HAS UUT BEEN DROPPED
                                                BNE     ST6EXT                  ;LEAVE NOW IF SO

                                                ADD     #2000,KPAR3             ;POINT TO NEXT 32KWORDS
                                                BCS     25$                     ;DON'T ALLOW OVERFLOW IF 4 MBYTES
                                                CMP     L$HIME,KPAR3            ;IF THERE'S NO MORE MEMORY AVAILABLE
                                                BLO     25$                     ;  THEN CHECK FOR MORE ITERATIONS
                                                JMP     3$                      ;  ELSE DO IT AGAIN

                                        25$:    CLR     MMUSR0                  ;SHUT DOWN MEMORY MANAGEMENT
                                                DEC     ITRCNT                  ;IF NO MORE ITERATIONS LEFT
                                                BEQ     ST6EXT                  ;  THEN LEAVE TEST
                                                JMP     2$                      ;  ELSE DO IT AGAIN

                                        ST6EXT: CLR     MMUSR0                  ;MAKE SURE IT'S OFF
                                                EXIT    TST
                                                TRAP    C$EXIT
                                                .WORD   L10024-.

                                                ENDSUB
                                        L10026:
                                                TRAP    C$ESUB

                                                ENDTST
                                        L10024:
                                                TRAP    C$ETST
```

```
                                .SBTTL TEST 7: SMALL RING TEST

                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;
                                ;TEST 7 - SMALL RING TEST
                                ;       THIS TEST IS SIMILAR TO TEST 6, HOWEVER, RING DEPTH
                                ;       USED IN THIS TEST IS THE MINIMUM.
                                ;
                                ;;*****************************************************************
                                ;;*****************************************************************

                                        BGNTST
040702                          T7::
040702

040702  032764  000001  000014          BIT     #DRPFLG,LUNFLG(R4)      ;IF UUT NOT DROPPED
040710  001402                          BEQ     1$                      ;  THEN DO TEST
040712                                  EXIT    TST                     ;  ELSE GET OUT
040712  104432                          TRAP    C$EXIT
040714  000526                          .WORD   L10027-.
040716  012737  025647  002330  1$:     MOV     #CTRL,FRUIS             ;FAILING FRU IN CASE OF ERROR
040724  012737  000001  000000G         MOV     #1,ITRCNT               ;SET UP FOR ONE TEST ITERATION
040732  022737  000001  002312          CMP     #1,PASCNT               ;IF FIRST PASS
040740  001403                          BEQ     2$                      ;  THEN START TEST
040742  012737  000012  000000G         MOV     #10.,ITRCNT             ;  ELSE DO 10 ITERATIONS

040750  012705  000000          2$:     MOV     #0,R5                   ;SET UP R5 AS INDEX TO STEP TABLES
040754  012737  000001  002336          MOV     #1,INISTP               ;STEP 1 FOR ERROR PRINTOUT
040762  016437  000004  002272          MOV     TUVEC(R4),STPTBL        ;PUT VECTOR IN STEP 1
040770  006237  002272                  ASR     STPTBL                  ;DIVIDE BY TWO
040774  006237  002272                  ASR     STPTBL                  ;DIVIDE BY FOUR
041000  013737  002272  002306          MOV     STPTBL,CMPTBL+4         ;PUT VECTOR IN STEP 3 COMPARE
041006  052737  104400  002272          BIS     #104400,STPTBL          ;REST OF STEP ONE
041014  012737  005700  002302          MOV     #B.S1!B.QB!B.DI!B.OD!B.MP,CMPTBL
                                                                        ;STEP 1 COMPARE VALUE
041022  012737  060050  002274          MOV     #COMMAR,STPTBL+2        ;STEP 2 - COMM AREA ADDRESS
041030  012737  010211  002304          MOV     #010211,CMPTBL+2        ;STEP 2 COMPARE
041036  012737  100000  002276          MOV     #B.PP,STPTBL+4          ;STEP 3 - HIGH ADDRESS AND PRGE/POLL
041044  112737  000040  002307          MOVB    #40,CMPTBL+5            ;REST OF STEP 3 COMPARE
041052  012737  000000  002300          MOV     #0,STPTBL+6             ;STEP 4
041060  012737  040000  002310          MOV     #040000,CMPTBL+6        ;STEP 4 COMPARE

041066  012737  000012  002326          MOV     #10.,CMARLG             ;LENGTH OF COMM AREA FOR THIS TEST
041074  004737  031426                  JSR     PC,BAKPAT               ;FILL COMM AREA WITH ALL 1'S DATA

041100  004737  031330                  JSR     PC,STEP1                ;GO DO IT
041104  005737  002340                  TST     STEPST                  ;IF STATUS OKAY
041110  001412                          BEQ     5$                      ;  THEN CONTINUE TEST

041112                                  ERRDF   19.,EMSG9,PRIINI        ;"SA CONTENTS IN ERROR"
041112  104455                          TRAP    C$ERDF
041114  000023                          .WORD   19
041116  024271                          .WORD   EMSG9
041120  027002                          .WORD   PRIINI
041122                                  CKLOOP                          ;LOOP ON ERROR?
041122  104406                          TRAP    C$CLP1
041124                                  DODU    LOGUNT                  ;DROP UUT
```

```
                041124  013700  002332                  MOV     LOGUNT,R0
                041130  104451                          TRAP    C$DODU
          3667  041132                                  ESCAPE  TST                     ;LEAVE TST
                041132  104410                          TRAP    C$ESCAPE
                041134  000306                          .WORD   L10027-.
          3668
          3669  041136  005237  002336          5$:     INC     INISTP                  ;ADJUST STEP COUNTER
          3670  041142  062705  000002                  ADD     #2,R5                   ;ADJUST TABLE INDEX
          3671  041146  012737  000100  002346  6$:     MOV     #100.,OUTER             ;SET UP FOR DELAY ROUTINE
          3672  041154  016537  002302  002334          MOV     CMPTBL(R5),SAEXP        ;SET UP FOR COMPARE
          3673  041162  012737  037200  002344  7$:     MOV     #16000.,INNER           ;SET UP INNER
          3674  041170  017464  000002  000012          MOV     @TUSA(R4),TUSASV(R4)    ;GET SA CONTENTS
          3675  041176  022705  000006                  CMP     #6,R5                   ;ARE WE IN STEP 4?
          3676  041202  001005                          BNE     8$                      ;BRANCH IF NOT
          3677  041204  033764  002334  000012          BIT     SAEXP,TUSASV(R4)        ;JUST LOOK FOR STEP 4 BIT
          3678  041212  001024                          BNE     10$                     ;IT'S SET SO LET'S GO
          3679  041214  000404                          BR      9$                      ;STAY IN LOOP OTHERWISE
          3680  041216  023764  002334  000012  8$:     CMP     SAEXP,TUSASV(R4)        ;IF SA IS WHAT WE EXPECT
          3681  041224  001417                          BEQ     10$                     ;  THEN MOVE ALONG
          3682  041226  004737  031302          9$:     JSR     PC,PDELAY               ;  ELSE GIVE UUT SOME TIME
          3683  041232  005737  002350                  TST     TOUT                    ;IF NO TIMEOUT YET
          3684  041236  001751                          BEQ     7$                      ;  THEN GO TAKE ANOTHER LOOK
          3685
          3686  041240                                  ERRDF   20.,EMSG9,PRIINI        ;"SA CONTENTS IN ERROR"
                041240  104455                          TRAP    C$ERDF
                041242  000024                          .WORD   20
                041244  024271                          .WORD   EMSG9
                041246  027002                          .WORD   PRIINI
          3687  041250                                  CKLOOP
                041250  104406                          TRAP    C$CLP1
          3688  041252                                  DODU    LOGUNT
                041252  013700  002332                  MOV     LOGUNT,R0
                041256  104451                          TRAP    C$DODU
          3689  041260                                  ESCAPE  TST
                041260  104410                          TRAP    C$ESCAPE
                041262  000160                          .WORD   L10027-.
          3690
          3691  041264  016574  002272  000002  10$:    MOV     STPTBL(R5),@TUSA(R4)    ;WRITE NEXT STEP TO UUT
          3692  041272  022705  000004                  CMP     #4,R5                   ;IF STEP 3
          3693  041276  001404                          BEQ     15$                     ;  THEN DO PURGE/POLL STUFF
          3694  041300  022705  000006                  CMP     #6,R5                   ;IF NOT IN STEP 4
          3695  041304  001314                          BNE     5$                      ;  THEN GO BACK TO MAIN LOOP
          3696  041306  000440                          BR      20$                     ;  ELSE GO CHECK RESULTS
          3697
          3698  041310                          15$:    DELAY   1                       ;GIVE PORT SOME TIME
                041310  012727  000001                  MOV     #1,(PC)+
                041314  000000                          .WORD   0
                041316  013727  002116                  MOV     L$DLY,(PC)+
                041322  000000                          .WORD   0
                041324  005367  177772                  DEC     -6(PC)
                041330  001375                          BNE     .-4
                041332  005367  177756                  DEC     -22(PC)
                041336  001367                          BNE     .-20
          3699  041340  017464  000002  000012          MOV     @TUSA(R4),TUSASV(R4)    ;GET SA CONTENTS
          3700  041346  001412                          BEQ     16$                     ;BRANCH IF OKAY
          3701
          3702  041350                                  ERRDF   21.,EMSG13,PRIINI       ;SA NOT 0 IN PURGE/POLL
```

TEST 7: SMALL RING TEST

```
                041350  104455                          TRAP    C$ERDF
                041352  000025                          .WORD   21
                041354  024454                          .WORD   EMSG13
                041356  027002                          .WORD   PRIINI
          3703  041360                                  CKLOOP
                041360  104406                          TRAP    C$CLP1
          3704  041362                                  DODU    LOGUNT
                041362  013700  002332                  MOV     LOGUNT,R0
                041366  104451                          TRAP    C$DODU
          3705  041370                                  ESCAPE  TST
                041370  104410                          TRAP    C$ESCAPE
                041372  000050                          .WORD   L10027-.
          3706
          3707  041374  012774  000000  000002  16$:    MOV     #0,@TUSA(R4)            ;WRITE 0'S TO SA
          3708  041402  005774  000000                  TST     @TUIP(R4)               ;AND READ IP
          3709  041406  000653                          BR      5$                      ;GO WAIT FOR NEXT TRANSITION
          3710
          3711  041410  004737  031456          20$.    JSR     PC,CHKCOM               ;GO CHECK COMM AREA
          3712  041414  032764  000001  000014          BIT     #DRPFLG,LUNFLG(R4)      ;HAS UUT BEEN DROPPED
          3713  041422  C01005                          BNE     T7EXT                   ;LEAVE NOW IF SO
          3714  041424  005337  000000G                 DEC     ITRCNT                  ;IF NO MORE ITERATIONS LEFT
          3715  041430  001402                          BEQ     T7EXT                   ;  THEN LEAVE TEST
          3716  041432  000137  040750                  JMP     2$                      ;  ELSE DO IT AGAIN
          3717
          3718  041436                          T7EXT:  EXIT    TST
                041436  104432                          TRAP    C$EXIT
                041440  000002                          .WORD   L10027-.
          3719
          3720  041442                                  ENDTST
                041442                          L10027:
                041442  104401                          TRAP    C$ETST
```

```
3723                                    .SBTTL TEST 8: MAXIMUM RING BUFFER TEST
3724
3725 041444                                    BGNTST
     041444                             T8::
3726
3727 041444  032764  000001  000014            BIT     #DRPFLG,LUNFLG(R4)      ;IF UUT NOT DROPPED
3728 041452  001402                            BEQ     1$                      ;  THEN DO TEST
3729 041454                                    EXIT    TST                     ;  ELSE GET OUT
     041454  104432                            TRAP    C$EXIT
     041456  000526                            .WORD   L10030-.
3730 041460  012737  025647  002330  1$:       MOV     #CTRL,FRUIS             ;FAILING FRU IN CASE OF ERROR
3731 041466  012737  000001  000000G           MOV     #1,ITRCNT               ;SET UP FOR ONE TEST ITERATION
3732 041474  022737  000001  002312            CMP     #1,PASCNT               ;IF FIRST PASS
3733 041502  001403                            BEQ     2$                      ;  THEN START TEST
3734 041504  012737  000012  000000G           MOV     #10.,ITRCNT             ;  ELSE DO 10 ITERATIONS
3735
3736 041512  012705  000000            2$:     MOV     #0,R5                   ;SET UP R5 AS INDEX TO STEP TABLES
3737 041516  012737  000001  002336            MOV     #1,INISTP               ;STEP 1 FOR ERROR PRINTOUT
3738 041524  016437  000004  002272            MOV     TUVEC(R4),STPTBL        ;PUT VECTOR IN STEP 1
3739 041532  006237  002272                    ASR     STPTBL                  ;DIVIDE BY TWO
3740 041536  006237  002272                    ASR     STPTBL                  ;DIVIDE BY FOUR
3741 041542  013737  002272  002306            MOV     STPTBL,CMPTBL+4         ;PUT VECTOR IN STEP 3 COMPARE
3742 041550  052737  137400  002272            BIS     #137400,STPTBL          ;REST OF STEP ONE
3743 041556  012737  005700  002302            MOV     #8.S1!8.QB!8.DI!8.OD!8.MP,CMPTBL
3744                                                                            ;STEP 1 COMPARE VALUE
3745 041564  012737  060050  002274            MOV     #COMMAR,STPTBL+2        ;STEP 2 - COMM AREA ADDRESS
3746 041572  012737  010277  002304            MOV     #010277,CMPTBL+2        ;STEP 2 COMPARE
3747 041600  012737  100000  002276            MOV     #8.PP,STPTBL+4          ;STEP 3 - HIGH ADDRESS AND PRGE/POLL
3748 041606  112737  000040  002307            MOVB    #40,CMPTBL+5            ;REST OF STEP 3 COMPARE
3749 041614  012737  000000  002300            MOV     #0,STPTBL+6             ;STEP 4
3750 041622  012737  040000  002310            MOV     #040000,CMPTBL+6        ;STEP 4 COMPARE
3751
3752 041630  012737  001002  002326            MOV     #514.,CMARLG            ;LENGTH OF COMM AREA FOR THIS TEST
3753 041636  004737  031426                    JSR     PC,BAKPAT               ;FILL COMM AREA WITH ALL 1'S DATA
3754
3755 041642  004737  031330                    JSR     PC,STEP1                ;GO DO IT
3756 041646  005737  002340                    TST     STEPST                  ;IF STATUS OKAY
3757 041652  001412                            BEQ     5$                      ;  THEN CONTINUE TEST
3758
3759 041654                                    ERRDF   22.,EMSG9,PRIINI        ;"SA CONTENTS IN ERROR"
     041654  104455                            TRAP    C$ERDF
     041656  000026                            .WORD   22
     041660  024271                            .WORD   EMSG9
     041662  027002                            .WORD   PRIINI
3760 041664                                    CKLOOP                          ;LOOP ON ERROR?
     041664  104406                            TRAP    C$CLP1
3761 041666                                    DODU    LOGUNT                  ;DROP UUT
     041666  013700  002332                    MOV     LOGUNT,R0
     041672  104451                            TRAP    C$DODU
3762 041674                                    ESCAPE  TST                     ;LEAVE TST
     041674  104410                            TRAP    C$ESCAPE
     041676  000306                            .WORD   L10030-.
3763
3764 041700  005237  002336            5$:     INC     INISTP                  ;ADJUST STEP COUNTER
3765 041704  062705  000002                    ADD     #2,R5                   ;ADJUST TABLE INDEX
3766 041710  012737  000100  002346  6$:       MOV     #100,OUTER              ;SET UP FOR DELAY ROUTINE
3767 041716  016537  002302  002334            MOV     CMPTBL(R5),SAEXP        ;SET UP FOR COMPARE
```

TEST 8: MAXIMUM RING BUFFER TEST

```
 3768 041724 012737 037200 002344 7$:    MOV     #16000.,INNER           ;SET UP INNER
 3769 041732 017464 000002 000012        MOV     @TUSA(R4),TUSASV(R4)    ;GET SA CONTENTS
 3770 041740 022705 000006               CMP     #6,R5                   ;ARE WE IN STEP 4?
 3771 041744 001005                      BNE     8$                      ;BRANCH IF NOT
 3772 041746 033764 002334 000012        BIT     SAEXP,TUSASV(R4)        ;JUST LOOK FOR STEP 4 BIT
 3773 041754 001024                      BNE     10$                     ;IT'S SET SO LET'S GO
 3774 041756 000404                      BR      9$                      ;STAY IN LOOP OTHERWISE
 3775 041760 023764 002334 000012 8$:    CMP     SAEXP,TUSASV(R4)        ;IF SA IS WHAT WE EXPECT
 3776 041766 001417                      BEQ     10$                     ;  THEN MOVE ALONG
 3777 041770 004737 031302        9$:    JSR     PC,PDELAY               ;  ELSE GIVE UUT SOME TIME
 3778 041774 005737 002350               TST     TOUT                    ;IF NO TIMEOUT YET
 3779 042000 001751                      BEQ     7$                      ;  THEN GO TAKE ANOTHER LOOK
 3780
 3781 042002                             ERRDF   23.,EMSG9,PRIINI        ;"SA CONTENTS IN ERROR"
      042002 104455                      TRAP    C$ERDF
      042004 000027                      .WORD   23
      042006 024271                      .WORD   EMSG9
      042010 027002                      .WORD   PRIINI
 3782 042012                             CKLOOP
      042012 104406                      TRAP    C$CLP1
 3783 042014                             DODU    LOGUNT
      042014 013700 002332               MOV     LOGUNT,R0
      042020 104451                      TRAP    C$DODU
 3784 042022                             ESCAPE  TST
      042022 104410                      TRAP    C$ESCAPE
      042024 000160                      .WORD   L10030-.
 3785
 3786 042026 016574 002272 000002 10$:   MOV     STPTBL(R5),@TUSA(R4)    ;WRITE NEXT STEP TO UUT
 3787 042034 022705 000004               CMP     #4,R5                   ;IF STEP 3
 3788 042040 001404                      BEQ     15$                     ;  THEN DO PURGE/POLL STUFF
 3789 042042 022705 000006               CMP     #6,R5                   ;IF NOT IN STEP 4
 3790 042046 001314                      BNE     5$                      ;  THEN GO BACK TO MAIN LOOP
 3791 042050 000440                      BR      20$                     ;  ELSE GO CHECK RESULTS
 3792
 3793 042052                     15$:    DELAY   1                       ;GIVE PORT SOME TIME
      042052 012727 000001               MOV     #1,(PC)+
      042056 000000                      .WORD   0
      042060 013727 002116               MOV     L$DLY,(PC)+
      042064 000000                      .WORD   0
      042066 005367 177772               DEC     -6(PC)
      042072 001375                      BNE     .-4
      042074 005367 177756               DEC     -22(PC)
      042100 001367                      BNE     .-20
 3794 042102 017464 000002 000012        MOV     @TUSA(R4),TUSASV(R4)    ;GET SA CONTENTS
 3795 042110 001412                      BEQ     16$                     ;BRANCH IF OKAY
 3796
 3797 042112                             ERRDF   24.,EMSG13,PRIINI       ;SA NOT 0 IN PURGE/POLL
      042112 104455                      TRAP    C$ERDF
      042114 000030                      .WORD   24
      042116 024454                      .WORD   EMSG13
      042120 027002                      .WORD   PRIINI
 3798 042122                             CKLOOP
      042122 104406                      TRAP    C$CLP1
 3799 042124                             DODU    LOGUNT
      042124 013700 002332               MOV     LOGUNT,R0
      042130 104451                      TRAP    C$DODU
 3800 042132                             ESCAPE  TST
```

TEST 8: MAXIMUM RING BUFFER TEST

```
        042132  104410                          TRAP    C$ESCAPE
        042134  000050                          .WORD   L10030-.
   3801
   3802 042136  012774  000000  000002  16$:    MOV     #0,@TUSA(R4)            ;WRITE 0'S TO SA
   3803 042144  005774  000000                  TST     @TUIP(R4)               ;AND READ IP
   3804 042150  000653                          BR      5$                      ;GO WAIT FOR NEXT TRANSITION
   3805
   3806 042152  004737  031456          20$:    JSR     PC,CHKCOM               ;GO CHECK COMM AREA
   3807 042156  032764  000001  000014          BIT     #DRPFLG,LUNFLG(R4)      ;HAS UUT BEEN DROPPED
   3808 042164  001005                          BNE     T8EXT                   ;LEAVE NOW IF SO
   3809 042166  005337  000000G                 DEC     ITRCNT                  ;IF NO MORE ITERATIONS LEFT
   3810 042172  001402                          BEQ     T8EXT                   ;   THEN LEAVE TEST
   3811 042174  000137  041512                  JMP     2$                      ;  ELSE DO IT AGAIN
   3812
   3813 042200                          T8EXT:  EXIT    TST
        042200  104432                          TRAP    C$EXIT
        042202  000002                          .WORD   L10030-.
   3814
   3815 042204                                  ENDTST
        042204                          L10030:
        042204  104401                          TRAP    C$ETST
```

```
3819                                    .SBTTL  TEST 9:GET DUST STATUS
3820
3821 042206                                     BGNTST
     042206                             T9::
3822 042206  032764  000001  000014             BIT     #DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
3823 042214  001022                             BNE     T9EXT                   ;GET OUT IF NOT AVAILABLE
3824 042216  012737  025647  002330             MOV     #CTRL,FRUIS             ;DEFAULT FRU IS CONTROLLER
3825 042224  005064  000014                     CLR     LUNFLG(R4)              ;CLEAR ALL FLAGS
3826 042230  004737  031716                     JSR     PC,PRTINT               ;GO DO A PORT INITIALIZE
3827 042234  032764  000001  000014             BIT     #DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
3828 042242  001007                             BNE     T9EXT                   ;NO, BRANCH TO EXIT
3829 042244  052764  000010  000014             BIS     #TEST.9,LUNFLG(R4)      ;SET TEST 9 FLAG
3830 042252  012705  002370                     MOV     #GDUST,R5               ;SET UP TO DO GET DUST STATUS COMMAND
3831 042256  004737  032350                     JSR     PC,CLSDRV               ;GO ISSUE THE COMMAND
3832 042262                             T9EXT:  EXIT    TST
     042262  104432                             TRAP    C$EXIT
     042264  000002                             .WORD   L10031-.
3833 042266                                     ENDTST
     042266                             L10031:
     042266  104401                             TRAP    C$ETST
```

```
3835                                        .SBTTL  TEST 10: FUNCTIONAL FAULT DETECTION TEST (Internal Drive Test 1)
3836
3837 042270                                        BGNTST
     042270                                 T10::
3838 042270  032764  000001  000014                BIT     #DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
3839 042276  001062                                BNE     T10EXT                  ;NO, BRANCH TO EXIT
3840 042300                                        MANUAL                          ;MANUAL INTERVENTION ALLOWED ?
     042300  104450                                TRAP    C$MANI
3841 042302                                        BNCOMPLETE      T10EXT          ;NO, BRANCH TO EXIT
     042302  103060                                BCC     T10EXT
3842 042304                                 1$:     PRINTF  #T10MS1                 ;PRINT TEST 10 MESSAGE
     042304  012746  025724                        MOV     #T10MS1, (SP)
     042310  012746  000001                        MOV     #1,-(SP)
     042314  010600                                MOV     SP,R0
     042316  104417                                TRAP    C$PNTF
     042320  062706  000004                        ADD     #4,SP
3843 042324                                        PRINTF  #T10MS2                 ;PRINT TEST 10 MESSAGE
     042324  012746  026032                        MOV     #T10MS2, (SP)
     042330  012746  000001                        MOV     #1,-(SP)
     042334  010600                                MOV     SP,R0
     042336  104417                                TRAP    C$PNTF
     042340  062706  000004                        ADD     #4,SP
3844 042344                                        PRINTF  #T10MS3                 ;PRINT TEST 10 MESSAGE
     042344  012746  026057                        MOV     #T10MS3, (SP)
     042350  012746  000001                        MOV     #1,-(SP)
     042354  010600                                MOV     SP,R0
     042356  104417                                TRAP    C$PNTF
     042360  062706  000004                        ADD     #4,SP
3845 042364                                        PRINTF  #T10MS4                 ;PRINT TEST 10 MESSAGE
     042364  012746  026134                        MOV     #T10MS4,-(SP)
     042370  012746  000001                        MOV     #1,-(SP)
     042374  010600                                MOV     SP,R0
     042376  104417                                TRAP    C$PNTF
     042400  062706  000004                        ADD     #4,SP
3846 042404                                        GMANIL  QUESTN,ANSWER,1,YES     ;GET OPERATOR INPUT
     042404  104443                                TRAP    C$GMAN
     042406  000404                                BR      10000$
     042410  002354                                .WORD   ANSWER
     042412  000130                                .WORD   T$CODE
     042414  026716                                .WORD   QUESTN
     042416  000001                                .WORD   1
     042420                                 10000$:
3847 042420  005737  002354                        TST     ANSWER                  ;DID OPERATOR ANSWER YES ?
3848 042424  001407                                BEQ     T10EXT                  ;NO, BRANCH TO EXIT
3849 042426  005037  002354                        CLR     ANSWER                  ;CLEAR OPERATOR ANSWER
3850 042432  112737  000061  002424                MOVB    #61,TSTNAM              ;LOAD DRIVE TEST NAME (ASCII 1)
3851 042440  004737  032150                        JSR     PC,DRVTST               ;GO RUN THE INTERNAL DRIVE TEST
3852 042444                                 T10EXT: EXIT    TST
     042444  104432                                TRAP    C$EXIT
     042446  000002                                .WORD   L10032-.
3853 042450                                        ENDTST
     042450                                 L10032:
     042450  104401                                TRAP    C$ETST
```

```
3855                                    .SBTTL  TEST 11: TENSION FAULT ISOLATION TEST (Internal Drive Test 2)
3856
3857 042452                                    BGNTST
     042452                             T11::
3858 042452  032764  000001  000014            BIT     #DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
3859 042460  001042                            BNE     T11EXT                  ;NO, BRANCH TO EXIT
3860 042462                                    MANUAL                          ;MANUAL INTERVENTION ALLOWED ?
     042462  104450                            TRAP    C$MANI
3861 042464                                    BNCOMPLETE      T11EXT          ;NO, BRANCH TO EXIT
     042464  103040                            BCC     T11EXT
3862 042466                             1$:    PRINTF  #T11MS1                 ;PRINT TEST 11 MESSAGE
     042466  012746  026221                    MOV     #T11MS1,-(SP)
     042472  012746  000001                    MOV     #1,-(SP)
     042476  010600                            MOV     SP,R0
     042500  104417                            TRAP    C$PNTF
     042502  062706  000004                    ADD     #4,SP
3863 042506                                    PRINTF  #MMSG                   ;PRINT REQUIREMENT MESSAGE
     042506  012746  026536                    MOV     #MMSG,-(SP)
     042512  012746  000001                    MOV     #1,-(SP)
     042516  010600                            MOV     SP,R0
     042520  104417                            TRAP    C$PNTF
     042522  062706  000004                    ADD     #4,SP
3864 042526                                    GMANIL  QUESTN,ANSWER,1,YES     ;GET OPERATOR INPUT
     042526  104443                            TRAP    C$GMAN
     042530  000404                            BR      10000$
     042532  002354                            .WORD   ANSWER
     042534  000130                            .WORD   T$CODE
     042536  026716                            .WORD   QUESTN
     042540  000001                            .WORD   1
     042542                             10000$:
3865 042542  005737  002354                    TST     ANSWER                  ;DID OPERATOR ANSWER YES ?
3866 042546  001407                            BEQ     T11EXT                  ;NO, BRANCH TO EXIT
3867 042550  005037  002354                    CLR     ANSWER                  ;CLEAR OPERATOR ANSWER
3868 042554  112737  000062  002424            MOVB    #62,TSTNAM              ;LOAD PROGRAM NAME (ASCII 2)
3869 042562  004737  032150                    JSR     PC,DRVTST               ;GO RUN THE INTERNAL DRIVE TEST
3870 042566                             T11EXT: EXIT    TST
     042566  104432                            TRAP    C$EXIT
     042570  000002                            .WORD   L10033-.
3871 042572                                    ENDTST
     042572                             L10033:
     042572  104401                            TRAP    C$ETST
```

```
3873                                        .SBTTL  TEST 12: VELOCITY FAULT ISOLATION TEST (Internal Drive Test 3)
3874
3875 042574                                         BGNTST
     042574                                 T12::
3876 042574  032764  000001  000014                BIT     @DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
3877 042602  001042                                BNE     T12EXT                  ;NO, BRANCH TO EXIT
3878 042604                                         MANUAL                          ;MANUAL INTERVENTION ALLOWED ?
     042604  104450                                TRAP    C$MANI
3879 042606                                         BNCOMPLETE      T12EXT          ;NO, BRANCH TO EXIT
     042606  103040                                BCC     T12EXT
3880 042610                                 1$:     PRINTF  #T12MS1                 ;PRINT TEST 12 MESSAGE
     042610  012746  026324                        MOV     #T12MS1,-(SP)
     042614  012746  000001                        MOV     #1,-(SP)
     042620  010600                                MOV     SP,R0
     042622  104417                                TRAP    C$PNTF
     042624  062706  000004                        ADD     #4,SP
3881 042630                                         PRINTF  #MMSG                   ;PRINT TEST REQUIREMENT MESSAGE
     042630  012746  026536                        MOV     #MMSG,-(SP)
     042634  012746  000001                        MOV     #1,-(SP)
     042640  010600                                MOV     SP,R0
     042642  104417                                TRAP    C$PNTF
     042644  062706  000004                        ADD     #4,SP
3882 042650                                         GMANIL  QUESTN,ANSWER,1,YES     ;GET OPERATOR INPUT
     042650  104443                                TRAP    C$GMAN
     042652  000404                                BR      10000$
     042654  002354                                .WORD   ANSWER
     042656  000130                                .WORD   T$CODE
     042660  026716                                .WORD   QUESTN
     042662  000001                                .WORD   1
     042664                                 10000$:
3883 042664  005737  002354                        TST     ANSWER                  ;DID OPERATOR ANSWER YES ?
3884 042670  001407                                BEQ     T12EXT                  ;NO, BRANCH TO EXIT
3885 042672  005037  002354                        CLR     ANSWER                  ;CLEAR OPERATOR ANSWER
3886 042676  112737  000063  002424                MOVB    #63,TSTNAM              ;LOAD PROGRAM NAME (ASCII 3)
3887 042704  004737  032150                        JSR     PC,DRVTST               ;GO RUN THE INTERNAL DRIVE TEST
3888 042710                                 T12EXT: EXIT    TST
     042710  104432                                TRAP    C$EXIT
     042712  000002                                .WORD   L10034-.
3889 042714                                         ENDTST
     042714                                 L10034:
     042714  104401                                TRAP    C$ETST
```

```
3891                                            .SBTTL  TEST 13: SELECT A DRIVE RESIDENT TEST (Internal Drive Tests 1 99)
3892
3893 042716                                             BGNTST
     042716                                     T13::
3894 042716  032764  000001  000014                     BIT     @DRPFLG,LUNFLG(R4)      ;IS THE DRIVE AVAILABLE
3895 042724  001065                                     BNE     T13EXT                  ;NO, BRANCH TO EXIT
3896 042726                                             MANUAL                          ;MANUAL INTERVENTION ALLOWED ?
     042726  104450                                     TRAP    C$MANI
3897 042730                                             BNCOMPLETE      T13EXT          ;NO, BRANCH TO EXIT
     042730  103063                                     BCC     T13EXT
3898 042732                                     1$:     PRINTF  @T13MS1                 ;PRINT TEST 13 MESSAGE
     042732  012746  026430                             MOV     @T13MS1,-(SP)
     042736  012746  000001                             MOV     #1,-(SP)
     042742  010600                                     MOV     SP,R0
     042744  104417                                     TRAP    C$PNTF
     042746  062706  000004                             ADD     #4,SP
3899 042752                                             PRINTF  @MMSG                   ;PRINT TEST REQUIREMENT MESSAGE
     042752  012746  026536                             MOV     @MMSG,-(SP)
     042756  012746  000001                             MOV     #1,-(SP)
     042762  010600                                     MOV     SP,R0
     042764  104417                                     TRAP    C$PNTF
     042766  062706  000004                             ADD     #4,SP
3900 042772                                             GMANID  SELTST,MANTBL,A,,1,2,NO ;ASK OPERATOR FOR TEST NUMBER
     042772  104443                                     TRAP    C$GMAN
     042774  000406                                     BR      10000$
     042776  022754                                     .WORD   MANTBL
     043000  000142                                     .WORD   T$CODE
     043002  026646                                     .WORD   SELTST
     043004  000000                                     .WORD
     043006  000001                                     .WORD   T$LOLIM
     043010  000002                                     .WORD   T$HILIM
     043012                                     10000$:
3901 043012  012702  002424                             MOV     @TSTNAM,R2              ;GET ADDRESS OF DRIVE TEST NAME
3902 043016  012703  022754                             MOV     @MANTBL,R3              ;GET ADDRESS OF OPERATOR INPUT DATA
3903 043022  112322                                     MOVB    (R3)+,(R2)+             ;LOAD 1ST DIGIT OF TEST NAME
3904 043024  105713                                     TSTB    (R3)                    ;CHECK FOR A 2ND DIGIT
3905 043026  001401                                     BEQ     10$                     ;BRANCH IF NONE
3906 043030  111312                                     MOVB    (R3),(R2)               ;LOAD 2ND DIGIT OF TEST NAME
3907 043032                                     10$:    GMANIL  QUESTN,ANSWER,1,YES     ;ASK OPERATOR IF READY
     043032  104443                                     TRAP    C$GMAN
     043034  000404                                     BR      10001$
     043036  002354                                     .WORD   ANSWER
     043040  000130                                     .WORD   T$CODE
     043042  026716                                     .WORD   QUESTN
     043044  000001                                     .WORD   1
     043046                                     10001$:
3908 043046  005737  002354                             TST     ANSWER                  ;DID OPERATOR ANSWER YES ?
3909 043052  001412                                     BEQ     T13EXT                  ;NO, BRANCH TO EXIT
3910 043054  005037  002354                             CLR     ANSWER                  ;CLEAR OPERATOR ANSWER
3911 043060  004737  032150                             JSR     PC,DRVTST               ;GO RUN THE INTERNAL DRIVE TEST
3912 043064  012702  002424                             MOV     @TSTNAM,R2              ;GET ADDRESS OF DRIVE TEST NAME
3913 043070  112722  000040                             MOVB    #40,(R2)+               ;RETURN DRIVE TEST NAME TO ASCII SPACES
3914 043074  112712  000040                             MOVB    #40,(R2)
3915 043100                                     T13EXT: EXIT    TST
     043100  104432                                     TRAP    C$EXIT
     043102  000002                                     .WORD   L10035-.
3916 043104                                             ENDTST
```

```
        043104                          L10035:
        043104  104401                          TRAP    C$ETST
   3917 043106                                  ENDMOD
   3918
   3919                                 .TITLE PARAMETER CODING
   3930
   3931                                 .SBTTL  HARDWARE PARAMETER CODING SECTION
   3959
   3960 043106                                  BGNMOD
   3961
   3962                                 ;++
   3963                                 ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
   3964                                 ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
   3965                                 ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
   3966                                 ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
   3967                                 ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
   3968                                 ; WITH THE OPERATOR.
   3969                                 ;--
   3970
   3971 043106                                  BGNHRD
        043106  000044                          .WORD L10036-L$HARD/2
        043110                          L$HARD::
   3972
   3978
   3979 043110                                  GPRMA   TUIPAD,0,0,160002,177564,YES
        043110  000031                          .WORD   T$CODE
        043112  043146                          .WORD   TUIPAD
        043114  160002                          .WORD   T$LOLIM
        043116  177564                          .WORD   T$HILIM
   3980 043120                                  GPRMD   TUVECT,2,0,777,60,776,YES
        043120  001032                          .WORD   T$CODE
        043122  043163                          .WORD   TUVECT
        043124  000777                          .WORD   777
        043126  000060                          .WORD   T$LOLIM
        043130  000776                          .WORD   T$HILIM
   3981 043132                                  GPRMD   TUUNT,4,0,777,0,251,YES
        043132  002032                          .WORD   T$CODE
        043134  043175                          .WORD   TUUNT
        043136  000777                          .WORD   777
        043140  000000                          .WORD   T$LOLIM
        043142  000251                          .WORD   T$HILIM
   3982
   3983 043144                                  EXIT HRD
        043144  026004                          .WORD   T$CODE
   3984
   3985 043146     124     125     111  TUIPAD: .ASCIZ  ?TUIP ADDRESS?
   3986 043163     124     125     040  TUVECT: .ASCIZ  ?TU VECTOR?
   3987 043175     124     057     115  TUUNT:  .ASCIZ  ?T/MSCP UNIT NUMBER?
   3988                                         .EVEN
   3989
   3990
   3991 043220                                  ENDHRD
                                                .EVEN
        043220                          L10036:
   3992
   3999
```

```
                                .SBTTL  SOFTWARE PARAMETER CODING SECTION

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

043220                          BGNSFT
043220  000000                  .WORD L10037 L$SOFT/2
043222                  L$SOFT::

                                .EVEN

                                ENDSFT
                                .EVEN
043222                  L10037:


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

060000                          .=60000         ;START OF THE THIRD 8KBYTE BLOCK
                                                ;OF VIRTUAL MEMORY SPACE.  ACCESIBLE
                                                ;VIA PAR/PDR 2.
060000                  RDBUF::
060000                  COMMBF::
060000                          .BLKW   20.     ;BUFFER SPACE PRECEDING COMM AREA
060050                  COMMAR::
060050                          .BLKW   514.    ;MAXIMUM COMM AREA LENGTH
062054                  LASTBF::
062054                          .BLKW   20.     ;BUFFER SPACE SUCCEEDING COMM AREA

062124                          LASTAD
                                .EVEN
062124  000000                  .WORD   0
062126  000000                  .WORD   0
062130                  L$LAST::
062130                          ENDMOD
000001                  .END
```

Symbol table

```
ABORT   002466          CKCMEX  031614          C$INLP= 000020          EMSG5   024140 G        G$RADA= 000140
ADR     = 000020 G      CLSDRV  032350 G        C$MANI= 000050          EMSG6   024161 G        G$RADB= 000000
ANSWER  002354 G        CMARLG  002326 G        C$MAP = 000102          EMSG7   024212 G        G$RADD= 000040
ASSEMB= 000010          CMDCNT  002746 G        C$MEM = 000031          EMSG8   024233 G        G$RADL= 000120
BAKPAT  031426 G        CMDREF  002744 G        C$MMU = 000103          EMSG9   024271 G        G$RADO= 000020
BIT0    = 000001 G      CMDRNG  002726 G        C$MSG = 000023          ENCODE  023734 G        G$XFER= 000004
BIT00 = 000001 G        CMDSAV  022750 G        C$OPNR= 000034          END     033706          G$YES = 000010
BIT01 = 000002 G        CMMERR  002322 G        C$OPNW= 000104          ENDCLE  033754          HELP  = 000000
BIT02 = 000004 G        CMPTBL  002302 G        C$PNTB= 000014          ERR   = 100000 G        HIADDR= 000002 G
BIT03 = 000010 G        CMTBLS  002324 G        C$PNTF= 000017          EVL   = 000004 G        HOE   = 100000 G
BIT04 = 000020 G        CNTER = 000000 G        C$PNTS= 000016          EXELOC  002410          HSTIMO= 000000 G
BIT05 = 000040 G        CNTFLG  002740 G        C$PNTX= 000015          EXT     036116          IBE   = 010000 G
BIT06 = 000100 G        CNTHI   002736 G        C$PUTB= 000072          EXTINT  030672          IDU   = 000040 G
BIT07 = 000200 G        CNTRLC= 000003 G        C$PUTW= 000073          EXTVEC  031300          IER   = 020000 G
BIT08 = 000400 G        COMMAR  060050 G        C$QIO = 000377          E$END = 002100          ILLINT  030674 G
BIT09 = 001000 G        COMMBF  060000 G        C$RDBU= 000007          E$LOAD= 000035          ILOOP   031746
BIT1  = 000002 G        CONID = 177777 G        C$REFG= 000047          FAULTC  024066 G        IMM   = 000200 G
BIT10 = 002000 G        CPFLAG  002362 G        C$REL = 000077          FLAG  = 040000 G        IMSG    033712
BIT11 = 004000 G        CPFLG = ****** GX       C$RESE= 000033          FLAGS   024047 G        INISTP  002336 G
BIT12 = 010000 G        CRD   = 177776 G        C$REVI= 000004          FRUERR  030624 G        INNER   002344 G
BIT13 = 020000 G        CREFNO  023436 G        C$RFLA= 000021          FRUIS   002330 G        INTFLG= 000002 G
BIT14 = 040000 G        CTRL    025647 G        C$RPT = 000025          F$AU  = 000015          INTMMU  031616 G
BIT15 = 100000 G        C$AU  = 000052          C$SEFG= 000046          F$AUTO= 000020          INTMSG  030540 G
BIT2  = 000004 G        C$AUTO= 000061          C$SPRI= 000041          F$BGN = 000040          INTRCV  030664 G
BIT3  = 000010 G        C$BRK = 000022          C$SVEC= 000037          F$CLEA= 000007          INTTBL  032140
BIT4  = 000020 G        C$BSEG= 000004          C$TOME= 000076          F$DU  = 000016          INVMSG  030604 G
BIT5  = 000040 G        C$BSUB= 000002          DFPTBL  002224 G        F$END = 000041          ISR   = 000100 G
BIT6  = 000100 G        C$CLCK= 000062          DIAGMC= 000000          F$HARD= 000004          ITRCNT= ****** GX
BIT7  = 000200 G        C$CLEA= 000012          DISCAC= 000014 G        F$HW  = 000013          IXE   = 004000 G
BIT8  = 000400 G        C$CLOS= 000035          DONEFL= 000020 G        F$INIT= 000006          I$AU  = 000041
BIT9  = 001000 G        C$CLP1= 000006          DRPFLG= 000001 G        F$JMP = 000050          I$AUTO= 000041
BOE   = 000400 G        C$CPBF= 000074          DRVE    025716 G        F$MOD = 000000          I$CLN = 000041
BRFLAG= 000004 G        C$CPME= 000075          DRVER = 000011 G        F$MSG = 000011          I$DU  = 000041
BUFDES  023653 G        C$CVEC= 000036          DRVTST  032150          F$PROT= 000021          I$HRD = 000041
BYTCNT  023627 G        C$DCLN= 000044          DSCEND  002736 G        F$PWR = 000017          I$INIT= 000041
B.DI  = 000400 G        C$DODU= 000051          DSCRNG  002712 G        F$RPT = 000012          I$MOD = 000041
B.ER  = 100000 G        C$DRPT= 000024          EF.CON= 000036 G        F$SEG = 000003          I$MSG = 000041
B.GO  = 000001 G        C$DU  = 000053          EF.NEW= 000035 G        F$SOFT= 000005          I$PROT= 000040
B.IE  = 000200 G        C$EDIT= 000000          EF.PWR= 000034 G        F$SRV = 000010          I$PTAB= 000041
B.LF  = 000002 G        C$ERDF= 000055          EF.RES= 000037 G        F$SUB = 000002          I$PWR = 000041
B.MP  = 000100 G        C$ERHR= 000056          EF.STA= 000040 G        F$SW  = 000014          I$RPT = 000041
B.NV  = 002000 G        C$ERRO= 000060          ELPERR  027306 G        F$TEST= 000001          I$SEG = 000041
B.OD  = 000200 G        C$ERSF= 000054          EMSG10  024316 G        GDSERR  030256 G        I$SETU= 000041
B.PI  = 000001 G        C$ERSO= 000057          EMSG11  024344 G        GDUST   002370          I$SFT = 000041
B.PP  = 100000 G        C$ESCA= 000010          EMSG12  024405 G        GO    = 000001 G        I$SRV = 000041
B.QB  = 001000 G        C$ESEG= 000005          EMSG13  024454 G        G$CNTO= 000200          I$SUB = 000041
B.S1  = 004000 G        C$ESUB= 000003          EMSG14  024503 G        G$DELM= 000372          I$TST = 000041
B.S2  = 010000 G        C$ETST= 000001          EMSG15  024532 G        G$DISP= 000003          J$JMP = 000167
B.S3  = 020000 G        C$EXIT= 000032          EMSG16  024567 G        G$EXCP= 000400          KPAR0 = 172340 G
B.S4  = 040000 G        C$FREQ= 000101          EMSG17  024637 G        G$HILI= 000002          KPAR1 = 172342 G
B.WR  = 040000 G        C$FRME= 000100          EMSG18  024701 G        G$LOLI= 000001          KPAR2 = 172344 G
CCR   = 177746 G        C$GETB= 000026          EMSG19  024743 G        G$NO  = 000000          KPAR3 = 172346 G
CDRECV  032556 G        C$GETW= 000027          EMSG20  025013 G        G$OFFS= 000400          KPAR4 = 172350 G
CHKCAC  030704 G        C$GMAN= 000043          EMSG21  025052 G        G$OFSI= 000376          KPAR5 = 172352 G
CHKCOM  031456 G        C$GPHR= 000042          EMSG22  025104 G        G$PRMA= 000001          KPAR6 = 172354 G
CHKMSG  033326          C$GPRI= 000040          EMSG23  025135 G        G$PRMD= 000002          KPAR7 = 172356 G
CHKRSP  033042          C$INIT= 000011          EMSG24  025215 G        G$PRML= 000000          KPDR0 = 172300 G
```

Symbol table

```
KPDR1 = 172302 G    L$ICP   002104 G    MMUSR0= 177572 G    PRIO6 = 000300 G    TSTNAM  002424
KPDR2 = 172304 G    L$INIT  033464 G    MMUSR1= 177574 G    PRIO7 = 000340 G    TUIP  = 000000 G
KPDR3 = 172306 G    L$LADP  002026 G    MMUSR2= 177576 G    PROGRH  002360 G    TUIPAD  043146
KPDR4 = 172310 G    L$LAST  062130 G    MMUSR3= 172516 G    PROGRL  002356 G    TUIPSV= 000010 G
KPDR5 = 172312 G    L$LOAD  002100 G    MM22ON= 000020 G    PRTDRV  032456 G    TUSA  = 000002 G
KPDR6 = 172314 G    L$LUN   002074 G    MODIFY  023520 G    PRTINT  031716 G    TUSASV= 000012 G
KPDR7 = 172316 G    L$MREV  002050 G    MSCPUN= 000006 G    P.BCNT= 000014 G    TUUNT   043175
KTEXT   031146      L$NAME  002000 G    MSCPVR= 000000 G    P.BUFF= 000020 G    TUVEC = 000004 G
KTFLAG  002314 G    L$PRIO  002042 G    MSGLEN= 177774 G    P.CRF = 000000 G    TUVECT  043163
KTTEST  030776 G    L$PROT  022760 G    NEXT    033604      P.ENDC= 000010 G    TXFER = 000005 G
LASTBF  062054 G    L$PRT   002112 G    NOKT    031142      P.FLGS= 000017 G    T$ARGC= 000001
LESI    025632 G    L$REPP  002062 G    NUPASS  033570      P.IND1= 000020 G    T$CODE= 026004
LINE1   022774 G    L$REV   002010 G    ONEFIL= 000001      P.IND2= 000022 G    T$ERRN= 000030
LINE2   023030 G    L$RPT = ****** GX   OPCODE  023500 G    P.MOD = 000012 G    T$EXCP= 000000
LINE3   023110 G    L$SOFT  043222 G    OP.ABT= 000006 G    P.OPCD= 000010 G    T$FLAG= 000041
LINE4   023140 G    L$SPC   002056 G    OP.ELP= 000003 G    P.STS = 000012 G    T$GMAN= 000000
LINE5   023203 G    L$SPCP  002020 G    OP.END= 000200 G    P.TIMO= 000024 G    T$HILI= 000251
LINE6   023260 G    L$SPTP  002024 G    OP.GDS= 000001 G    QUESTN  026716 G    T$LAST= 000001
LINE7   023323 G    L$STA   002030 G    OP.REC= 000005 G    RBUF  = 177562 G    T$LOLI= 000000
LOE   = 040000 G    L$SW    002234 G    OUTER   002346 G    RCSR  = 177560 G    T$!SYM= 010000
LOGUNT  002332 G    L$TEST  002114 G    OWN   = 100000 G    RCVDAT  002436      T$LTNO= 000015
LOOP    031736      L$TIML  002014 G    O$APTS= 000000      RCVERR  027726 G    T$NEST= 177777
LOT   = 000010 G    L$UNIT  002012 G    O$AU  = 000000      RDBUF   060000 G    T$NS0 = 000000
LSCT    025670      L10000  002232      O$BGNR= 000001      RESPBF  002502 G    T$NS1 = 000005
LUNBLK  002234 G    L10001  002234      O$BGNS= 000000      RNGSTP= 000004 G    T$NS2 = 000002
LUNFLG= 000014 G    L10003  030654      O$DU  = 000001      RSPBUF  002506 G    T$PTNU= 000000
L$ACP   002110 G    L10004  030662      O$ERRT= 000001      RSPEND  002716 G    T$SAVL= 177777
L$APT   002036 G    L10005  030672      O$GNSW= 000000      RSPRNG  002716 G    T$SEGL= 177777
L$AU    034010 G    L10006  030702      O$POIN= 000001      RSPSAV  022752 G    T$SUBN= 000000
L$AUT   002070 G    L10007  033742      O$SETU= 000000      RSPSTP= 000104 G    T$TAGL= 177777
L$AUTO= ****** GX   L10010  033772      PAROFF  002320 G    RSTVEC  031162 G    T$TAGN= 010040
L$CCP   002106 G    L10011  034006      PASCNT  002312 G    SAEXP   002334 G    T$TEMP= 000000
L$CLEA  033744 G    L10012  034014      PCKSIZ  002742 G    SELTST  026646 G    T$TEST= 000015
L$CO    002032 G    L10013  034512      PDELAY  031302 G    SFPTBL  002234 G    T$TSTM= 177777
L$DEPO  002011 G    L10014  034202      PDLYEX  031326      START   033536      T$TSTS= 000001
L$DESC  002156 G    L10015  034446      PDRECV  032670 G    STATUS  023755 G    T$$AU = 010012
L$DESP  002076 G    L10016  034742      PKRECV  023706 G    STEPST  002340 G    T$$CLE= 010010
L$DEVP  002060 G    L10017  035360      PKSENT  023414 G    STEP1   031330 G    T$$DU = 010011
L$DISP  002124 G    L10020  036122      PNT   = 001000 G    STPTBL  002272 G    T$$HAR= 010036
L$DLY   002116 G    L10021  037254      PRGNAM  023543 G    STP1ER  031420      T$$HW = 010000
L$DTP   002040 G    L10022  036650      PRGVER  023775 G    STP1EX  031424      T$$INI= 010007
L$DTYP  002034 G    L10023  037252      PRI   = 002000 G    ST5EXT  037242      T$$MSG= 010003
L$DU    033774 G    L10024  040700      PRIDAT  027142 G    ST6EXT  040666      T$$PRO= 010002
L$DUT   002072 G    L10025  040026      PRIERR  027222 G    SVCGBL= 000000      T$$SOF= 010037
L$DVTY  022766 G    L10026  040676      PRIEX   030650      SVCINS= 000000      T$$SRV= 010006
L$EF    002052 G    L10027  041442      PRIINI  027002 G    SVCSUB= 000000      T$$SUB= 010026
L$ENVI  002044 G    L10030  042204      PRIIP   027172 G    SVCTAG= 000000      T$$SW = 010001
L$ERRT= ****** GX   L10031  042266      PRIO7 = ****** GX   SVCTST= 000000      T$$TES= 010035
L$ETP   002102 G    L10032  042450      PRIPAD  027066 G    S$LSYM= 010000      T1      034016 G
L$EXP1  002046 G    L10033  042572      PRISA   027026 G    S1    = 004000 G    T1.1    034044
L$EXP4  002064 G    L10034  042714      PRIVAD  027114 G    TEMP    002352 G    T1.2    034230
L$EXP5  002066 G    L10035  043104      PRIO0 = 000000 G    TEST.9= 000010 G    T10     042270 G
L$HARD  043110 G    L10036  043220      PRIO1 = 000040 G    TF.BLK= 000010 G    T10EXT  042444
L$HIME  002120 G    L10037  043222      PRIO2 = 000100 G    TIMOUT  024026 G    T10MS1  025724 G
L$HPCP  002016 G    MANTBL  022754 G    PRIO3 = 000140 G    TOUT    002350 G    T10MS2  026032 G
L$HPTP  002022 G    MMON  = 000001 G    PRIO4 = 000200 G    TRAP4   030656 G    T10MS3  026057 G
L$HW    002224 G    MMSG    026536 G    PRIO5 = 000240 G    TRP4FG  002316 G    T10MS4  026134 G
```

Symbol table

```
T11     042452 G    T2EXT   034736      T6EXT   040022      UAM   = 000200 G    WRER6   025452 G
T11EXT  042566      T3      034744 G    T6.1    037256      VECTOR  031212 G    WRER7   025534 G
T11MS1  026221 G    T3EXT   035354      T6.2    040030      VEC4  = 000004 G    WRINTO  027226 G
T12     042574 G    T4      035362 G    T7      040702 G    WRBUF   002750 G    WRPRTE  027252 G
T12EXT  042710      T4EXT   036102      T7EXT   041436      WRDATA  002342 G    WR1     023350 G
T12MS1  026324 G    T5      036124 G    T8      041444 G    WRER1   025250 G    X$ALWA= 000000
T13     042716 G    T5EXT   036640      T8EXT   042200      WRER2   025302 G    X$FALS= 000040
T13EXT  043100      T5.1    036124      T9      042206 G    WRER3   025323 G    X$OFFS= 000400
T13MS1  026430 G    T5.2    036652      T9EXT   042262      WRER4   025351 G    X$TRUE= 000020
T2      034514 G    T6      037256 G    T9FLAG= ****** GX   WRER5   025375 G

. ABS.  062130    000   (RW,I,GBL,ABS,OVR)
        000000    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 291
Work  file writes: 299
Size of work file: 34376 Words  ( 135 Pages)
Size of core pool: 19684 Words  ( 75 Pages)
Operating  system: RSX-11M/PLUS (Under VAX/VMS)

Elapsed time: 00:14:50.59
CZTU28.BIN,CZTU28/-SP=SVC40R.MLB/ML,CZTU28
```