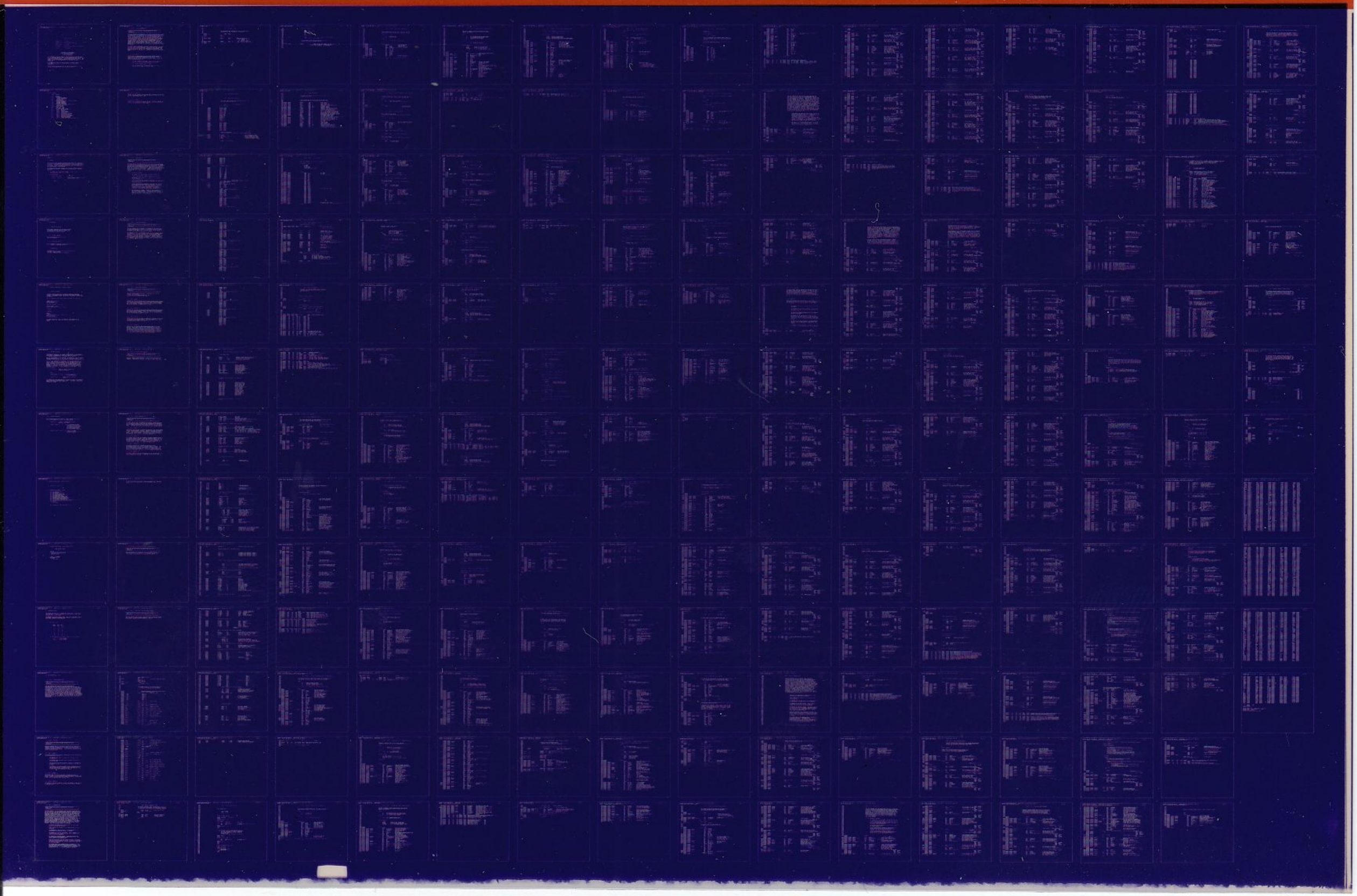
TK25

TK25 FRT END FUNC #1
CZTKEB0

AH-T776B-MC
1 OF 1 OCT 1985

digital
MADE IN USA



.REM\

IDENTIFICATION

PRODUCT ID: AC-T775B-MC

PRODUCT TITLE: CZTKEB0 TK25 FRT END FUNC #1

PRODUCT DATE: 7 MAY 1985

DEPARTMENT: TAPE AND OPTICAL DIAGNOSTIC ENGINEERING

AUTHOR: RAYMOND CHANG

TABLE OF CONTENTS

TABLE OF CONTENTS

ABSTRACT

1.0 ABSTRACT

THIS IS A PDP-11/LSI RESIDENT DIAGNOSTIC WHICH CHECKS THE FUNCTIONALITY OF AN TK25 MAGTAPE SUBSYSTEM WHILE CONNECTED TO A PDP-11 SYSTEM (Q-BUS OR UNIBUS). THE PROGRAM HAS BEEN DIVIDED INTO FOUR MAJOR PIECES: CZTKE, CZTKF, CZTKG, CZTKH. SUCCESSFUL RUN EXAMPLES, AND TEST DESCRIPTIONS HAVE BEEN PROVIDED FOR EACH PROGRAM.

THE PROGRAMS PROVIDE ERROR MESSAGES WHICH IDENTIFY FAILING FUNCTIONS, AND AID IN DEVICE REPAIR. REFERENCE THE FOLLOWING DIGITAL EQUIPMENT DOCUMENTS:

1. CIQPMA0 XXDP+ PROGRAMMER'S MANUAL; DOCUMENT NUMBER AC-S296A-AC; DATE: 14 JULY 1980.

1.1 REVISION HISTORY

CZTKEA0 (APRIL 1984)	NEW RELEASE
CZTKEB0 (7 JUNE 1984)	REMOVE INTERRUPT PRIORITY HARDWARE QUESTION FROM P TABLE.

2.0 REQUIREMENTS

2.1 HARDWARE REQUIREMENTS

PDP-11 FAMILY PROCESSOR WITH 32K WORDS OF MEMORY
TK25 MAGTAPE SUBSYSTEM (DRIVE AND CONTROLLER)
CAUTION:DIAGNOSTIC REQUIRES 32K WORDS OF MEMORY
(28K USEABLE I.E. 4K FOR I/O PAGE)

2.1.1 OPTIONAL HARDWARE

FOUR TK25 CONTROLLERS PER PDP-11, ONE
DRIVE PER CONTROLLER

2.2 SOFTWARE REQUIREMENTS

PDP-11 DIAGNOSTIC SUPERVISOR (CIQPMA0 VERSION 34 OR LATER)
PDP 11 DIAGNOSTIC LOADER/MONITOR (XXDP+)

2.3 PREREQUISITES

FUNCTIONAL PDP-11/LSI FAMILY CENTRAL PROCESSOR AND MEMORY
FUNCTIONAL CONSOLE TERMINAL
FUNCTIONAL STANDALONE DIAGNOSTIC SUPERVISOR

3.0 OPERATING INSTRUCTIONS - OPERATOR COMMANDS

3.1 OPERATOR COMMANDS

THE TK25 DIAGNOSTICS ARE PDP-11 DIAGNOSTIC SUPERVISOR COMPATIBLE PROGRAMS. ALL LOADING AND RUN TIME INSTRUCTIONS CAN BE REFERENCED IN THE PDP-11 PROGRAMMER'S MANUAL "CIQPMA0 XXDP" PROGRAMMER'S MANUAL NUMBER AC-S296A-AC.

BOOT THE DIAGNOSTIC XXDP+ MEDIA (OPERATOR RESPONSES ARE UNDERLINED)

```
CHMDLE0 XXDP+ DL MONITOR
BOOTED VIA UNIT 0
28K NON-UNIBUS SYSTEM

ENTER DATE <DD-MMM-YY>: 29-JAN 82
                        ---------
RESTART ADDRESS: 152010
THIS IS XXDP+ TYPE "H" OR "H/L" FOR HELP.

.R CZTKEA
 - ------

CZTKEA.BIC

DRS-E0
CZTKE-A-0
CZTKEA TK-25 FRT END FUNC #1 UNIT IS TK25
RSTRT ADR 147642
DR>START/FLAG:PNT:H0E
   -----------------
```

THE ABOVE COMMAND WILL START THE DIAGNOSTIC. THE COMMAND HAS TWO SWITCHES ON WHICH ARE "PRINT EACH TEST NBR. AS EXECUTED" AND "HALT ON ERROR".

3.2 HARDWARE PARAMETERS

AFTER INITIAL STARTING OF THE PROGRAM (START COMMAND TO THE DIAGNOSTIC SUPERVISOR), THE PROGRAM WILL ISSUE THE "CHANGE HW?" QUESTION TO ASK IF THE HARDWARE PARAMETERS ARE TO BE CHANGED (BY THE OPERATOR).

ON A "N" (NO) RESPONCE TO THE QUESTION, THE PROGRAM WILL USE IT'S DEFAULT HARDWARE PARAMETER VALUES. IT WILL DEFAU'T TO ONE UNIT SELECTED (UNIT 0), THE DEFAULT TSBA/TSDB WILL BE 172522 AND THE INTERRUPT VECTOR WILL BE 224.

ON A "Y" (YES) RESPONSE TO THE QUESTION, THE FOLLOWING QUESTIONS WILL THEN BE ASKED TO ALLOW THE OPERATOR TO SELECT THE UNITS TO BE TESTED. A VALUE, IF PRESENT, LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ONLY IF A CARRIAGE RETURN IS TYPED AS A RESPONSE. A "(D)" IN A QUESTION INDICATES THAT A DECIMAL NUMBER IS REQUIRED AS A RESPONSE. AN "(O)" INDICATES AN OCTAL NUMBER IS BEING SOLICITED. AN "(L)" THAT A LOGICAL RESPONSE IS TO BE MADE: "Y" FOR YES, "N' FOR NO.

```
UNITS (D) ? < ENTER THE NUMBER OF CONTROLLERS
              PRESENT TO BE TESTED>


UNIT 0

DEVICE ADDRESS (O) 172522 ? <ENTER THE ADDRESS OF THE
                                TSBA/TSDB REGISTER>

VECTOR (O) 224 ?        <ENTER ADDRESS OF INTERRUPT
                        VECTOR>
```

THE ADDRESS AND VECTOR QUESTIONS WILL BE ASKED FOR EACH OF THE NUMBER OF UNITS (CONTROLLERS) SPECIFIED IN THE " UNITS ?" QUESTION. LOGICAL UNIT NUMBERS ARE ASSIGNED IN ORDER BEGINNING AT 0. UP TO EIGHT UNITS CAN BE SELECTED FOR TESTING.

3.3 SOFTWARE PARAMETERS

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

CHANGE SW (L) ? < TYPE "Y" TO CAUSE THE FOLLOWING
QUESTIONS TO BE ASKED.>

INHIBIT ITERATIONS (L) N ? < TYPE "Y" TO PREVENT MULTIPLE
ITERATIONS OF CERTAIN TESTS.
THIS CAUSES EACH TEST PASS TO
RUN AS QUICKLY AS POSSIBLE.
ONLY QUICK-RUNNING LOGIC
TESTS USE MULTIPLE ITERATIONS.>

ENABLE RAM DUMP ON ERROR (L) N? < TYPE "Y" TO DUMP
SELECTED RAM CONTENTS IN THE
CONTROLLER MODULE.>

4.0 OPERATING INSTRUCTIONS - SAMPLE PRINTOUTS

4.1 SUCCESSUL RUN EXAMPLES

4.1.1 SUCCESSFUL RUN EXAMPLE - CZTKE -

```
TST: 001 INITIALIZATION TEST
TST: 002 RAM TEST
TST: 003 COMMAND REJECT TEST
TST: 004 WRITE CHARACTERISTICS TEST
TST: 005 VOLUME CHECK TEST
TST: 006 COMPLETION INTERRUPT TEST
TST: 007 BASIC PACKET PROTOCAL TEST
TST: 008 NON-TAPE MOTION COMMANDS TEST
TST: 009 DMA MEMORY ADDRESSING TEST
TST: 010 INITIALIZATION AFTER WRITE CHARACTERISTICS TEST
TST: 011 BASIC WRITE SUBSYSTEM MEMORY TEST
CZTKE EOP 1
    0 TOTAL ERRS
```

NOTE: PROGRAM NOW STARTS OVER AGAIN AT TEST 1

4.2 OPERATING INSTRUCTIONS - SAMPLE ERROR MESSAGE

ERROR MESSAGE EXAMPLE

```
TST: 001
CZTKE DVC FTL ERR  00001 ON UNIT 00 TST 001 SUB 000 PC: 017300
  NON-EXISTANT DEVICE REGISTER
  ADDRESS: 172500


  UNIT     0 DROPPED
PASS ABRTD THS UNIT
CZTKE EOP    1
    1 TOTAL ERRS
```

5.0 PROGRAM RUN TIMES

THE AVERAGE RUN TIMES OF THE PROGRAMS ARE LISTED BELOW. THESE FIGURES ARE TO BE USED AS A GUIDE. THE TIMING WAS DONE ON A PDP-11/23 (LSI) PROCESSOR WITH A LA-120 CONSOLE.

THE PROGRAMS RUN IN NON-ITERATIVE MODE. EACH TEST IS RUN ONCE, WITH NO ITERATIONS. THEREFOR, THE DEFAULT MODE (NORMALLY ITERATIVE) AND THE NON-ITERATIVE MODE TIMES ARE IDENTICAL.

5.1 RUN TIMES - CZTKE

TEST NUMBER	N/I SECS.	DEF SECS.
1	8	8
2	11	11
3	9	9
4	10	10
5	12	12
6	10	10
7	2	2
8	8	8
9	5	5
10	8	8
11	13	13

THE TIMES REQUIRED TO RUN TESTS 1 THROUGH 11 IN ONE COMMAND:

Q.V.	1 MIN 46 SECONDS
DEFAULT	1 MIN 46 SECONDS

6.0 TEST DESCRIPTIONS - CZTKE

6.1 TEST 1 - INITIALIZATION TEST 1

```
**************************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TK25'S                 *
* CONTROLLER                                                             *
**************************************************************************
```

THIS TEST VERIFIES THAT THE MODULE'S DEVICE REGISTERS ARE ACCESSIBLE ON THE BUS (SUBTEST 1) AND THEN CHECKS THAT THE BUILT-IN INITIALIZATION SELF-TEST MICRODIAGNOSTIC DID NOT FIND ANY BASIC PROBLEMS WITH THE MODULE. AREAS OF LOGIC TESTED BY THE SELF-TEST SEQUENCER ARE AS FOLLOWS: ROM AND PIPELINE REGISTER, SEQUENCER, INTERNAL BUSES, 2901 MICROPROCESSOR, RAM AND TRANSPORT STATUS FLOPS. THIS TEST INITIALIZES THE CONTROLLER BY ISSUING THE BUS INIT SIGNAL VIA A RESET INSTRUCTION, OR BY WRITING INTO THE TSSR REGISTER, AND THEN CHECKS THE CONTENTS OF THE TSSR REGISTER. SUCCESSFUL INITIALIZATION IS INDICATED BY SUBSYSTEM READY (SSR) AND NEED BUFFER ADDRESS (NBA) BITS BEING SET (1) AND ALL OTHER BITS (EXCEPT A17, A16, AND OFL, WHICH ARE IGNORED FOR THIS TEST) BEING CLEARED (0). IF THE CONTENTS OF THE TSSR ARE NOT AS EXPECTED, AN ERROR REPORT IS ISSUED LISTING THE EXPECTED DATA, ACTUAL DATA, AND THE DISCREPANCIES.

6.2 TEST 2 - RAM TEST

```
**************************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TK25'S                 *
* CONTROLLER                                                             *
**************************************************************************
```

THIS TEST VERIFIES THAT ALL LOCATIONS OF THE RAM ON THE CONTROLLER CAN PROPERLY STORE AND READ BACK ALL DATA PATTERNS, AND THAT EACH RAM LOCATION IS UNIQUELY ADDRESSED (IE: THAT ONE AND ONLY ONE LOCATION IS ACCESSED BY ANY PARTICULAR ADDRESS). THESE TESTS ARE PERFORMED BY THREE SUBTESTS DESCRIBED BELOW.

6.2.1 TEST 2, SUBTEST 1: -

THIS SUBTEST VERIFIES EACH LOCATION BY PERFORMING THE FOLLOWING SEQUENCE FOR EACH ADDRESS 0-377 (OCTAL):

1. THE ADDRESS TO BE TESTED IS LOADED INTO THE TSDB+1 (VIA A HI-WRITE BYTE).
2. THE ADDRESSED RAM LOCATION IS READ, THEN WRITTEN INTO THE LOW BYTE OF THE TSBA.
3. THE LOW BYTE OF THE TSBA IS CHECKED TO SEE IF IT CONTAINS THE DATA PATTERN ORIGINALLY WRITTEN; A DISCREPANCY IS REPORTED AS AN ERROR.
4. THE ADDRESS OF THE LOCATION BEING TESTED IS AGAIN WRITTEN INTO TSDB+1 (HI-BYTE WRITE), TO CAUSE THE LOCATION UNDER TEST TO AGAIN BE READ INTO THE LOW BYTE OF TSBA. THE LOW BYTE OF TSBA IS AGAIN CHECKED AND DISCREPANCIES REPORTED.

6.2.2 TEST 2, SUBTEST 2: -

THIS SUBTEST USES THE SAME RAM READ/WRITE TECHNIQUES AS SUBTEST 1, EXCEPT THAT MEMORY IS FILLED WITH ZEROS AND A ONES WORD IS "WALKED" DOWN THROUGH. PRIOR TO THE ALL ONES WRITE TO MEMORY THE MEMORY IS CHECKED TO BE SURE THAT THE ZERO WORD HASN'T "PICKED" A BIT.

6.2.3 TEST 2, SUBTEST 3: -

THIS SUBTEST IS SIMILAR TO SUBTEST 2, EXCEPT THAT MEMORY IS FIRST SET TO ALL ONES AND A BYTE OF ZEROS IS "WALKED" DOWN THROUGH MEMORY BEGINNING AT LOCATION TOP-2.

6.3 TEST 3: COMMAND REJECT

```
**************************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TK25'S                 *
* CONTROLLER                                                             *
**************************************************************************
```

THIS TEST VERIFIES THAT ALL COMMANDS OTHER THAN WRITE CHARACTERISTICS ARE REJECTED DUE TO THE NEED BUFFER ADDRESS (NBA) BIT BEING SET IN TSSR, AND THAT THE TSBA, AND THE TSSR REGISTERS ARE SET IN THE PROPER STATE AFTER EACH COMMAND IS REJECTED. THIS TEST CHECKS THE MICROPROCESSOR SEQUENCING, BASIC COMMAND DECODING, AND DATI DMA HANDLING. THE TEST CONTAINS TWO SUBTESTS: SUBTEST 1 SEQUENCES THROUGH ALL COMMAND WORDS (OTHER THAN WRITE CHARACTERISTICS) WITH THE INTERRRUPT ENABLE BIT (IE) BIT CLEAR AND VERIFIES THAT AN INTERRUPT IS NOT GENERATED BY THE REJECTED COMMAND; SUBTEST TWO PERFORMS SIMILARLY TO SUBTEST 1 BUT SETS THE IE BIT IN EACH COMMAND WORD AND VERIFIES THAT AN INTERRUPT IS GENERATED WHEN THE COMMAND IS REJECTED. THE SUBTEST 1 SETS UP THE INTERRUPT SERVICE ROUTINE TO FLAG UNEXPECTED INTERRUPTS. THE COMMAND WORD IN THE COMMAND BUFFER IS INITIALIZED TO 100000 (OCTAL) AND THE REMAINING THREE WORDS IN THE COMMAND BUFFER ARE SET TO KNOWN UNIQUE PATTERNS. THEN THE FOLLOWING SEQUENCE IS PERFORMED:

1. INITIALIZE THE CONTROLLER BY WRITING INTO THE TSSR, PROPER INITIAL CONDITIONS ARE VERIFIED.

2. TSDB IS WRITTEN WITH ADDRESS OF THE COMMAND BUFFER TO START PROCESSING.

3. THE PROGRAM WAITS FOR SSR TO SET; IF SSR DOES NOT SET, AN ERROR REPORT IS ISSUED AND THE TEST IS ABORTED.

4. THE CONTENTS OF THE TSSR ARE CHECKED. TSSR IS CORRECT IF IT CONTAINS EITHER OCTAL 102206 OR 102306 (BIT 6 DEPENDS ON THE STATE OF THE TAPE TRANSPORT).

5. THE CONTENTS OF TSBA ARE CHECKED. TSBA SHOULD CONTAIN THE INITIAL COMMAND BUFFER ADDRESS (LOADED IN STEP 2) IE: TSBA SHOULD POINT TO THE COMMAND PACKET.

6. USING THE MAINTENANCE MODE WRAP AROUND FUNCTIONS, THE COMMAND IMAGE BLOCK IN THE CONTROLLER'S RAM 20 - 27 (OCTAL)) ARE CHECKED; THE SHOULD CONTAIN A COPY OF THE FOUR COMMAND PACKET WORDS AS SET UP IN CPU MEMORY.

7. THE COMMAND WORD IN THE COMMAND BUFFER IS INCREMENTED TO THE NEXT PATTERN NOT CONTAINING WRITE CHARACTERISTICS; E.G., THE REMAINING THREE WORDS OF THE COMMAND BUFFER ARE SEQUENCED WITH PSEUDO-RANDOM DATA. IF THE COMMAND WORD HAS NOT REACHED ITS MAXIMUM VALUE (17777+1) THE TEST SEQUENCE IS REPEATED.

6.4 TEST 4 - WRITE CHARACTERISTICS

```
**********************************************************************
* NOTE: IF THIS TEST DETECTS A FAILURE REPLACE THE TK25'S            *
* CONTROLLER                                                         *
**********************************************************************
```

THIS TEST VERIFIES BASIC OPERATION OF THE WRITE CHARACTERISTICS COMMAND. IT VERIFIES THAT THE COMMAND BLOCK AND CHARACTERISTICS DATA BLOCK ARE FETCHED PROPERLY FROM CPU MEMORY, THE NEED BUFFER ADDRESS (NBA) BIT IN THE TSSR IS HANDLED PROPERLY, AND THAT A PROPER MESSAGE PACKET IS STORED, WHERE APPROPRIATE. THIS TEST DOES NOT CHECK THAT THE VARIOUS FUNCTIONS ENABLED BY CHARACTERISTICS MODE DATA BITS OPERATE PROPERLY; THE FUNCTIONING OF THESE BITS IS VERIFIED IN SUBSEQUENT TESTS. ALL COMMANDS EXECUTED IN THIS TEST HAVE THE INTERRUPT ENABLE (IE) BIT CLEARED TO ZERO, SO NO INTERRUPTS SHOULD BE GENERATED. HOWEVER, THE PROGRAM RUNS AT PROCEESOR PRIORITY ZERO, WITH THE INTERRUPT SERVICE ROUTINE SET UP TO FLAG UNEXPECTED INTERRUPTS. IF AN INTERRUPT OCCURS A PROBLEM EXISTS IN EITHER THE LSI-11 BUS INTERFACE SECTION OR IN THE ROM OR PIPELINE.

THIS TESTS VARIOUS MICROPROGRAM SEQUENCES, COMMAND DECODING, DMA LOGIC, AND BASIC PACKET PROTOCAL HANDLING. THIS IS THE FIRST TEST IN WHICH DATA DMA CYCLES (FOR STORING THE MESSAGE PACKET) ARE PERFORMED. ANY ERRORS IN THE BODY OF THE TEST (IE: ERRORS OTHER THAN INITIALIZATION ERRORS RELATED TO THE TRANSPORT BUS) DEFINITELY INDICATES A BAD CONTROLLER MODULE.

6.4.1 TEST 4, SUBTEST 1: -

VERIFIES BASIC STANDARD OPERATION (USING PROPER MESSAGE BUFFER AND LENGTH DATA IN AN INCREMENTING SERIES OF VALUES FOR THE FOURTH CHARACTERISTICS DATA IN THE CHARACTERISTICS DATA BLOCK.). AFTER THE COMMAND IS EXECUTED FOR EACH VALUE OF THE FOURTH CHARACTERISTICS DATA WORD, THE PROGRAM VERIFIES THAT:

1. THE TSSR IS CORRECT, INCLUDING A CHECK THAT THE NBA BIT IS CLEARED AND THAT NORMAL TERMINATION WAS ACCOMPLISHED.

2. THAT A PROPER MESSAGE PACKET IS STORED.

3. THAT THE COMMAND PACKET CHRACTERISTIC DATA, AND MESSAGE PACKET IMAGE BLOCKS IN CONTROLLER RAM ARE CORRECT.

6.4.2 TEST 4, SUBTEST 2: -

VERIFIES THAT THE COMMAND IS REJECTED AND THAT THE NBA BIT DOES NOT GET CLEARED IF NONZERO BITS ARE SET INTO ANY RESERVED OR UNUSED FIELD WITHIN THE FIRST THREE COMMAND PACKET WORDS.

6.4.3 TEST 4, SUBTEST 3: -

VERIFIES THAT THE COMMAND IS REJECTED AND THAT THE NBA BIT DOES NOT GET CLEARED IF THE MESSAGE BUFFER ADDRESS SPECIFIED IN THE CHARACTERISTICS DATA BLOCK DOES NOT SPECIFY A LEGAL ADDRESS.

6.5 TEST 5: VOLUME CHECK

```
**************************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TK25'S                 *
* CONTROLLER                                                             *
**************************************************************************
```

THIS TEST VERIFIES THAT THE VOLUME CHECK (VCK) BIT, A FLAG HELD WITHIN THE CONTROLLER AND APPEARING IN XSTO, IS SET BY INITAILIZE AND CLEARED BY EXECUTING A WRITE CHARACTERISTICS COMMAND WITH THE CVC SET. IT IS ALSO VERIFIED THAT A WRITE CHARACTERISTICS WITH THE CVC BIT CLEAR DOES NOT AFFECT THE STATE OF THE VOLUME CHECK BIT. THE ACTUAL FUNCTION OF VOLUME CHECK, THAT OF PREVENTING OR ALLOWING A TAPE MOTIN COMMAND DEPENDING ON WHETHER VOLUME CHECK IS SET OR CLEAR , IS NOT CHECKED BY THIS TEST; THIS FUNCTIONALITY IS CHECKED IN THE INDIVIDUAL TESTS OF TAPE MOTION COMMANDS.

THE TEST PROCEEDS AS FOLLOWS:

1. THE CONTROLLER IS INITIALIZED BY WRITING INTO THE TSSR.
2. A WRITE CHARACTERISTICS COMMAND IS ISSUED (WITH CVC=0)
3. THE PREVIOUS STEP IS REPEATED TO VERIFY THAT VCK DOES NOT CHANGE (REMAINS AT 0).
4. A WRITE CHARACTERISTICS COMMAND IS ISUED WITH CVC=1 AND THE VCK BIT IN XSTO IN THE MESSAGE BUFFER IS EXAMINED; THE VCK BIT SHOULD BE CLEAR (0).
5. A WRITE CHARACTERISTICS COMMAND IS ISUED WITH CVC=0 AND THE VCK BIT IN XSTO IN THE MESSAGE BUFFER IS EXAMINED; THE VCK BIT SHOULD REMAIN CLEAR (0). THIS SHOULD CAUSE RAM LOCATION 0 TO BE WRITTEN TO ALL 1'S SINCE 2901 REGISTERS 10 AND 11, SPECIFYING THE RAM ADDRESS, SHOULD BE 0. RAM LOCATION IS VERIFIED BY LOW BYTE OF TSBA WHICH SHOULD CONTAIN ALL 1'S.
6. THE ENTIRE RAM IS SCANNED. LOCATION 0 SHOULD CONTAIN ALL 1'S AND THE REMAINING LOCATIONS, EXCEPT FOR THE MESSAGE BUFFER IMAGE AREA, SHOULD CONTAIN 0. DISCREPANCIES ARE REPORTED. AN ERROR AT THIS POINT IS MOST LIKELY DUE TO A ROM, PIPELINE OR SEQUENCER PROBLEM OR A TIMING PROBLEM.

6.6 TEST 6 - COMPLETION INTERRUPT

```
**************************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TK25'S                 *
* CONTROLLER                                                             *
**************************************************************************
```

THIS TEST VERIFIES THAT AN INTERUPT IS GENERATED AT THE COMPLETION OF THE WRITE CHARACTERISTICS COMMAND IF THE INTERRUPT ENABLE (IE) BIT IN THE COMMAND HEADER WORD IS SET. THIS TEST CHECKS THE FUNCTIONING OF THE INTERRUPT LOGIC AND BASIC PROCESSING OF THE IE BIT.

THE SEQUENCES OF TEST 7 ARE REPEATED, EXCEPT THAT THE INTERRUPT SERVICE ROUTINE IS SET UP TO EXPECT INTERRUPTS AND EACH WRITE CHARACTERISTICS COMMAND IS ISSUED WITH THE IE BIT SET (1). IT IS VERIFIED, WHERE APPROPRIATE , THAT THE STATUS BIT IN XST0 OF ANY MESSAGE PACKET IS SET AND THAT A COMPLETTION INTERRUPT IS GENERATED. FINALLY A SEQUENCE OF TWO COMMANDS IS ISSUED, THE FIRST WITH IE=1, AND THE SECOND WITH IE=0. IT IS VERIFIED THAT NO INTERRUPT IS GENERATED AFTER THE SECOND COMMAND AND THAT THE IE BIT IN XST0 IS 0.

6.7 TEST 7 - BASIC PACKET PROTOCAL

```
**************************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TK25'S                 *
* CONTROLLER                                                             *
**************************************************************************
```

THIS TEST VERIFIES BASIC OPERATION OF THE MESSAGE BUFFER RELEASE COMMAND, THE FUNCTION OF
THE ACK BIT IN THE COMMAND HEADER WORD, AND THE
REGISTER MODIFICATION REFUSED (RMR) LOGIC.

6.7.1 TEST 7, SUBTEST 1: -

VERIFIES THAT THE BASIC MESSAGE BUFFER RELEASE COMMAND OPERATES PROPERLY WHEN MESSAGE BUFFER RELEASE INTERRUPTS ARE DISABLED (ERI=0 ON PREVIOUS WRITE CHARACTERISTICS COMMAND). CHECKS THAT TSSR IS UPDATED PROPERLY AND THAT NO INTERRUPT IS GENERATED (EVEN IF THE IE BIT IN THE COMMAND WORD IS SET) AND THAT NO MESSAGE PACKET IS STORED.

6.7.2 TEST 7, SUBTEST 2: -

VERIFIES THAT THE BASIC MESSAGE BUFFER RELEASE COMMAND OPERATES PROPERLY WHEN MESSAGE BUFFER RELEASE INTERRUPTS ARE ENABLED (ERI=1 ON PREVIOUS WRITE CHARACTERISTICS COMMAND). CHECKS THAT TSSR IS UPDATED PROPERLY AND THAT AN INTERRUPT IS GENERATED (IF THE IE BIT IN THE COMMAND WORD IS SET) BUT THAT NO MESSAGE PACKET IS STORED.

6.7.3 TEST 7, SUBTEST 3: -

VERIFIES THAT AFTER THE CPU GIVES UP OWNERSHIP OF A MESSAGE BUFFER (VIA THE MESSAGE BUFFER RELEASE COMMAND), THAT A NEW COMMAND (E.G., WRITE CHARACTERISTICS) IS PROPERLY EXECUTED WHEN ISSUED WITH THE ACK BIT IN THE COMMAND HEADER EITHER SET OR CLEAR.

6.7.4 TEST 7, SUBTEST 4: -

VERIFIES THAT THE REGISTER VERIFICATION REFUSED (RMR) BIT IN TSSR OPERATES PROPERLY WHEN A COMMAND (WRITE CHARACTERISTICS) IS BEING EXECUTED. THE PROGRAM ISSUES THE WRITE CHARACTERISTICS COMMAND (FROM ONE COMMAND BUFFER) THEN IMMEDIATELY WRITES THE ADDRESS OF ANOTHER COMMAND BUFFER (CONTAINING ANOTHER WRITE CHARACTERISTICS COMMAND, BUT WITH THIS ONE SPECIFYING DIFFERENT CHARACTERISTICS DATA). WHEN SSR SETS, THE PROGRAM VERIFIES THAT THE FIRST COMMAND COMPLETED PROPERLY, THAT RMR IS SET, AND THAT THE SECOND COMMAND IS IGNORED.

6.8 TEST 8 - NON-TAPE MOTION COMMANDS

```
**********************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TK25'S             *
* CONTROLLER                                                         *
**********************************************************************
```

THIS TEST VERIFIES PROPER OPERATION OF THE GET STATUS AND INITIALIZE COMMANDS. THREE SUBTESTS ARE USED. THE FIRST TWO VERIFY THAT THE RESPECTIVE COMMANDS RUN TO COMPLETION AND STORE A VALID MESSAGE PACKET.

6.9 TEST 9 - MEMORY ADDRESSING TEST

```
********* \*****************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TK25'S       *
* CONTROLLER                                                   *
****************************************************************
```

THIS TEST VERIFIES THAT THE CONTROLLER CAN PROPERLY ADDRESS AND ACCESS ALL AVAILABLE CPU MEMORY (OTHER THAN THAT OCCUPIED BY THE DIAGNOSTIC AND THE DIAGNOSTIC SUPERVISOR CODE) FOR BOTH READING (DATI) AND WRITING (DATO). VERIFIED ARE THE PDP-11 BUS DRIVERS FOR ALL AVAILABLE ADDRESS LINES. UP TO THIS POINT ONLY 16 BITS HAVE BEEN USED FOR DMA TRANSFERS.

6.9.1 TEST 9, SUBTEST 1: -

THIS SUBTEST VERIFIES THAT THE CONTROLLER CAN FETCH A GET STATUS COMMAND FROM ALL AVAILABLE MEMORY LOCATIONS. TWO WORD BLOCKS ARE TESTED ONE AT A TIME BY FIRST SETTING ALL AVAILABLE MEMORY TO A BACKGROUND PATTERN OF 125252. A GET STATUS COMMAND IS THEN EXECUTED TO VARIOUS ADDRESSES IN EACH AVAILABLE MEMORY 4K BLOCK. THE VARIOUS ADDRESSES ARE DETERMINED BY FLOATING FIRST A (1) THEN A (0) THROUGH THE ADDRESS BITS.

6.9.2 TEST 9, SUBTEST 2: -

THIS SUBTEST VERIFIES THAT THE CONTROLLER CAN DEPOSIT MESSAGE PACKETS TO ALL AVAILABLE MEMORY LOCATIONS. FIRST ALL AVAILABLE MEMORY IS SET TO A BACKGROUND PATTERN OF 125252. WRITE CHARACTERISTICS COMMANDS ARE THEN EXECUTED WITH MESSAGE BUFFER ADDRESSES SET TO VARIOUS ADDRESSES IN EACH AVAILABLE MEMORY LOCATION. THE VARIOUS ADDRESSES ARE DETERMINED BY FLOATING FIRST A (1) THEN A (0) THROUGH THE ADDRESS BITS.

6.9.3 TEST 9, SUBTEST 3: -

THIS SUBTEST VERIFIES THAT A CONTROLLER CAN FETCH A WRITE CHARACTERISTICS DATA BLOCK FROM ALL AVAILABLE MEMORY LOCATIONS. FIRST ALL AVAILABLE MEMORY IS SET TO A BACKGROUND PATTERND OF 125252. THE WRITE CHARACTERISTICS COMMANDS ARE EXECUTED WITH CHARACTERISTIC DATA BLOCKS AT VARIOUS MEMORY ADDRESSES. THE VARIOUS MEMORY ADDRESSES ARE DETERMINED BY FLOATING FIRST A (1) THEN A (0) THROUGH THE ADDRESS BITS.

6.9.4 TEST 9, SUBTEST 4: -

THIS SUBTEST VERIFIES THAT THE NXM ERROR BIT IN THE TSSR REGISTER IS SET WHEN ATTEMPTING TO FETCH DATA (A CHARACTERISTIC DATA BLOCK) FROM SELECTED NONEXISTANT LOCATIONS. IF NXM FAILS TO SET IT IS LIKELY THAT

AN LSI-11 BUS DRIVER IS FAILING TO ASSERT AN ADDRESS LINE. ADDRESSES TESTED INCLUDE ALL COMBINATIONS OF HIGH ORDER ADDRESS BITS (IE: BITS 16-21).

6.10 TEST 10 - INITIALIZE AFTER WRITE CHARACTERISTICS

```
**********************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TK25'S             *
* CONTROLLER                                                         *
**********************************************************************
```

THIS TEST VERIFIES THAT A HARDWARE INITIALIZE COMMAND INVOKED AFTER A WRITE CHARACTERISTICS COMMAND SETS UP THE COMMAND, MESSAGE AND CHARACTERISTIC IMAGE BLOCK IN THE CONTROLLER RAM CORRECTLY.

6.11 TEST 11 - BASIC WRITE SUBSYSTEM MEMORY COMMAND

```
**********************************************************************
* NOTE: IF THIS TEST DETECTS AN ERROR REPLACE THE TK25'S             *
* CONTROLLER                                                         *
**********************************************************************
```

THIS TEST VERIFIES THAT THE WRITE SUBSYSTEM MEMORY COMMAND WITH A BSEL0 SELECT CODE OF 0 (NO-OP) EXECUTES CORRECTLY. THE TEST FURTHER VERIFIES MICROPROGRAM COMMAND DECODING AND HANDLING SEQUENCES.

```
                \
                                        .SBTTL  PROGRAM HEADER
                                        .MCALL  SVC
000000                                  SVC                     ; INITIALIZE SUPERVISOR MACROS
                                        .ENABLE LC
                                        .NLIST  BEX,CND
000000                                  .ENABL  AMA,ABS
        002000                          . = 2000
002000                                          BGNMOD  TUV2A
002000                                  TUV2A::

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


002000                                          POINTER BGNSU,BGNSFT,BGNAU,BGNDU,BGNRPT,BGNSETUP
002000                                          HEADER  CZTKE,B,0,655.,0
002000                                  L$NAME::                ;DIAGNOSTIC NAME
002000      103                                 .ASCII /C/
002001      132                                 .ASCII /Z/
002002      124                                 .ASCII /T/
002003      113                                 .ASCII /K/
002004      105                                 .ASCII /E/
002005      000                                 .BYTE   0
002006      000                                 .BYTE   0
002007      000                                 .BYTE   0
002010                                  L$REV::                 ;REVISION LEVEL
002010      102                                 .ASCII  /B/
002011                                  L$DEPO::                ;0
002011      060                                 .ASCII  /0/
002012                                  L$UNIT::                ;NUMBER OF UNITS
002012  000001                                  .WORD   T$PTHV
002014                                  L$TIML::                ;LONGEST TEST TIME
002014  001217                                  .WORD   655.
002016                                  L$HPCP::                ;POINTER TO H.W. QUES.
002016  046052                                  .WORD   L$HARD
002020                                  L$SPCP::                ;POINTER TO S.W. QUES.
002020  046200                                  .WORD   L$SOFT
002022                                  L$HPTP::                ;PTR. TO DEF. H.W. PTABLE
002022  002124                                  .WORD   L$HW
002024                                  L$SPTP::                ;PTR. TO S.W. PTABLE
002024  002134                                  .WORD   L$SW
002026                                  L$LADP::                ;DIAG. END ADDRESS
002026  046424                                  .WORD   L$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$DTP::                 ;PTR. TO DISPATCH TABLE
002040  046372                                  .WORD   L$DISPATCH
```

PROGRAM HEADER

```
002042                          L$PRIO::                ;DIAGNOSTIC RUN PRIORITY
002042  000000                          .WORD   0
002044                          L$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     003                          .BYTE   C$REVISION
002051     003                          .BYTE   C$EDIT
002052                          L$EF::                  ;DIAG. EVENT FLAGS
002052  000000                          .WORD   0
002054  000000                          .WORD   0
002056                          L$SPC::
002056  000000                          .WORD   0
002060                          L$DEVP::                ; POINTER TO DEVICE TYPE LIST
002060  003334                          .WORD   L$DVTYP
002062                          L$REPP::                ;PTR. TO REPORT CODE
002062  022674                          .WORD   L$RPT
002064                          L$EXP4::
002064  000000                          .WORD   0
002066                          L$EXP5::
002066  000000                          .WORD   0
002070                          L$AUT::                 ;PTR. TO ADD UNIT CODE
002070  022366                          .WORD   L$AU
002072                          L$DUT::                 ;PTR. TO DROP UNIT CODE
002072  022464                          .WORD   L$DU
002074                          L$LUN::                 ;LUN FOR EXERCISERS TO FILL
002074  000000                          .WORD   0
002076                          L$DESP::                ;POINTER TO DIAG. DESCRIPTION
002076  003342                          .WORD   L$DESC
002100                          L$LOAD::                ;GENERATE SPECIAL AUTOLOAD EMT
002100  104035                        EMT     E$LOAD
002102                          L$ETP::                 ;POINTER TO ERRTBL
002102  000000                          .WORD   0
002104                          L$ICP::                 ;PTR. TO INIT CODE
002104  021606                          .WORD   L$INIT
002106                          L$CCP::                 ;PTR. TO CLEAN-UP CODE
002106  022646                          .WORD   L$CLEAN
002110                          L$ACP::                 ;PTR. TO AUTO CODE
002110  022572                          .WORD   L$AUTO
002112                          L$PRT::                 ;PTR. TO PROTECT TABLE
002112  021576                          .WORD   L$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
```

```
                                .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 RUN-TIME P-TABLE.
                            ;--
002122                              BGNHW   DFPTBL          ;DEFAULT HARD-P-TABLE
002122  000003                      .WORD   L10000-L$HW/2
002124                      L$HW::
002124                      DFPTBL::

002124  172522                      .WORD   172522          ; 2ND (OF 2) REGISTERS.
002126  000224                      .WORD   224             ; INTERRUPT VECTOR
002130  000240                      .WORD   PRI05           ; INTERRUPT PRIORITY.
002132                              ENDHW
002132                      L10000:
```

```
                                .SBTTL  SOFTWARE P-TABLE

                        ;++
                        ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
                        ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
                        ;--
002132                          BGNSW   SFPTBL
002132  000004                 .WORD   L10001-L$SW/2
002134                  L$SW::
002134                  SFPTBL::

002134  000000          TRANSTST::      .WORD   0       ;ENABLE RAM DUMP IF =1
002136  000000          NOITS::         .WORD   0       ; INHIBIT ITERATION OPTION.
                                                        ; ... 0 = ITERATE.
                                                        ; ...NZ = INHIBIT ITERATE.
002140  000031          LERRMAX::       .WORD   25.     ; LOCAL (PER TEST) ERROR LIMIT
002142  000310          GERRMAX::       .WORD   200.    ; GLOBAL (PER UNIT) ERROR LIMIT
002144                          ENDSW
002144                  L10001:
```

```
805
812                                     .SBTTL  GLOBAL EQUATES SECTION
817
823
824
825
826                                     .SBTTL  GLOBAL EQUATES SECTION
827
828                                 ;++
829                                 ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
830                                 ; ARE USED IN MORE THAN ONE TEST.
831                                 ;--
932
836 002144                              EQUALS          ; GET STANDARD EQUATES.
                                    ;
                                    ; BIT DIFINITIONS
                                    ;
            100000                  BIT15== 100000
            040000                  BIT14== 40000
            020000                  BIT13== 20000
            010000                  BIT12== 10000
            004000                  BIT11== 4000
            002000                  BIT10== 2000
            001000                  BIT09== 1000
            000400                  BIT08== 400
            000200                  BIT07== 200
            000100                  BIT06== 100
            000040                  BIT05== 40
            000020                  BIT04== 20
            000010                  BIT03== 10
            000004                  BIT02== 4
            000002                  BIT01== 2
            000001                  BIT00== 1
                                    ;
            001000                  BIT9==  BIT09
            000400                  BIT8==  BIT08
            000200                  BIT7==  BIT07
            000100                  BIT6==  BIT06
            000040                  BIT5==  BIT05
            000020                  BIT4==  BIT04
            000010                  BIT3==  BIT03
            000004                  BIT2==  BIT02
            000002                  BIT1==  BIT01
            000001                  BIT0==  BIT00
                                    ;
                                    ; EVENT FLAG DEFINITIONS
                                    ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                                    ;
            000040                  EF.START==      32.                     ; START COMMAND WAS ISSUED
            000037                  EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
            000036                  EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
            000035                  EF.NEW==        29.                     ; A NEW PASS HAS BEEN STARTED
            000034                  EF.PWR==        28.                     ; A POWER-FAIL/POWER-UP OCCURRED
                                    ;
                                    ;
                                    ; PRIORITY LEVEL DEFINITIONS
                                    ;
```

GLOBAL EQUATES SECTION

```
000340                  PRI07== 340
000300                  PRI06== 300
000240                  PRI05== 240
000200                  PRI04== 200
000140                  PRI03== 140
000100                  PRI02== 100
000040                  PRI01== 40
000000                  PRI00== 0
                        ;
                        ;OPERATOR FLAG BITS
                        ;
000004                  EVL==     4
000010                  LOT==    10
000020                  ADR==    20
000040                  IDU==    40
000100                  ISR==   100
000200                  UAM==   200
000400                  BOE==   400
001000                  PNT==  1000
002000                  PRI==  2000
004000                  IXE==  4000
010000                  IBE== 10000
020000                  IER== 20000
040000                  LOE== 40000
100000                  HOE== 100000
837
838 002144                      KT11    ,,      ;DEFINE MEMORY MANAGEMENT REGISTERS
                        .SBTTL  MEMORY MANAGEMENT DEFINITIONS
                        ;*KT11 VECTOR ADDRESS
000250                  MMVEC=  250
                        ;*KT11 STATUS REGISTER ADDRESSES
177572                  SR0=    177572
177574                  SR1=    177574
177576                  SR2=    177576
172516                  SR3=    172516
                        .IF NB
                        ;*USER "I" PAGE DESCRIPTOR REGISTERS
                        UIPDR0= 177600
                        UIPDR1= 177602
                        UIPDR2= 177604
                        UIPDR3= 177606
                        UIPDR4= 177610
                        UIPDR5= 177612
                        UIPDR6= 177614
                        UIPDR7= 177616
                        .IF NB
                        ;*USER "D" PAGE DESCRIPTOR REGISTORS
                        UDPDR0= 177620
                        UDPDR1= 177622
                        UDPDR2= 177624
                        UDPDR3= 177626
                        UDPDR4= 177630
                        UDPDR5= 177632
                        UDPDR6= 177634
                        UDPDR7= 177636
                        .ENDC
                        ;*USER "I" PAGE ADDRESS REGISTERS
```

```
UIPAR0= 177640
UIPAR1= 177642
UIPAR2= 177644
UIPAR3= 177646
UIPAR4= 177650
UIPAR5= 177652
UIPAR6= 177654
UIPAR7= 177656
.IF NB
;*USER "D" PAGE ADDRESS REGISTERS
UDPAR0= 177660
UDPAR1= 177662
UDPAR2= 177664
UDPAR3= 177666
UDPAR4= 177670
UDPAR5= 177672
UDPAR6= 177674
UDPAR7= 177676
.ENDC
.ENDC
.IF NB
;*SUPERVISOR "I" PAGE DESCRIPTOR REGISTERS
SIPDR0= 172200
SIPDR1= 172202
SIPDR2= 172204
SIPDR3= 172206
SIPDR4= 172210
SIPDR5= 172212
SIPDR6= 172214
SIPDR7= 172216
.IF NB
;*SUPERVISOR "D" PAGE DESCRIPTOR REGISTERS
SDPDR0= 172220
SDPDR1= 172222
SDPDR2= 172224
SDPDR3= 172226
SDPDR4= 172230
SDPDR5= 172232
SDPDR6= 172234
SDPDR7= 172236
.ENDC
;*SUPERVISOR "I" PAGE ADDRESS REGISTERS
SIPAR0= 172240
SIPAR1= 172242
SIPAR2= 172244
SIPAR3= 172246
SIPAR4= 172250
SIPAR5= 172252
SIPAR6= 172254
SIPAR7= 172256
.IF NB
;*SUPERVISOR "D" PAGE ADDRESS REGISTERS
SDPAR0= 172260
SDPAR1= 172262
SDPAR2= 172264
SDPAR3= 172266
SDPAR4= 172270
```

```
                        SDPAR5= 172272
                        SDPAR6= 172274
                        SDPAR7= 172276
                        .ENDC
                        .ENDC
                        ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
172300                  KIPDR0= 172300
172302                  KIPDR1= 172302
172304                  KIPDR2= 172304
172306                  KIPDR3= 172306
172310                  KIPDR4= 172310
172312                  KIPDR5= 172312
172314                  KIPDR6= 172314
172316                  KIPDR7= 172316
                        .IF NB
                        ;*KERNEL "D" PAGE DESCRIPTOR REGISTERS
                        KDPDR0= 172320
                        KDPDR1= 172322
                        KDPDR2= 172324
                        KDPDR3= 172326
                        KDPDR4= 172330
                        KDPDR5= 172332
                        KDPDR6= 172334
                        KDPDR7= 172336
                        .ENDC
                        ;*KERNEL "I" PAGE ADDRESS REGISTERS
172340                  KIPAR0= 172340
172342                  KIPAR1= 172342
172344                  KIPAR2= 172344
172346                  KIPAR3= 172346
172350                  KIPAR4= 172350
172352                  KIPAR5= 172352
172354                  KIPAR6= 172354
172356                  KIPAR7= 172356
                        .IF NB
                        ;*KERNEL "D" PAGE ADDRESS REGISTERS
                        KDPAR0= 172360
                        KDPAR1= 172362
                        KDPAR2= 172364
                        KDPAR3= 172366
                        KDPAR4= 172370
                        KDPAR5= 172372
                        KDPAR6= 172374
                        KDPAR7= 172376
                        .ENDC
```

```
                                .SBTT   TK-25 REGISTER AND PACKET DEFINITIONS


                        ;
                        ; SOME GENERAL EQUATES.
                        ;

000004                  ERRVEC==        4               ; POINTER TO ERROR VECTOR FOR BUS TIME OUT.
000060                  TTIVEC==        60              ; INTERRUPT VECTOR FOR CONSOLE INPUT
177560                  TTICSR==        177560          ; BUS ADDRESS OF CONSOLE INPUT
177562                  TTIBFR==        177562          ; CONSOLE INPUT DATA BUFFER

                        ;+
                        ;BIT DEFINITIONS FOR TSSR REGISTER
                        ;-

100000                  SC=     BIT15                   ;SPECIAL CONDITION
040000                  BIE=    BIT14                   ;BUS INTERFACE ERROR
020000                  SCE=    BIT13                   ;SANITY CHECK ERROR
010000                  RMR=    BIT12                   ;MODIFICATION REFUSED
004000                  NXM=    BIT11                   ;NONEXISTANT MEMORY ERROR
002000                  NBA=    BIT10                   ;NEED BUFFER ADDRESS
001400                  HIADDR= BIT9!BIT8               ;EXTENDED ADDRESS BITS
000200                  SSR=    BIT7                    ;SUB SYSTEM READY
000100                  OFL=    BIT6                    ;OFF LINE BIT
000060                  FATERR= BIT4!BIT5               ;FATAL TERMINATION ERROR CODES
000016                  TERCLS= BIT3!BIT2!BIT1          ;TERMINATION CODES


                        ;+
                        ;
                        ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0
                        ;(XST0)
                        ;
                        ;-

100000                  XSOTMK= BIT15                   ;TAPE MARK DETECTED
040000                  XSORLS= BIT14                   ;RECORD LENGTH SHORT
020000                  XSOLET= BIT13                   ;LOGICAL END OF TAPE
010000                  XSORLL= BIT12                   ;RECORD LENGTH LONG
004000                  XSOWLE= BIT11                   ;WRITE LOCK ERROR
002000                  XSONEF= BIT10                   ;NON EXECUTABLE FUNCTION
001000                  XSOILC= BIT9                    ;ILLEGAL COMMAND
000400                  XSOILA= BIT8                    ;ILLEGAL ADDRESS
000200                  XSOMOT= BIT7                    ;TAPE IN MOTION
000100                  XSOONL= BIT6                    ;TRANSPORT ON LINE
000040                  XSOIE=  BIT5                    ;INTERRUPT ENABLE
000020                  XSOVCK= BIT4                    ;VOLUME CHECK BIT
000010                  XSOPED= BIT3                    ;PHASE ENCODED DRIVE
000004                  XSOWLK= BIT2                    ;WRITE LOCKED
000002                  XSOBOT= BIT1                    ;BEGINNING OF TAPE
000001                  XSOEOT= BIT0                    ;END OF TAPE


                        ;+
                        ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 1
                        ;(XST1)
                        ;-
```

```
                        X1.DLT  = BIT15                  ;DATA LATE
                        X1.SPARE= BIT14                  ;NOT USED
                        X1.COR  = BIT13                  ;CORRECTABLE DATA ERROR
                        X1.MBZ  = BIT12+BIT11+BIT10+BIT9+BIT7+BIT6+BIT5+BIT4+BIT3+BIT2+BIT0 ;ALWAYS 0
                        X1.RBP  = BIT8                   ;READ BUS PARITY ERROR
                        X1.UNC  = BIT1                   ;UNCORRECTABLE DATA OR HARD ERROR

                        ;+
                        ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
                        ;(XST2)
                        ;-
                        X2.OPM  = BIT15                  ;OPERATION IN PROGRESS (TAPE MOVING)
                        X2.RCE  = BIT14                  ;RAM CHECKSUM ERROR
                        X2.SPARE= BIT13+BIT12+BIT11+BIT9+BIT8   ;NOT USED BY TK-25 (ALWAYS=0)
                        X2.WCF  = BIT10                  ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
                        X2.EXTF = BIT7                   ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
                        X2.BUFE = BIT6                   ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
                        X2.REV  = 000077                 ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
                        X2.UNIT = BIT2+BIT1+BIT0         ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.

                        ;+
                        ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
                        ;(XST3)
                        ;-
                        X3.MDE  = 177400                 ;MICRO-DIAGNOSTIC ERROR CODE
                        X3.SPARE= BIT7                   ;NOT USED BY TK-25
                        X3.OPI  = BIT6                   ;OPERATION INCOMPLETE
                        X3.REV  = BIT5                   ;REVERSE
                        X3.TRF  = BIT4                   ;TRANSPORT RESPONSE FAILURE
                        X3.DCK  = BIT3                   ;DENSITY CHECK
                        X3.MBZ  =BIT2+BIT1               ;NOT USED ALWAYS 0
                        X3.RIB  = BIT0                   ;REVERSE INTO BOT

                        ;+
                        ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4
                        ;(XST4)
                        ;-
                        X4.HSP  = BIT15                  ;HIGH SPEED
                        X4.RCE  = BIT14                  ;RETRY COUNT EXCEEDED
                        X4.TSM  = BIT13                  ;TRANSPORT SPECIAL MODE
                        X4.MBZ  = BIT12+BIT11+BIT10+BIT9+BIT8   ;NOT USED ALWAYS 0
                        X4.WRC  = 000377                 ;WRITE RETRY COUNT FIELD


                        ;+
                        ;
                        ;TSSR TERMINATION CODES (BIT 0-2)
                        ;
                        ;-

                                TSREJ=  3*2                      ;COMMAND REJECTED
                                UNREC=  6                        ;UNRECOVERABLE ERROR

                        ;+
                        ;
                        ;DEVICE REGISTER OFFSETS
                        ;
```

```
                        ;-
                        TSBA==  -2
177776                  TSBAL== -2
177776                  TSDB==  -2                      ;TSDB/TSBA REGISTER
177776                  TSDBL== -2                      ;TSDB/TSBA REGISTER
177777                  TSBAH== -1
177777                  TSDBH== -1                      ;TSDB/TSBA REGISTER HIGH BYTE
000000                  TSSR==  0                       ;TSSR REGISTER
000001                  TSSRH== 1                       ;TSSR REGISTER HIGH BYTE

                        ;+
                        ; TSDB ADDRESS BIT DEFINITIONS
                        ;-
000003                  A1716   = BIT1+BIT0             ;ADDRESS BITS 17;16 ARE IN 1:0

                        ;+
                        ; COMMAND DEFINITIONS
                        ;-
000017                  P.GETSTAT       = 17            ;GET STATUS
000013                  P.INIT          = 13            ;INITIALIZE
000012                  P.CONTROL       = 12            ;CONTROL COMMANDS
000011                  P.FORMAT        = 11            ;FORMAT
000010                  P.POSITION      = 10            ;POSITION
000006                  P.WRTSUB        = 6             ;SUBSYSTEM WRITE
000005                  P.WRITE         = 5             ;WRITE
000004                  P.WRTCHAR       = 4             ;WRITE CHARACTERISTICS
000001                  P.READ          = 1             ;READ

                        ;+
                        ; COMMAND PACKET HEADER WORD BIT DEFINITIONS
                        ;-
100000                          P.ACK   = BIT15                 ;BUFFER AVAIL FOR CONTROLLER
040000                          P.CVC   = BIT14                 ;CLEAR VOLUME CHECK
020000                          P.OPP   = BIT13                 ;REVERSE SEQUENCE OF DATA BITS
010000                          P.SWB   = BIT12                 ;SWAP BYTES IN MEMORY
007400                          P.MODE  = BIT11!BIT10!BIT9!BIT8 ;EXTENDED COMMAND MODE FIELD
000200                          P.IE    = BIT7                  ;INTERRUPT ENABLE
000140                          P.FMT= BIT6!BIT5                ;PACKET HEADER TYPE (ALWAYS=0)
000037                          P.CMD   = 37                    ;MAJOR COMMAND FIELD
                        ;+
                        ; CONTROL COMMAND MODE CODES
                        ;-
000000                          PC.RELEASE      = 0*256.        ;RELEASE BUFFER
000400                          PC.REWIND       = 1*256.        ;REWIND
001000                          PC.NOOP         = 2*256.        ;NO-OP
002000                          PC.IEREW        = 4*256.        ;REWIND IMMEDIATE INTERRUPT
002400                          PC.ERASE        = 5*256.        ;SECURITY ERASE

                        ;+
                        ; CONTROLLER RAM DEFINITIONS
                        ;-
000167                  RMCHBEG = 167                   ;CHARACTERISTICS IO DATA BEGIN RAM ADDRESS
000200                  RMCHEND = 200                   ;CHARACTERISTICS IO DATA END RAM ADDRESS
000020                  RMPKTBEG= 20                    ;COMMAND PACKET BEGIN RAM ADDRESS
000027                  RMPKTEND= 27                    ;COMMAND PACKET END RAM ADDRESS
000104                  RMMSGBEG= 104                   ;MESSAGE BUFFER BEGIN RAM ADDRESS
```

```
 1014          000117                  RMMSGEND= 117                           ;MESSAGE BUFFER END RAM ADDRESS
 1015                                  ;+
 1016                                  ;
 1017                                  ;REGISTER DEFINITIONS IN THE MESSAGE BUFFER
 1018                                  ;
 1019                                  ;-
 1020
 1021          000006                  XST0==  6                               ;EXTENDED STATUS REGISTER 0 (WORD 4)
 1022          000010                  XST1==  8.                              ;EXTENDED STATUS REGISTER 1 (WORD 5)
 1023          000012                  XST2==  10.                             ;EXTENDED STATUS REGISTER 2 (WORD 6)
 1024          000014                  XST3==  12.                             ;EXTENDED STATUS REGISTER 3 (WORD 7)
 1025          000016                  XST4==  14.                             ;EXTENDED STATUS REGISTER 4 (WORD 8)
 1026
 1027
 1028                                  ;+
 1029                                  ;
 1030                                  ;OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
 1031                                  ;
 1032                                  ;-
 1033
 1034          000002                          PKLOW   = 2                     ;LOW ORDER CHARACTERISTIC DATA POINTER
 1035          000004                          PKHI    = 4                     ;HIGH ORDER CHARACTERISTIC DATA POINTER
 1036          000006                          PKBCNT  = 6                     ;NUMBER OF BYTES IN DATA PACKET
 1037
 1038          000010                          EXBCNT=10                       ;NUMBER OF BYTES IN EXTENDED DATA PACKET
 1039
 1040                                  ;+
 1041                                  ;DATA PACKET OFFSETS FOR WRITE SUBSYTEM COMMAND
 1042                                  ;-
 1043          000000                          BSEL0   = 0                     ;BYTE 0
 1044          000001                          BSEL1   = 1                     ;BYTE 1
 1045          000002                          SEL2    = 2                     ;WORD 2
 1046          000004                          SELDATA = 4                     ;WORD 3
 1047
 1048                                  ;+
 1049                                  ;BSEL0 SELECT CODES FOR WRITE SUBSYSTEM COMMAND
 1050                                  ;-
 1051          000000                  PW.NOP          = 0                     ;NO-OP
 1052          000001                  PW.RDRAM        = 1                     ;READ RAM
 1053          000002                  PW.WTRAM        = 2                     ;WRITE RAM
 1054          000003                  PW.RFIFO        = 3                     ;READ FIFO
 1055          000004                  PW.WFIFO        = 4                     ;WRITE FIFO
 1056          000005                  PW.RDSTAT       = 5                     ;READ STATUS
 1057          000006                  PW.WCTL         = 6                     ;WRITE TAPE CONTROL
 1058          000007                  PW.WFMT         = 7                     ;WRITE TAPE FORMAT
 1059          000010                  PW.WMISC        = 10                    ;WRITE MISCELLANEOUS
 1060          000011                  PW.WNPR         = 11                    ;WRITE NPR CONTROL
 1061          000020                  PW.D22          = 20                    ;DO MICROTEST 22
 1062          000021                  PW.D11          = 21                    ;DO MICROTEST 11
 1063          000022                  PW.D13          = 22                    ;DO MICROTEST 13
 1064          000023                  PW.NO1311       = 23                    ;DISABLE MICROTEST 11 AND 13
 1065          000024                  PW.RDEXT        = 24                    ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPO
RTS
 1066
 1067                                  ;+
 1068                                  ;BSEL1 CODES FOR WRITE TAPE CONTROL
 1069                                  ;-
 1070          000200                  WC.IFAD         = BIT7                  ;IFAD - FORMATTER ADDRESS
```

```
1071    000100          WC.IOTAD        = BIT6          ;ITAD0  - TRANSPORT ADDRESS BIT 0
1072    000040          WC.I1TAD        = BIT5          ;ITAD1  - TRANSPORT ADDRESS BIT 1
1073    000020          WC.I5RESV       = BIT4          ;IRESV5 - RESERVED #5
1074    000010          WC.IREW         = BIT3          ;IREW   - REWIND
1075    000004          WC.IRWU         = BIT2          ;IRWU   - REWIND AND UNLOAD
1076    000002          WC.IFEN         = BIT1          ;IFEN   - FORMATTER ENABLE
1077    000001          WC.IGO          = BIT0          ;GO
1078
1079                    ;+
1080                    ;BSEL1 CODES FOR WRITE FORMAT
1081                    ;-
1082    000200          WF.IHISP        = BIT7          ;IHISP  - HIGH SPEED
1083    000100          WF.IWRT         = BIT6          ;IWRT   - WRITE
1084    000040          WF.IREV         = BIT5          ;IREV   - REVERSE
1085    000020          WF.IWFM         = BIT4          ;IWFM   - WRITE FILE MARK
1086    000010          WF.IEDIT        = BIT3          ;IEDIT  - EDIT
1087    000004          WF.IERASE       = BIT2          ;IERASE - ERASE
1088    000002          WF.I3RESV       = BIT1          ;IRESV3 - RESERVED #3
1089    000001          WF.I4RESV       = BIT0          ;IRESV4 - RESERVED #4
1090
1091
1092                    ;+
1093                    ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
1094                    ;-
1095    000200          MS.EXT          = BIT7          ;INVERT SENSE OF EXTENDED FEATURES SWITCH
1096    000020          MS.RSFIFO       = BIT4          ;RESET FIFO AND INPUT PARITY ERRORR
1097    000010          MS.RSTAPE       = BIT3          ;RESET TAPE STATUS IN 2 FLIP-FLOPS
1098    000006          MS.ATTN         = BIT2!BIT1     ;ATTENTION TRIGGER FIELD
1099    000001          MS.RSD          = BIT0          ;RESET TIMER A,B THEN DELAY TIMES IN SEL2
1100                    ;+
1101                    ; MS.ATTN SUBCODES
1102                    ;-
1103    000000                  MSA.NOP = 0*2           ;NO-OP (NOTHING TRIGGERED)
1104    000002                  MSA.VOL = 1*2           ;SIMULATE ON-LINE/OFF-LINE TRANSISTION
1105    000004                  MSA.NRAM= 2*2           ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
1106    000006                  MSA.FRAM= 3*2           ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)
1107                    ;+
1108                    ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
1109                    ;-
1110    000200          NP.IR           = BIT7          ;INTERRUPT REQUEST (0-1 TRANSITION)
1111    000100          NP.OUT          = BIT6          ;TAPE DATA DIRECTION OUT (0= IN)
1112    000040          NP.LOOP         = BIT5          ;ENABLE TRANSPORT LOOPBACK
1113    000020          NP.WRP          = BIT4          ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
1114                    ;+
1115                    ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
1116                    ;-
1117
1118    000200          S2.DIM          = BIT7          ;WORD #9 BYTE 2 CA.B 1.. MISS
1119    000100          S2.ILW          = BIT6          ;               ILW H
1120    000040          S2.OUTRDY       = BIT5          ;               OUT RDY H
1121    000020          S2.INRDY        = BIT4          ;               IN RDY H
1122    000010          S2.ATIMR        = BIT3          ;               TIMER A FLAG H
1123    000004          S2.BTIMR        = BIT2          ;               TIMER B FLAG H
1124    000003          S2.UNDEF        = BIT1+BIT0     ;(UNDEFINED)
1125    100000          S1.PARIN        = BIT15         ;WORD #8 BYTE 1 PARIN H
1126    040000          S1.I2RESV       = BIT14         ;               IRESV2
1127    020000          S1.I1RESV       = BIT13         ;               IRESV1
```

```
1128    010000          S1.IEOT     = BIT12         ;                   IEOT L
1129    004000          S1.IIDENT   = BIT11         ;                   IIDENT H
1130    002000          S1.ICER     = BIT10         ;                   ICER H
1131    001000          S1.IFMK     = BIT9          ;                   IFMK H
1132    000400          S1.IHER     = BIT8          ;                   IHER H
1133    000200          S0.ISPEED   = BIT7          ;WORD #8 BYTE 0     ISPEED H
1134    000100          S0.IRDY     = BIT6          ;                   IRDY L
1135    000040          S0.IONL     = BIT5          ;                   IONL L
1136    000020          S0.ILDP     = BIT4          ;                   ILDP L
1137    000010          S0.IDBY     = BIT3          ;                   IDBY L
1138    000004          S0.IRWD     = BIT2          ;                   IRWD L
1139    000002          S0.IFBY     = BIT1          ;                   IFBY L
1140    000001          S0.IFPT     = BIT0          ;                   IFPT L
1141                    ;       SPECIAL KEYBORD STUFF FOR MOVER PROGRAM
1142
1143    177560              TKS     =177560         ;KEYBOARD STATUS REGISTER
1144    177562              TKB     =177562         ;KEYBOARD DATA REGISTER
1145    177564              TPS     =177564         ;CONSOLE PRINTER STATUS REGISTER
1146    177566              TPB     =177566         ;CONSOLE PRINTER DATA REGISTER
1147    007776              HIMEM   =007776         ;HIGH MEMORY MASK VALUE
1148                    ;       CONTROLLER DEFINITIONS
1149                    ;
1150                    ;
1151    174400              CSR     =174400         ;STATUS AND CONTROL REGISTER
1152    174402              BAR     =174402         ;DL ADDRESS REGISTER
1153    174404              DAR     =174404         ;PLATTER ADDRESS
1154    174406              MPR     =174406         ;MULTIPURPOSE REGISTER
1155                    ;
1156                    ;
1157                    ;
1158                    ;
1159                    ;
1160                    ;       CONTROLLER COMMANDS
1161                    ;
1162                    ;
1163    000004              DLGETS  =4              ;GET STATUS COMMAND
1164    000006              SEEK    =6              ;SEEK TRACK AND HEAD SELECT
1165    000010              DLRDHD  =10             ;READ SECTOR HEADER
1166    000014              READ    =14             ;READ COMMAND
1167    000016              DLRDNH  =16             ;READ SECTOR NO HEADER CHECK
1168                    ;
1169                    ;
1170                    ;
1171                    ;
1172                    ;
1173                    ;
1174    000001              READY   =1              ;DRIVE READY BIT IN STATUS REG.
1175    000013              DLSR    =13             ;STATUS AND RESET
1176    177730              DLERR   =177730         ;MASK FOR COVER OPEN
1177    000006              DLUN    =6              ;HEADS UNLOADED
1178    000177              DLCYL   =000177         ;MASK FOR CYLINDER ADDRESS
1179    100200              DLDNER  =100200         ;DONE SET OR ERROR SET BITS
1180                    ;
1181                    ;
1182                    ;
1183                    ;
1184    177560              TTICSR  =       177560  ;KEYBOARD INPUT STATUS
```

```
1185        177562                  TTIBFR  =       177562          ;KEYBOARD DATA REGISTER
1186        177564                  TTOCSR  =       177564          ;CONSOLE PRINTER STATUS REGISTER
1187        177566                  TTOBFR  =       177566          ;CONSOLE PRINTER DATA REGISTER
1188
```

```
1190                                    .SBTTL  SPECIAL MACROS AND OPDEFS.
1191
1192
1193                            ;+
1194                            ;SAVE GENERAL REGS 1 TO 5
1195                            ;-
1196
1197                                    .MACRO  SAVREG
1198                                    JSR     R5,REGSAV
1199                                    .ENDM
1200
1201                            ;+
1202                            ; MACRO TO FORCE AN ERROR
1203                            ;-
1204                            .MACRO  FORCERROR       TAG,NOTSSR
1205                            .NLIST
1206                            .IIF NDF LISTALL, .NLIST
1207                            .LIST
1208                            .IF B NOTSSR
1209                                    MOV     TSSR(R5),R1             ;READ TSSR
1210                            .ENDC
1211                                    MOV     FORCER,FORCER           ;IS FORCER SET? (LEAVE C BIT ALONE)
1212                                    BNE     TAG                     ;BR IF YES
1213                            .NLIST
1214                            .IIF NDF LISTALL, .LIST
1215                            .LIST
1216                            .ENDM
1217
1218                            ;+
1219                            ; MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
1220                            ;       WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
1221                            ;       SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
1222                            ;       FORCER TO 177777
1223                            ;       TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
1224                            ;-
1225                            .MACRO  FORCEXIT        TAG
1226                            .NLIST
1227                            .IIF NDF LISTALL, .NLIST
1228                            .LIST
1229                                    MOV     FORCER,FORCER           ;IS FORCER NEGATIVE?
1230                                    BMI     TAG                     ;BR IF YES
1231                            .NLIST
1232                            .IIF NDF LISTALL, .LIST
1233                            .LIST
1234                            .ENDM
1235                            ;+
1236                            ; MACRO TO INCREMENT ERROR COUNTS
1237                            ;-
1238                            .MACRO  NEXT.ERRNO
1239                            .NLIST
1240                            ;;;.IIF NDF LISTALL, .NLIST
1241                                    ERRNO=ERRNO+1
1242                            ;;;.IIF NDF LISTALL, .LIST
1243                            .LIST
1244                            .ENDM
1245
1246                            ;+
```

```
                                ;MACRO TO PERFORM XOR
                                ;
                                        .MACRO  XOR     A,B
                                        MOV     A,-(SP)
                                        BIC     B,(SP)
                                        BIC     A,B
                                        BIS     (SP)+,B
                                        .ENDM

        000000                          EN=0                            ; INITIALIZE ERROR NUMBER
                                        .SBTTL  FORCER  - FORCE ERROR FLAG

                                ;
                                ; THE FOLLOWING LOCATIONS MAY BE PATCHED BY THE USER
                                ; TO OBTAIN THE RESULTS DESCRIBED FOR EACH.
                                ;

002144  000000                  FORCER::        0       ; FORCE TYPE ALL HARD ERRORS (THE ONES CALLED
                                                ; - BY THE MACRO "IFERROR").  AN ERROR NEED NOT -
                                                ; - EXIST, JUST ASSUME AND TYPE THE MESSAGE.
```

```
                                .SBTTL  GLOBAL DATA SECTION

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

                        ;
                        ;THE FOLLOWING DATA ARE SET FOR EACH UNIT AT INIT TIME.
                        ;SINGLE UNIT DEFAULTS (LISTED) ARE IN THE DEFAULT P-TABLE.
                        ;
002146  000000          EPRTSW::        .WORD   0               ;PRINT SWITCH
002150  000000          UNITN::         .WORD   0               ;UNIT # UNDER TEST.
002152  000000          QVP::           .WORD   0               ;QUICK VERIFY FLAG.
002154  000000          CSRADDR::       .WORD   0               ;ADDRESS OF CSR FOR CURRENT DEVICE
002156  000224          IVEC::          .WORD   224             ;INTERRUPT VECTOR
002160  000200          IPRI::          .WORD   PRI04           ;INTERRUPT PRIORITY.
002162  000000          TSTCNT::        .WORD   0               ;NUMBER OF TESTS RUN IN THIS PASS
002164  000000          LOOPCNT::       .WORD   0               ;REMAINING ITERATION COUNT FOR TEST
002166  000000          DEVCNT::        .WORD   0               ;NUMBER OF DEVICE UNDER TEST
002170  000000          FATFLG::        .WORD   0               ;SET IF FATAL ERROR IS DETECTED IN TEST
002172  000000          INTRECV::       .WORD   0               ;SET IF TAPE INTERRUPT WAS RECEIVED
002174  000000          BENBSW::        .WORD   0               ;BUFFER ENABLE SWITCH SW 0=OFF;1=ON
002176  000000          EXPD::          .WORD   0               ;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
002200  000000          RECV::          .WORD   0               ;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
002202  000000          ERRHI::         .WORD   0               ;HIGH ADDRESS MEMORY ERROR
002204  000000          ERRLO::         .WORD   0               ;LOW ADDRESS MEMORY ERROR
002206                  RAMDATA::       .BLKW   16.             ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
002246  000000          RAMSIZ::        .WORD   0               ;RAM DATA SIZE FOR PRAMPKT ROUTINE
002250  000000          RCVHIADD::      .WORD   0               ;RECEIVED BUFFER HIGH ADDRESS
002252  000000          RCVLOADD::      .WORD   0               ;RECEIVED BUFFER LOW ADDRESS
002254  000000          COUNT::         .WORD   0               ;TEST COUNT PATTERN
002256  000000          DATA::          .WORD   0               ;TEST DATA
002260  000000          TSTFLAG::       .WORD   0               ;TEST FLAG WORD
002262  000000          TSTPTR::        .WORD   0               ;TSTBLK POINTER
002264  000000          PRMNO::         .WORD   0               ;PRINT ROUTINE TEMP
002266                  EXPMSG::        .BLKB   100.            ;EXPECTED MESSAGE BUFFER DATA
002432                  RECMSG::        .BLKB   100.            ;RECEIVED MESSAGE BUFFER DATA
002576                  TMPBFR::        .BLKB   80.             ;TEMPORARY STORAGE FOR PRINT
002716  000000          MESBFA::        .WORD   0               ;STORES ADDRESS OF MESSAGE BUFFER FOR ERR PRT
```

```
                                        .SBTTL  TSTBLK  - TEST DATA TABLE

                                ;+
                                ;
                                ;THIS TABLE CONTAINS TEST DATA USED IN SEVERAL TESTS
                                ;
                                ;IN SEQUENCE THE DATA IS:
                                ;
                                ;       ALL ZEROS
                                ;       ALL ONES
                                ;       WALKING ONES
                                ;       WALKING ZEROS
                                ;       ALTERNATING ONES AND ZEROS
                                ;
                                ;-

002720                          TSTBLK::
002720  000000                          .WORD   0                       ;ALL ZEROS
002722  177777                          .WORD   177777                  ;ALL ONES
002724  000001                          .WORD   BIT0                    ;DATA FOR WALKING ONES
002726  000002                          .WORD   BIT1
002730  000004                          .WORD   BIT2
002732  000010                          .WORD   BIT3
002734  000020                          .WORD   BIT4
002736  000040                          .WORD   BIT5
002740  000100                          .WORD   BIT6
002742  000200                          .WORD   BIT7
002744  000400                          .WORD   BIT8
002746  001000                          .WORD   BIT9
002750  002000                          .WORD   BIT10
002752  004000                          .WORD   BIT11
002754  010000                          .WORD   BIT12
002756  020000                          .WORD   BIT13
002760  040000                          .WORD   BIT14
002762  100000                          .WORD   BIT15
002764  177776                          .WORD   ^CBIT0                  ;DATA FOR WALKING ZEROS
002766  177775                          .WORD   ^CBIT1
002770  177773                          .WORD   ^CBIT2
002772  177767                          .WORD   ^CBIT3
002774  177757                          .WORD   ^CBIT4
002776  177737                          .WORD   ^CBIT5
003000  177677                          .WORD   ^CBIT6
003002  177577                          .WORD   ^CBIT7
003004  177377                          .WORD   ^CBIT8
003006  176777                          .WORD   ^CBIT9
003010  175777                          .WORD   ^CBIT10
003012  173777                          .WORD   ^CBIT11
003014  167777                          .WORD   ^CBIT12
003016  157777                          .WORD   ^CBIT13
003020  137777                          .WORD   ^CBIT14
003022  077777                          .WORD   ^CBIT15
003024  125252                          .WORD   125252                  ;ALTERNATING ONES, ZEROS
003026  052525                          .WORD   052525                  ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE
003030                          TBLEND==.
```

```
                                        .SBTTL  GLOBAL ENVIRONMENT STORAGE
                                ;
                                ;STORAGE FOR DEVICE REGISTERS
                                ;
003030  000000  100000  000000  DUMMY:  0,100000,0,0                    ;DUMMY DEVICE REGISTERS...
003040  000000  000000  000000          0,0,0,0,0,0,0,0
                                                                        ;...FOR MULTI-UNIT CHECKOUT.


003060  000000                  DUFLG::         .WORD   0               ;"DROPPED UNIT' FLAG.
                                                                        ;INHIBITS CODE IN "CLEAN-UP".
003062  000000                  NODEV::         .WORD   0               ;FLAG TO SAY NO DEVICE.

003064  000000                  TEMP1::         .WORD   0               ;SOME TEMP LOCATIONS.
003066  000000                  TEMP2::         .WORD   0
003070  000000                  XXCOMM::        .WORD   0               ;XXDP+ COMM BLOCK POINTER.
003072  000000                  FREE::          .WORD   0               ;1ST FREE MEMORY ADDRESS...
003074  000000                  FRESIZ::        .WORD   0               ;...AND SIZE (IN WORDS).
003076  000000                  FREEHI:         .WORD   0               ;LAST WORD IN FREE SPACE
003100  000000                  KTFLG::         .WORD   0               ;KT11, MEM AVAIL FLAG -
                                                                        ;- .WORD        0 = <24K OR NO KT -
                                                                        ;- NZ = >24K AND KT.
003102  000000                  KTENABLE::      .WORD   0               ;SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
003104  002000                  PST32W::        .WORD   2000            ;32W BLOCK ADDRESS FOR 32K START
003106  000000                  SIFLAG::        .WORD   0               ;
003110  000000                  BADDAT::        .WORD   0               ;ACTUAL DATA
003112  000000                  GDDAT::         .WORD   0               ;EXPECTED DATA
003114  000000                  LOOPFL::        .WORD   0
003116                          CTAB::                                  ;CONFIGURATION TABLES.
003116  000000                  CTABM::         .WORD   0               ;CONFIG WORK.
003120  000000                                  .WORD   0
003122  000000                                  .WORD   0
003124  000000                                  .WORD   0
003126  177777                                  .WORD   -1                      ;END OF MEM TABLE.
003130                          CTABE::
                                ;ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
                                ;
                                ;       0       =       UNIT NOT TESTED
                                ;       100000  =       UNIT ONLINE, NO ERRORS
                                ;       10XXXX  =       UNIT ONLINE, ENCOUNTERED XXXX ERRORS
                                ;       160000  =       UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
                                ;       160001  =       UNIT DROPPED, NOT IDLE AT START
                                ;       14XXXX  =       UNIT DROPPED, ENCOUNTERED XXXX ERRORS
                                ;
003130                          ERTABL:         .BLKW   64.
003330  000000                  ERTABE:         .WORD   0

003332  000000  SKIPT:  .WORD   0                               ;1=SKIP SUBTEST 0=NO SKIP OF SUBTEST
```

```
                                        .SBTTL  GLOBAL TEXT MESSAGES
                                ;++
                                ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
                                ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
                                ; MORE THAN ONE TEST.
                                ;--



                                ;+
                                ;NAMES OF DEVICES SUPPORTED
                                ;-

003334                                  DEVTYP  <TK-25>
003334                          L$DVTYP::
003334  124  113  055                    .ASCIZ  /TK-25/
                                        .EVEN


                                ;+
                                ;TEST DESCRIPTION
                                ;-
003342                                  DESCRIPT <CZTKEB TK-25 FRT END FUNC #1>
003342                          L$DESC::
003342  103  132  124                    .ASCIZ  /CZTKEB TK-25 FRT END FUNC #1/
                                        .EVEN

                                ;+
                                ;BIT TO ASCII CONVERSION FOR TSSR REGISTER
                                ;-

003400  003440  003443  003447  TSSRBIT::       .WORD   1$,2$,3$,4$,5$,6$,7$,8$
003420  003501  003505  003511                  .WORD   9$,10$,11$,12$,13$,14$,15$,16$
003440    123    103    000    1$:     .ASCIZ  'SC'
003443    102    111    105    2$:     .ASCIZ  'BIE'
003447    123    103    105    3$:     .ASCIZ  'SCE'
003453    122    115    122    4$:     .ASCIZ  'RMR'
003457    116    130    115    5$:     .ASCIZ  'NXM'
003463    116    102    101    6$:     .ASCIZ  'NBA'
003467    102    111    124    7$:     .ASCIZ  'BIT9'
003474    102    111    124    8$:     .ASCIZ  'BIT8'
003501    123    123    122    9$:     .ASCIZ  'SSR'
003505    117    106    114    10$:    .ASCIZ  'OFL'
003511    102    111    124    11$:    .ASCIZ  'BIT5'
003516    102    111    124    12$:    .ASCIZ  'BIT4'
003523    102    111    124    13$:    .ASCIZ  'BIT3'
003530    102    111    124    14$:    .ASCIZ  'BIT2'
003535    102    111    124    15$:    .ASCIZ  'BIT1'
003542    102    111    124    16$:    .ASCIZ  'BIT0'
                                        .EVEN
003550    124    123    123    SFIERR: .ASCIZ  'TSSR ERROR AFTER SOFT INIT'
003603    124    123    123    SFHERR: .ASCIZ  'TSSR ERROR AFTER BUS RESET'
003636    040    040    116    NXR:    .ASCIZ  /  NON-EXISTANT DEVICE REGISTER/
003675    045    101    040    NXRX:   .ASCIZ  /%A  ADDRESS: %O6/
003716    045    101    040    TSSX:   .ASCII  /%A  TSBA,TSSR EXP'D: %O6%A,%O6%N/
003756    045    101    040            .ASCIZ  /%A  TSBA,TSSR REC'D: %O6%A,%O6/
004015    045    116    045    FUSI:   .ASCII  /%N%A/
```

```
 1468 004021  040  040  125  USI:     .ASCIZ  /  UNEXPECTED INTERRUPT/
 1469 004050  040  040  111  NSI:     .ASCIZ  /  INTERRUPT EXPECTED, NOT RECEIVED/
 1470 004113  045  116  045  FNOINTR:         .ASCII  /%N%A/
 1471 004117  040  040  116  NOINTR: .ASCIZ  /  NO INTERRUPT WAS GENERATED/
 1472 004154  040  040  111  IFAULT: .ASCIZ  /  INTERRUPT FAULT/
 1473 004176  045  101  040  INTX:   .ASCIZ  /%A  CPU PC: %O6%A  TSBA: %O6/
 1474 004233  040  040  042  NOINIT: .ASCIZ  /  "BUS-INIT" DIDN'T INITIALIZE CONTROLLER/
 1475 004305  040  040  042  NSINIT: .ASCIZ  /  "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
 1476 004355  040  040  042  BRINIT: .ASCIZ  /  "BUS-RESET" DIDN'T INITIALIZE THE DPU/
 1477
 1478 004425  000            NUL:    .ASCIZ  //
 1479 004426  045  116  000  NULCR:  .ASCIZ  /%N/
 1480 004431  045  101  040  EXPGOT: .ASCIZ  /%A EXP'D: %O6%A, REC'D: %O6/
 1481 004465  045  116  045  EXPGT2: .ASCIZ  /%N%A EXP'D: %O6%A, %O6%N%A REC'D: %O%A, %O6/
 1482 004541  045  101  040  DUAD12: .ASCIZ  /%A REG(W) WRITTEN TO: %O6%A REG(R) READ; EXP'D: %O6%A, REC'D: %O6/
 1483 004643  122  101  115  PKTRAM::        .ASCIZ  'RAM Contents Do Not Match Packet Sent'
 1484 004711  040  040  103  SCME:   .ASCIZ  /  CONFIG DOESN'T MATCH MFG. MASTER/
 1485 004754  127  122  111  WRTMSG: .ASCIZ  'WRITE CHARACTERISTICS Failed'
 1486 005011  124  123  123  WRTERR: .ASCIZ  'TSSR Incorrect After WRITE Command, More Bits Set Than SSR'
 1487 005104  124  123  123  RDERR:  .ASCIZ  'TSSR Incorrect After READ Command, More Bits Set Than SSR'
 1488                         .EVEN
 1489
 1490
 1491
```

```
                                        .SBTTL  GLOBAL ERROR REPORT SECTION

                                ;++
                                ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX
                                ; CALLS THAT ARE USED IN MORE THAN ONE TEST.
                                ; ASCII TEXT STRINGS ARE FOUND IN THE GLOBAL TEXT SECTION.
                                ;--

                                        BGNMSG  NXRERR                  ;NON-EXISTANT DEVICE REGISTER.
005176                          NXRERR::
005176                                  PRINTX  #NXRX,NODEV             ;NODEV = NEXM ADDRESS.
005176  013746  003062                   MOV     NODEV,-(SP)
005202  012746  003675                   MOV     #NXRX,-(SP)
005206  012746  000002                   MOV     #2,-(SP)
005212  010600                           MOV     SP,R0
005214  104415                           TRAP    C$PNTX
005216  062706  000006                   ADD     #6,SP
005222  004737  005230                   JSR     PC,EXTEND               ; PRINT EXTENSION IF REQUIRED.
005226                                  ENDMSG
005226                          L10002:
005226  104423                          TRAP    C$MSG


                                ;
                                ; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
                                ; TO ANY OF THE ABOVE ERROR SIGNATURES.
                                ;
005230  005727                  EXTEND: TST     (PC)+
005232  000000                  EXTA:   0                               ; 0 = NO EXTENSION.
005234  001402                          BEQ     1$
005236  004777  177770                  JSR     PC,@EXTA                ; APPEND EXTENSION TEXT.
005242                          1$:     PRINTX  #NULCR                  ; PRINT A BLANK LINE
005242  012746  004426                   MOV     #NULCR,-(SP)
005246  012746  000001                   MOV     #1,-(SP)
005252  010600                           MOV     SP,R0
005254  104415                           TRAP    C$PNTX
005256  062706  000004                   ADD     #4,SP
005262  000207                          RTS     PC
```

```
 1519                                   .SBTTL  PRITSSR - PRINT TSSR CONTENTS
 1520
 1521                                ;+
 1522                                ;
 1523                                ;ROUTINE TO DISPLAY THE CONTENTS, AND BIT DEFINITIONS, OF
 1524                                ;THE TSSR REGISTER. THIS ROUTINE IS NORMALLY CALLED ONLY
 1525                                ;BY A MESSAGE PRINTING ROUTINE
 1526                                ;
 1527                                ;INPUTS:
 1528                                ;
 1529                                ;       R1      CONTENTS OF TSSR
 1530                                ;
 1531                                ;SUBORDINATE ROUTINES:
 1532                                ;
 1533                                ;       CHKAMB  CHECK FOR AMBIGUOUS CONTENTS
 1534                                ;
 1535                                ;-
 1536
 1537 005264                        PRITSSR:
 1538 005264                                SAVREG                          ;SAVE GENERAL REGISTERS
 1539 005270  010104                        MOV     R1,R4                   ;SAVE THE TSSR CONTENTS
 1540 005272                                PRINTB  #TSSRFOR,R4             ;PRINT THE CONTENTS OF TSSR
      005272  010446                        MOV     R4,-(SP)
      005274  012746  005736                MOV     #TSSRFOR,-(SP)
      005300  012746  000002                MOV     #2,-(SP)
      005304  010600                        MOV     SP,R0
      005306  104414                        TRAP    C$PNTB
      005310  062706  000006                ADD     #6,SP
 1541 005314  010400                        MOV     R4,R0                   ;GET TSSR BACK FOR CHKAMB
 1542 005316  004737  016540                JSR     PC,CHKAMB               ;ARE CONTENTS AMBIGUOUS ?
 1543 005322  103410                        BCS     5$                      ;BRANCH IF NOT
 1544 005324                                PRINTX  #AMBTSSR                ;SHOW CONTENTS ARE AMBIGUOUS
      005324  012746  006156                MOV     #AMBTSSR,-(SP)
      005330  012746  000001                MOV     #1,-(SP)
      005334  010600                        MOV     SP,R0
      005336  104415                        TRAP    C$PNTX
      005340  062706  000004                ADD     #4,SP
 1545 005344  010403                5$:     MOV     R4,R3                   ;CONTENTS OF TSSR
 1546 005346  042703  001476                BIC     #HIADDR!FATERR!TERCLS,R3        ;CLEAR ALL MULTIPLE BIT FIELDS
 1547 005352  001434                        BEQ     20$                     ;NO BITS ARE SET
 1548 005354  012702  002576                MOV     #TMPBFR,R2              ;TEMPORARY ASCII BUFFER
 1549 005360  012701  003400                MOV     #TSSRBIT,R1             ;ASCII EQUIVALENT OF BITS
 1550 005364  005703                10$:    TST     R3                      ;REMAINING BITS TO CONVERT
 1551 005366  001413                        BEQ     15$                     ;BRANCH WHEN ALL ARE DONE
 1552 005370  000241                        CLC                             ;CLEAR CARRY FOR SHIFT
 1553 005372  006103                        ROL     R3                      ;SHIFT NEXT BIT TO CARRY
 1554 005374  103006                        BCC     13$                     ;BRANCH IF BIT NOT SET
 1555 005376  011100                        MOV     (R1),R0                 ;POINTER TO BIT DEFINITION
 1556 005400  112022                11$:    MOVB    (R0)+,(R2)+             ;MOVE ASCIZ TO BUFFER
 1557 005402  001376                        BNE     11$                     ;MOVE ALL BITS
 1558 005404  112762  000054  177777        MOVB    #',,-1(R2)              ;INSERT A COMMA TO TERMINATE
 1559 005412  005721                13$:    TST     (R1)+                   ;POINT TO NEXT DESCRIPTION
 1560 005414  000763                        BR      10$                     ;GET THE REMAINING BITS
 1561 005416  105042                15$:    CLRB    -(R2)                   ;TERMINATE THE LINE
 1562 005420                                PRINTX  #TSSDEF,#TMPBFR         ;PRINT THE BIT DEFINITIONS
      005420  012746  002576                MOV     #TMPBFR,-(SP)
      005424  012746  006127                MOV     #TSSDEF,-(SP)
```

```
                005430  012746  000002                  MOV     #2,-(SP)
                005434  010600                          MOV     SP,R0
                005436  104415                          TRAP    C$PNTX
                005440  062706  000006                  ADD     #6,SP
          1563
          1564  005444  010403                  20$:    MOV     R4,R3           ;GET THE TSSR CONTENTS
          1565  005446  042703  177761                  BIC     #^TCTERCLS,R3   ;CLEAR ALL BUT TERMINATION
          1566  005452  016303  006220                  MOV     TCOCOD(R3),R3   ;GET THE TERMINATION CODE MEANING
          1567  005456                                  PRINTX  #TCOASC,R3      ;PRINT THE TERMINATION CODE
                005456  010346                          MOV     R3,-(SP)
                005460  012746  006017                  MOV     #TCOASC,-(SP)
                005464  012746  000002                  MOV     #2,-(SP)
                005470  010600                          MOV     SP,R0
                005472  104415                          TRAP    C$PNTX
                005474  062706  000006                  ADD     #6,SP
          1568  005500  010403                          MOV     R4,R3           ;TSSR CONTENTS AGAIN
          1569  005502  042703  177717                  BIC     #^TCFATERR,R3   ;CLEAR ALL BUT FATAL TERMINATION
          1570  005506  001421                          BEQ     25$             ;DON'T PRINT IF ZERO
          1571  005510  006203                          ASR     R3
          1572  005512  006203                          ASR     R3
          1573  005514  006203                          ASR     R3              ;ALINE TERMINATION CODE FOR INDEX
          1574  005516  016303  006560                  MOV     TSFCOD(R3),R3   ;GET THE FATAL TERMINATION CODE
          1575  005522                                  PRINTX  #TFCASC,R3      ;PRINT THE FATAL TERMINATION CODE
                005522  010346                          MOV     R3,-(SP)
                005524  012746  006060                  MOV     #TFCASC,-(SP)
                005530  012746  000002                  MOV     #2,-(SP)
                005534  010600                          MOV     SP,R0
                005536  104415                          TRAP    C$PNTX
                005540  062706  000006                  ADD     #6,SP
          1576  005544  012737  000031  002170          MOV     #25.,FATFLG     ;DROP UNIT AFTER THIS ERROR
          1577  005552  010403                  25$:    MOV     R4,R3           ;GET TSSR CONTENTS
          1578  005554  042703  176377                  BIC     #^CHIADDR,R3    ;CLEAR ALL BUT EXTENDED ADDRESS
          1579  005560  001411                          BEQ     30$             ;DON'T PRINT IF ZERO
          1580  005562                                  PRINTX  #TEXASC,R3      ;PRINT THE EXTENDED ADDRESS BITS
                005562  010346                          MOV     R3,-(SP)
                005564  012746  005756                  MOV     #TEXASC,-(SP)
                005570  012746  000002                  MOV     #2,-(SP)
                005574  010600                          MOV     SP,R0
                005576  104415                          TRAP    C$PNTX
                005600  062706  000006                  ADD     #6,SP
          1581  005604  022704  100210          30$:    CMP     #100210,R4      ;CHECK FOR MEDIA ERROR
          1582  005610  001003                          BNE     31$             ;BR, IF PROBABLY NOT TAPE ERROR
          1583  005612  012737  005672  002146          MOV     #EPRT3,EPRTSW   ;"PROBABLY MEDIA RELETED ERROR - BAD TAPE"
          1584  005620  005737  002146          31$:    TST     EPRTSW          ;CHECK FOR THE SWITCH EMPTY
          1585  005624  001003                          BNE     310$            ;BR, IF SWITCH IS NOT EMPTY
          1586  005626  012737  005672  002146          MOV     #EPRT1,EPRTSW   ;SET SWITCH TO DEFAULT
          1587  005634  013737  002146  005644  310$:   MOV     EPRTSW,32$+2    ;PUT REAL SWITCHABLE MESSAGE IN PLACE
          1588  005642                          32$:    PRINTB  #EPRT1          ;PRINT THE ERROR MESSAGE
                005642  012746  005672                  MOV     #EPRT1,-(SP)
                005646  012746  000001                  MOV     #1,-(SP)
                005652  010600                          MOV     SP,R0
                005654  104414                          TRAP    C$PNTB
                005656  062706  000004                  ADD     #4,SP
          1589  005662  012737  005672  002146          MOV     #EPRT1,EPRTSW   ;RESET TO NORMAL ERROR POINTER
          1590  005670  000207                          RTS     PC              ;RETURN TO CALLER
          1591
          1592  005672                          EPRT2:
```

```
1593 005672                                EPRT3:
1594 005672     045     116     045  EPRT1:  .ASCIZ  '%N%A *****REPLACE CONTROLLER*****%S'
1595 005736     045     116     045  TSSRFOR:        .ASCIZ  '%N%A TSSR = %O6'
1596 005756     045     116     045  TEXASC:         .ASCIZ  '%N%A Extended Address Bits = %O6'
1597 006017     045     116     045  TCOASC:         .ASCIZ  '%N%A Termination Class Code = %T'
1598 006060     045     116     045  TFCASC:         .ASCIZ  '%N%A Fatal Termination Class Code = %T'
1599 006127     045     116     045  TSSDEF:         .ASCIZ  '%N%A TSSR Bits Set: %T'
1600 006156     045     116     045  AMBTSSR:        .ASCIZ  '%N%A TSSR Contents Are Ambiguous'
1601                                            .EVEN
1602 006220  006240  006263  006311  TCOCOD: .WORD   1$,2$,3$,4$,5$,6$,7$,8$
1603 006240     116     157     162  1$:     .ASCIZ  'Normal Termination'
1604 006263     124     145     162  2$:     .ASCIZ  'Termination Condition'
1605 006311     124     141     160  3$:     .ASCIZ  'Tape Status Alert'
1606 006333     106     165     156  4$:     .ASCIZ  'Function Reject'
1607 006353     122     145     143  5$:     .ASCIZ  'Recoverable Error - Tape Position One Record Down'
1608 006435     122     145     143  6$:     .ASCIZ  'Recoverable Error - Tape Was Not Moved'
1609 006504     125     156     162  7$:     .ASCIZ  'Unrecoverable Error'
1610 006530     106     141     164  8$:     .ASCIZ  'Fatal Controller Error'
1611                                            .EVEN
1612
1613 006560  006570  006624  006635  TSFCOD: .WORD   1$,2$,3$,4$
1614 006570     111     156     164  1$:     .ASCIZ  'Internal Diagnostic Failure'
1615 006624     122     145     163  2$:     .ASCIZ  'Reserved'
1616 006635     102     165     163  3$:     .ASCIZ  'Bus Interface or Sanity Check Error'
1617 006701     122     145     163  4$:     .ASCIZ  'Reserved'
1618                                            .EVEN
```

```
                                        .SBTTL  PRIPKT  - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

                                ;+
                                ;THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
                                ;THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
                                ;
                                ;INPUT:
                                ;
                                ;       R0      NUMBER OF WORDS IN PACKET
                                ;       R3      HIGH ORDER COMMAND PACKET ADDRESS
                                ;       R4      ADDRESS OF COMMAND PACKET
                                ;
                                ;       NOTE:   R3 IS IGNORED IF THE KTENABLE FLAG IS CLEAR.
                                ;-

006712                          PRIPKT::
006712                                  SAVREG                          ;SAVE THE REGISTERS
006716  010005                          MOV     R0,R5                   ;SAVE NO. OF WORDS IN PACKET
006720  005737  003102                  TST     KTENABLE                ;ABOVE 28K UNDER TEST?
006724  001001                          BNE     10$                     ;BR IF YES
006726  005003                          CLR     R3                      ;SET HIGH ORDER ADDRESS TO 0
006730  010301                  10$:    MOV     R3,R1                   ;COPY HIGH ORDER ADDRESS
006732  010400                          MOV     R4,R0                   ;GET LOWER ADDRESS
006734  006100                          ROL     R0                      ;SHIFT BIT 15 INTO C BIT
006736  006101                          ROL     R1                      ;AND INTO HIGH ORDER.
006740                                  PRINTB  #PKTADD,R1,R4           ;PRINT PACKET ADDRESS
006740  010446                          MOV     R4,-(SP)
006742  010146                          MOV     R1,-(SP)
006744  012746  007116                  MOV     #PKTADD,-(SP)
006750  012746  000003                  MOV     #3,-(SP)
006754  010600                          MOV     SP,R0
006756  104414                          TRAP    C$PNTB
006760  062706  000010                  ADD     #10,SP
006764  010300                  15$:    MOV     R3,R0                   ;GET HIGH ORDER ADDRESS
006766  001404                          BEQ     20$                     ;BR IF NOT ABOVE 28K.
006770  010401                          MOV     R4,R1                   ;GET LOW ORDER ADDRESS
006772  004737  020112                  JSR     PC,SETMAP               ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
006776  010004                          MOV     R0,R4                   ;GET RETURNED PAR6 ADDRESS BIAS
007000  005001                  20$:    CLR     R1                      ;SAVE WORD NUMBER
007002  012402                  25$:    MOV     (R4)+,R2                ;GET PACKET CONTENTS
007004                                  PRINTB  #PKTFRM,R1,R2           ;PRINT THE DATA
007004  010246                          MOV     R2,-(SP)
007006  010146                          MOV     R1,-(SP)
007010  012746  007060                  MOV     #PKTFRM,-(SP)
007014  012746  000003                  MOV     #3,-(SP)
007020  010600                          MOV     SP,R0
007022  104414                          TRAP    C$PNTB
007024  062706  000010                  ADD     #10,SP
007030  005201                          INC     R1                      ;NEXT WORD NUMBER
007032  020105                          CMP     R1,R5                   ;DONE ALL PACKET WORDS?
007034  002762                          BLT     25$                     ;LOOP TILL ALL DONE
007036                                  PRINTB  #PKTNEW                 ;JUST A COUPLE NEW LINES
007036  012746  007153                  MOV     #PKTNEW,-(SP)
007042  012746  000001                  MOV     #1,-(SP)
007046  010600                          MOV     SP,R0
007050  104414                          TRAP    C$PNTB
007052  062706  000004                  ADD     #4,SP
```

PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET

```
1658 007056  000207                          RTS     PC              ;RETURN
1659
1660 007060     045     116     045  PKTFRM: .ASCIZ  '%N%A Packet Word #%D1%A = %06'
1661 007116     045     116     045  PKTADD: .ASCIZ  '%N%A Packet Address = %01%05'
1662
1663 007153     045     116     045  PKTNEW: .ASCIZ  '%N%N%A  '
1664                                  .EVEN
1665
```

PRIBXOR - PRINT EXPD, RECV AND XOR BYTE

```
                                        .SBTTL  PRIBXOR - PRINT EXPD, RECV AND XOR BYTE

                                ;+
                                ;
                                ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE DATA BYTE
                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      RECEIVED DATA
                                ;       R2      EXPECTED DATA
                                ;
                                ;OUTPUT:
                                ;
                                ;       R0      XOR OF EXPECTED/RECEIVED DATA
                                ;
                                ;-

007164                          PRIBXOR::
007164                                  SAVREG                          ;SAVE THE REGISTERS
007170  010203                          MOV     R2,R3                   ;EXPECTED DATA
007172                                  XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007202  012700  177400                  MOV     #^C<377>,R0             ;BYTE MASK
007206  040001                          BIC     R0,R1                   ;SAVE LOW BYTE RECV
007210  040002                          BIC     R0,R2                   ;SAVE LOW BYTE EXPD
007212  040003                          BIC     R0,R3                   ;SAVE LOW BYTE XOR
007214                                  PRINTB  #XORBFOR,R2,R1,R3 ;PRINT THE MESSAGE
007214  010346                          MOV     R3,-(SP)
007216  010146                          MOV     R1,-(SP)
007220  010246                          MOV     R2,-(SP)
007222  012746  007246                  MOV     #XORBFOR,-(SP)
007226  012746  000004                  MOV     #4,-(SP)
007232  010600                          MOV     SP,R0
007234  104414                          TRAP    C$PNTB
007236  062706  000012                  ADD     #12,SP
007242  010300                          MOV     R3,R0                   ;R0 HAS XOR ON RETURN
007244  000207                          RTS     PC                      ;RETURN TO CALLER

007246      045     116     045 XORBFOR:        .ASCIZ  '%N%A EXPD: %03%A RECV: %03%A XOR: %03'
                                        .EVEN
```

```
                        .SBTTL  PRIXOR  - PRINT EXPD, RECV AND XOR

                ;*
                ;
                ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE TWO
                ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
                ;
                ;INPUTS:
                ;
                ;       R1      RECEIVED DATA
                ;       R2      EXPECTED DATA
                ;
                ;OUTPUT:
                ;
                ;       R0      XOR OF EXPECTED/RECEIVED DATA
                ;
                ;-

007314          PRIXOR:
007314                  SAVREG                          ;SAVE THE REGISTERS
007320  010203          MOV     R2,R3                   ;EXPECTED DATA
007322                  XOR     R1,R3                   ;FORM THE EXCLUSIVE OR
007332                  PRINTB  #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
007332  010346          MOV     R3,-(SP)
007334  010146          MOV     R1,-(SP)
007336  010246          MOV     R2,-(SP)
007340  012746  007364  MOV     #XORFOR,-(SP)
007344  012746  000004  MOV     #4,-(SP)
007350  010600          MOV     SP,R0
007352  104414          TRAP    C$PNTB
007354  062706  000012  ADD     #12,SP
007360  010300          MOV     R3,R0                   ;R0 HAS XOR ON RETURN
007362  000207          RTS     PC                      ;RETURN TO CALLER

007364      045     116     045  XORFOR: .ASCIZ  '%N%A EXPD: %06%A RECV: %06%A XOR: %06'
                                .EVEN
```

```
                                        .SBTTL  PRIEQU  - PRINT BIT NUMBERS AS ASCII EQUIVALENT

                                ;+
                                ;
                                ;ROUTINE TO CONVERT BIT VALUES TO ASCII AND PRINT THE STRING
                                ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
                                ;
                                ;INPUTS:
                                ;
                                ;       R0      OCTAL VALUE TO CONVERT
                                ;       R1      TABLE OF POINTERS TO ASCII EQUIVALENT
                                ;
                                ;-

007432                          PRIEQU:
007432                                  SAVREG                          ;SAVE THE REGISTERS
007436  000207                          RTS     PC                      ;RETURN TO CALLER



                                        .SBTTL  PRIRAM  - PRINT RAM ADDRESS
                                ;+
                                ;
                                ;PRINT CONTROLLER RAM ADDRESS.
                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                                ;
                                ;INPUTS:
                                ;
                                ;       R4      RAM ADDRESS
                                ;
                                ;
007440                          PRIRAM:
007440                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
007444                                  PRINTB  @RAMFOR,R4      ;PRINT RAM ADDRESS IN ERROR
007444  010446                          MOV     R4,-(SP)
007446  012746  007470                  MOV     @RAMFOR, (SP)
007452  012746  000002                  MOV     #2,-(SP)
007456  010600                          MOV     SP,R0
007460  104414                          TRAP    C$PNTB
007462  062706  000006                  ADD     #6,SP
007466  000207                          RTS     PC              ;RETURN

007470     045     116     045  RAMFOR: .ASCIZ  '%N%A CONTROLLER RAM ADDRESS = %O6'
                                        .EVEN



                                        .SBTTL  PRIADD  - PRINT MEMORY ERROR ADDRESS
                                ;+
                                ;
                                ;PRINT MEMORY ADDRESS
                                ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
                                ;
                                ; IMPLICIT INPUTS
                                ;
                                ;       ERRHI     HIGH ORDER ADDRESS
                                ;       ERRLO     LOW ORDER ADDRESS
```

PRIADD - PRINT MEMORY ERROR ADDRESS

```
1781                                    ;
1782                                    ;-
1783 007532                             PRIADD:
1784 007532                                     SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
1785 007536  013700  002202                     MOV     ERRHI,R0                ;GET HIGH ADDRESSS
1786 007542  013701  002204                     MOV     ERRLO,R1                ;GET LOW ADDRESS
1787 007546  010102                             MOV     R1,R2                   ;COPY LOW ADDRESS
1788 007550  006101                             ROL     R1                      ;SHIFT BIT 15 TO C BIT
1789 007552  006100                             ROL     R0                      ;SHIFT INTO HIGH ORDER
1790 007554                                     PRINTB  #PRIA0,R0,R2            ;PRINT MEMORY ADDRESS IN ERROR
     007554  010246                             MOV     R2,-(SP)
     007556  010046                             MOV     R0,-(SP)
     007560  012746  007602                     MOV     #PRIA0,-(SP)
     007564  012746  000003                     MOV     #3, (SP)
     007570  010600                             MOV     SP,R0
     007572  104414                             TRAP    C$PNTB
     007574  062706  000010                     ADD     #10,SP
1791 007600  000207                             RTS     PC                      ;RETURN
1792
1793 007602     045     116     045  PRIA0:    .ASCIZ  '%N%A MEMORY ERROR ADDRESS = %01%05'
1794                                             .EVEN
1795
1796
1797                                             .SBTTL  PRITADD - PRINT MEMORY TEST ADDRESS
1798                                    ;+
1799                                    ;
1800                                    ;PRINT MEMORY ADDRESS
1801                                    ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
1802                                    ;
1803                                    ; IMPLICIT INPUTS
1804                                    ;
1805                                    ;       ERRHI   - HIGH ORDER ADDRESS
1806                                    ;       ERRLO   - LOW ORDER ADDRESS
1807                                    ;
1808                                    ;-
1809 007646                             PRITADD:
1810 007646                                     SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
1811 007652  013700  002202                     MOV     ERRHI,R0                ;GET HIGH ADDRESSS
1812 007656  013701  002204                     MOV     ERRLO,R1                ;GET LOW ADDRESS
1813 007662  010102                             MOV     R1,R2                   ;COPY LOW ADDRESS
1814 007664  006101                             ROL     R1                      ;SHIFT BIT 15 TO C BIT
1815 007666  006100                             ROL     R0                      ;SHIFT INTO HIGH ORDER
1816 007670                                     PRINTB  #PRITO,R0,R2            ;PRINT MEMORY ADDRESS IN ERROR
     007670  010246                             MOV     R2,-(SP)
     007672  010046                             MOV     R0,-(SP)
     007674  012746  007716                     MOV     #PRITO,-(SP)
     007700  012746  000003                     MOV     #3,-(SP)
     007704  010600                             MOV     SP,R0
     007706  104414                             TRAP    C$PNTB
     007710  062706  000010                     ADD     #10,SP
1817 007714  000207                             RTS     PC                      ;RETURN
1818
1819 007716     045     116     045  PRITO:    .ASCIZ  '%N%A MEMORY TEST ADDRESS = %01%05'
1820                                             .EVEN
1821
1822
1823
```

SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

```
                                        .SBTTL  SPACE   - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

                                ;+
                                ;
                                ;ROUTINE TO ISSUE A SPACE RECORDS
                                ;COMMAND (FORWARD OR REVERSE)
                                ;
                                ;INPUT:
                                ;
                                ;       R3      NUMBER OF RECORDS TO BE SPACED OVER
                                ;               BIT15 CONTROLS DIRECTION
                                ;               BIT15 = 0 IS FORWARD
                                ;               BIT15 = 1 IS REVERSE
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;
                                ;       REQUIRES A WRITE CHARACTERISTICS DONE PREVIOUSLY
                                ;
                                ;OUTPUT:
                                ;
                                ;       CARRY   SET - SPACE RECORDS COMMAND OK
                                ;               CLR - SPACE RECORDS FAILED
                                ;
                                ;
                                ;       R0      THE CONTENTS OF R4 IS MOVED TO R0
                                ;
                                ;
                                ;IMPLICIT OUTPUT:
                                ;
                                ;       TAPE HAS BEEN MOVED
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;
                                ;-

                                SPACE::
007760                                  SAVREG                  ;SAVE THE GENERAL REGISTERS
007764  012737  000764  010150            MOV     #500.,SDELAY    ;SET UP DELAY
007772  012737  140010  010140            MOV     #140010,80$     ;SET UP COMMAND, SPACE FORWARD
010000  005703                          TST     R3              ;CHECK FOR DIRECTION
010002  100403                          BMI     5$              ;BR, IF REVERSE INDICATED
010004  010337  010142                  MOV     R3,90$          ;LOAD UP NUMBER OF RECORDS TO SPACE
010010  000407                          BR      10$             ;GO DO COMMAND
010012  042703  100000          5$:     BIC     #BIT15,R3       ;CLEAR DIRECTION BIT
010016  010337  010142                  MOV     R3,90$          ;LOAD UP NUMBER OF RECORDS TO SPACE
010022  052737  000400  010140            BIS     #BIT8,80$       ;SET REVERSE BIT IN COMMAND PACKET
010030  012704  010140          10$:    MOV     #80$,R4         ;SET UP R4 WITH PACKET ADDRESS
010034  010465  177776                    MOV     R4,TSDB(R5)     ;SEND OUT COMMAND
010040  004737  016744          15$:    JSR     PC,WAITF        ;WAIT FOR SSR
010044  103420                          BCS     20$             ;BR, IF SSR IS SET AND OK
010046                                  DELAY   250             ;DELAY ABOUT .25 SECONDS
010046  012727  000250                  MOV     #250,(PC)+
010052  000000                          .WORD   0
010054  013727  002116                  MOV     L$DLY,(PC)+
010060  000000                          .WORD   0
010062  005367  177772                  DEC     -6(PC)
010066  001375                          BNE     .-4
```

```
        010070  005367  177756                  DEC     -22(PC)
        010074  001367                          BNE     .-20
   1876 010076  005337  010150                  DEC     SDELAY          ;BUMP DELAY COUNTER DOWN
   1877 010102  001356                          BNE     15$             ;BR, IF MORE DELAY
   1878 010104  000411                          BR      60$             ;BR IF TROUBLE CARRY = CLEAR
   1879 010106  016501  000000          20$:    MOV     TSSR(R5),R1     ;READ TSSR
   1880 010112  012702  000200                  MOV     #SSR,R2         ;SET UP EXPECTED
   1881 010116  020201                  25$:    CMP     R2,R1           ;ARE THEY OK
   1882 010120  001401                          BEQ     40$             ;BR, IF EQUAL = OK
   1883 010122  000402                          BR      60$             ;TROUBLE EXIT
   1884 010124  000261                  40$:    SEC                     ;SET CARRY NO TROUBLE
   1885 010126  000401                          BR      70$             ;EXIT
   1886 010130  000241                  60$:    CLC                     ;CARRY CLEAR = ERROR
   1887 010132                          70$:
   1888 010132  010400                          MOV     R4,R0           ;PASS PACKET ADDRESS
   1889 010134  000207                          RTS     PC              ;RETURN
```

```
 1891                                   ;
 1892                                   ;
 1893                                   ;
 1894                                   ;PACKET FOR SPACE COMMAND
 1895                                   ;
 1897 010136                                    .BLKB   10-<.-TUV2A&7>
 1899                                   ;
 1900                                   ;COMMAND WORD
 1901 010140  000000                    80$:    .WORD
 1902                                   ;NUMBER OF RECORDS TO BE SPACED OVER WORD
 1903 010142  000000                    90$:    .WORD
 1904 010144  000000                            .WORD
 1905 010146  000000                            .WORD
 1906 010150  000000                    SDELAY: .WORD   0               ;DELAY COUNTER
 1907                                           .EVEN
```

```
                                        .SBTTL  WRTCHR  - WRITE CHARACTERISTICS COMMAND

                                ;+
                                ;
                                ;ROUTINE TO ISSUE A WRITE CHARACTERISTICS
                                ;COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
                                ;
                                ;INPUT:
                                ;
                                ;       R4      ADDRESS OF PACKET FROM TEST
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;       REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
                                ;
                                ;OUTPUT:
                                ;
                                ;       R0      TSSR CONTENTS
                                ;       CARRY   SET - WRITE CHARACTERISTICS COMMAND OK
                                ;               CLR - WRITE CHARACTERISTICS FAILED
                                ;
                                ;IMPLICIT OUTPUT:
                                ;
                                ;       MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
                                ;       SOFTWARE SWITCHES SET AS FOLLOWS:
                                ;               BENBSW = BUFFER ENABLE SWITCH ON OR OFF
                                ;
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;
                                ;-

010152                          WRTCHR::
010152                                  SAVREG                  ;SAVE THE GENERAL REGISTERS
010156  005037  002174                  CLR     BENBSW          ;CLEAR BUFFER ENABLE SWITCH
010162  010465  177776          10$:    MOV     R4,TSDB(R5)     ;SEND OUT COMMAND
010166  004737  017060                  JSR     PC,CHKTSSR      ;WAIT FOR SSR
010172  103401                          BCS     20$             ;BR, IF SSR IS SET AND OK
010174  000423                          BR      60$             ;BR IF TROUBLE CARRY = CLEAR
010176  016501  000000          20$:    MOV     TSSR(R5),R1     ;READ TSSR
010202  012702  000200                  MOV     #SSR,R2         ;SET UP EXPECTED
010206  032701  000100                  BIT     #OFL,R1         ;WAS OFF LINE SET IN TSSR
010212  001402                          BEQ     25$             ;BR, IF NO OFL SET
010214  052702  000100                  BIS     #OFL,R2         ;MAKE THEM LOOK ALIKE
010220  020201                  25$:    CMP     R2,R1           ;ARE THEY OK
010222  001401                          BEQ     40$             ;BR, IF EQUAL = OK
010224  000407                          BR      60$             ;TROUBLE EXIT
010226  062704  000010          40$:    ADD     #8.,R4          ;POINT TO WRT CHARA DATA PACKET
010232  011403                          MOV     (R4),R3         ;GET ADDRESS OF MESSAGE BUFFER
010234  010337  002716                  MOV     R3,MESBFA       ;STORE FOR PRINT ROUTINES
010240  000261                          SEC                     ;SET CARRY NO TROUBLE
010242  000401                          BR      70$             ;EXIT
010244  000241                  60$:    CLC                     ;CARRY CLEAR = ERROR
010246  016500  000000          70$:    MOV     TSSR(R5),R0     ;RETURN TSSR CONTENTS
010252  000207                          RTS     PC              ;RETURN
```

```
                                        .SBTTL  REWIND  - POSITION TAPE (REWIND) COMMAND

                                ;+
                                ;
                                ;THIS ROUTINE WILL REWIND THE SELECTED TAPE.
                                ;
                                ;       CAUTION: THE ROUTINE DOES NOT WAIT FOR BOT
                                ;                TO ARRIVE. ALSO THE CALLER MUST CHECK FOR
                                ;                SSR TO SET IN THE TSSR
                                ;
                                ;
                                ;CALLING SEQUENCE:
                                ;
                                ;       DO A SOFT INIT
                                ;       DO A WRITE CHARACTERISTICS
                                ;       JSR     PC,REWIND
                                ;
                                ;INPUT:
                                ;
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;
                                ;
                                ;OUTPUT
                                ;
                                ;       R0      THE CONTENTS OF R4 IS PASSED TO R0
                                ;
                                ;
                                ;-
                                REWIND::
                                        SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
010254  012704  010350                  MOV     #RWPACK,R4      ;GET PACKET ADDRESS
010260  010465  177776                  MOV     R4,TSDB(R5)     ;SEND PACKET ADDRESS TO EXECUTE
010264  012703  000550                  MOV     #360.,R3        ;ENOUGH TIME FOR 2400' REEL TO REWIND
010270  004737  016744          10$:    JSR     PC,WAITF        ;WAIT FOR SSR TO SET
010274  103417                          BCS     20$             ;LEAVE WHEN SSR IS SET
010276                                  DELAY   250.            ;WAIT FOR .25 SECONDS
010276  012727  000372                  MOV     #250.,(PC)+
010302  000000                          .WORD   0
010304  013727  002116                  MOV     L$DLY,(PC)+
010310  000000                          .WORD   0
010312  005367  177772                  DEC     -6(PC)
010316  001375                          BNE     .-4
010320  005367  177756                  DEC     -22(PC)
010324  001367                          BNE     .-20
010326  005303                          DEC     R3              ;BUMP COUNTER DOWN
010330  001357                          BNE     10$             ;KEEP GOING
010332  000241                          CLC                     ;CLEAR CARRY TO SET ERROR
010334  010400                  20$:    MOV     R4,R0           ;PASS THE PACKET ADDRESS
010336  000207                          RTS     PC              ;RETURN

010340                                  .BLKB   10-<.-TUV2A&7>
010350                          RWPACK:
010350  102010                          .WORD   102010          ;POSTION COMMAND (REWIND)
010352  000000                          .WORD   0               ;NOT USED
```

```
                                        .SBTTL  CKRAM   - COMPARE RAM TO I/O PACKET

                                ;+
                                ;
                                ;ROUTINE TO READ THE FIRST 8 BYTES FROM RAM
                                ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
                                ;
                                ;INPUT:
                                ;
                                ;       R4      ADDRESS OF THE COMMAND PACKET
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;
                                ;OUTPUT:
                                ;
                                ;       CARRY   SET - RAM MATCHES PACKET
                                ;               CLR - RAM DOES NOT MATCH PACKET
                                ;
                                ;IMPLICIT OUTPUT:
                                ;
                                ;       THE TABLE RAMDATA IS FILLED WITH THE
                                ;       DATA HELD IN RAM.
                                ;       RAMSIZ IS SET TO 8. FOR PRAMPKT ROUTINE
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;
                                ;-

010354                          CKRAM::
010354                                  SAVREG                  ;SAVE THE GENERAL REGISTERS
010360  012701  002206                  MOV     #RAMDATA,R1     ;ADDRESS TO SAVE THE RAM DATA
010364  012702  000020                  MOV     #RMPKTBEG,R2    ;BYTE ADDRESS OF FIRST RAM DATA
010370  005003                          CLR     R3              ;CLEAR THE ERROR FLAG
010372  004737  017060                  JSR     PC,CHKTSSR      ;WAIT FOR SSR
010376  004737  017060          10$:    JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
010402  110265  177777                  MOVB    R2,TSDBH(R5)    ;SELECT NEXT RAM ADDRESS
010406  004737  017060                  JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
010412  116511  177776                  MOVB    TSBAL(R5),(R1)  ;READ THE RAM DATA
010416  122124                          CMPB    (R1)+,(R4)+     ;COMPARE TO EXPECTED
010420  001401                          BEQ     20$             ;BRANCH IF OK
010422  005203                          INC     R3              ;SET ERROR FLAG
010424  005202                  20$:    INC     R2              ;ADDRESS OF NEXT RAM LOCATION
010426  020227  000027                  CMP     R2,#RMPKTEND    ;REACHED END YET ?
010432  003761                          BLE     10$             ;BRANCH TILL ALL READ
010434  005703                          TST     R3              ;WAS AN ERROR FOUND ?
010436  001402                          BEQ     30$             ;BRANCH IF NOT
010440  000241                          CLC                     ;CLEAR CARRY TO SHOW ERROR
010442  000401                          BR      50$             ;AND EXIT
010444  000261                  30$:    SEC                     ;SHOW GOOD COMPARE
010446  012737  000010  002246  50$:    MOV     #8.,RAMSIZ      ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
010454  000207                          RTS     PC              ;RETURN
```

```
                                        .SBTTL RAMER - READ AND DISPLAY SELECTED RAM
                                ;+
                                ;
                                ;ROUTINE TO READ THE SELECTED RAM LOCATIONS
                                ;
                                ;INPUT:
                                ;
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;       CONSOLE WILL ALSO BE PRINTED TO
                                ;
                                ;IMPLICIT OUTPUT:
                                ;
                                ;       THE TABLE RAMDATA IS FILLED WITH THE
                                ;       DATA HELD IN RAM.
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;
                                ;-

010456                          RAMER::
010456                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
010462  013705  010642                  MOV     RAMR5H,R5               ;RESET R5 TO FIRST DEVICE REGISTER
010466  012701  002206                  MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
010472  013702  010640                  MOV     RAMHLD,R2               ;BYTE ADDRESS OF THE FIRST RAM DATA
010476  013703  002246                  MOV     RAMSIZ,R3               ;SET THE SIZE OF THE READ UP
010502  004737  017060          10$:    JSR     PC,CHKTSSR              ;WAIT FOR THE SSR TO SET
010506  110265  177777                  MOVB    R2,TSDBH(R5)            ;SELECT NEXT RAM ADDRESS
010512  004737  017060                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
010516  116521  177776                  MOVB    TSBAL(R5),(R1)+         ;READ THE RAM DATA
010522  062702  000001          20$:    ADD     #1,R2                   ;ADDRESS OF THE NEXT RAM LOCATION
010526  077313                          SOB     R3,10$                  ;NUMBER OF LOCATIONS COUNTER
010530  013704  002246                  MOV     RAMSIZ,R4               ;GET THE RAM SIZE
010534  013702  010640                  MOV     RAMHLD,R2               ;GET THE STARTING RAM ADDRESS
010540  060204                          ADD     R2,R4                   ;CALCULATE THE END ADDRESS
010542  162704  000001                  SUB     #1,R4                   ;CORRECT VALUE OF PRINTOUT
010546                                  PRINTX  #RAMIOP,R2,R4           ;RAM ADDRESS = 10 - 17, ETC.
010546  010446                          MOV     R4,-(SP)
010550  010246                          MOV     R2,-(SP)
010552  012746  010644                  MOV     #RAMIOP,-(SP)
010556  012746  000003                  MOV     #3,-(SP)
010562  010600                          MOV     SP,R0
010564  104415                          TRAP    C$PNTX
010566  062706  000010                  ADD     #10,SP
010572  012701  002206                  MOV     #RAMDATA,R1             ;ADDRESS OF WHERE RAM DATA IS
010576  013703  002246                  MOV     RAMSIZ,R3               ;THE SIZE OF THE RAM FIELD READ
010602  005004                  30$:    CLR     R4                      ;NO EXTRA DATA LEFT OVER
010604  112104                          MOVB    (R1)+,R4                ;PICK UP BYTE OF RAM DATA
010606  042704  177400                  BIC     #177400,R4              ;GET RID OF SIGN EXTEND
010612                                  PRINTX  #RAMPD,R4               ;"010 211 111 222 377 000 123 134 ETC."
010612  010446                          MOV     R4,-(SP)
010614  012746  010715                  MOV     #RAMPD, (SP)
010620  012746  000002                  MOV     #2,-(SP)
010624  010600                          MOV     SP,R0
010626  104415                          TRAP    C$PNTX
010630  062706  000006                  ADD     #6,SP
010634  077316                          SOB     R3,30$                  ;LOOP UNTIL ALL PRINTED
```

```
 2113 010636  000207                 50$:    RTS     PC              ;RETURN
 2114
 2115 010640  000000                 RAMHLD: .WORD   0               ;RAM ADDR HOLDER 1ST ADDRESS
 2116 010642  000000                 RAMR5H: .WORD   0               ;HOLDS R5 FOR LATER
 2117 010644     045     116     045 RAMIOP: .ASCIZ  '%N%A Ram Address (Octal) = %03%A   %03%N'
 2118 010715     045     101     040 RAMPD:  .ASCIZ  '%A %03%A '
 2119                                 .EVEN
 2120
```

```
                                        .SBTTL  CKRAM2  - COMPARE RAM TO I/O CHARACTERISTICS DATA
                                ;+
                                ;
                                ;ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
                                ;MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
                                ;
                                ;INPUT:
                                ;
                                ;       R4      ADDRESS OF THE CHARACTERISTICS DATA
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;
                                ;OUTPUT:
                                ;
                                ;       CARRY   SET - RAM MATCHES PACKET
                                ;               CLR - RAM DOES NOT MATCH PACKET
                                ;
                                ;IMPLICIT OUTPUT:
                                ;
                                ;       THE TABLE RAMDATA IS FILLED WITH THE
                                ;       DATA HELD IN RAM.
                                ;       RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;
                                ;-

010730                          CKRAM2::
010730                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
010734  012701  002206                  MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
010740  012702  000167                  MOV     #RMCHBEG,R2             ;BYTE ADDRESS OF FIRST RAM DATA
010744  005003                          CLR     R3                      ;CLEAR THE ERROR FLAG
010746  004737  017060                  JSR     PC,CHKTSSR              ;WAIT FOR SSR
010752  004737  017060          10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
010756  110265  177777                  MOVB    R2,TSDBH(R5)            ;SELECT NEXT RAM ADDRESS
010762  004737  017060                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
010766  116511  177776                  MOVB    TSBAL(R5),(R1)          ;READ THE RAM DATA
010772  122124                          CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
010774  001401                          BEQ     20$                     ;BRANCH IF OK
010776  005203                          INC     R3                      ;SET ERROR FLAG
011000  005202                  20$:    INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
011002  012737  000010  002246          MOV     #8.,RAMSIZ              ;ASSUME NORMAL NOT SET
011010  020227  000176                  CMP     R2,#RMCHEND-2           ;REACHED END YET ?
011014  003756                          BLE     10$                     ;BRANCH TILL ALL READ
011016  005703                  27$:    TST     R3                      ;WAS AN ERROR FOUND ?
011020  001402                          BEQ     30$                     ;BRANCH IF NOT
011022  000241                          CLC                             ;CLEAR CARRY TO SHOW ERROR
011024  000401                          BR      50$                     ;AND EXIT
011026  000261                  30$:    SEC                             ;SHOW GOOD COMPARE
011030  000207                  50$:    RTS     PC                      ;RETURN
```

```
                                        .SBTTL  CKMSG   - COMPARE WRITE CHAR. MESSAGE BUFFERS
                                ;*
                                ;
                                ;ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
                                ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
                                ;ERROR PRINT ROUTINES.
                                ;
                                ;INPUT:
                                ;
                                ;       R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
                                ;       R2      EXPD MESSAGE BUFFER ADDRESS
                                ;OUTPUT:
                                ;
                                ;       CARRY   SET - MESSAGE BUFFERS MATCH
                                ;               CLR -MESSAGE BUFFERS DON'T MATCH
                                ;
                                ;IMPLICIT OUTPUT:
                                ;
                                ;       EXPMSG          BUFFER IS SET TO EXPD DATA
                                ;       RECMSG          BUFFER IS SET TO RECV DATA
                                ;       RCVHIADD        SET TO HIGH ORDER ADDRESS OF RECV
                                ;       RCVLOADD        SET TO LOW ORDER ADDRESS OF RECV
                                ;
                                ;-
011032                          CKMSG::
011032                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
011036  010037  002250                  MOV     R0,RCVHIADD     ;SAVE RECV HIGH ADDRESS
011042  010137  002252                  MOV     R1,RCVLOAD      ;SAVE RECV LOW ADDRESS
011046  005737  003102                  TST     KTENABLE        ;TESTING ABOVE 28K?
011052  001403                          BEQ     10$             ;BR IF NO
011054  004737  020112                  JSR     PC,SETMAP       ;RETURN ADDRESS BIASED TO PAR6 IN R0
011060  010001                          MOV     R0,R1           ;GET RETURNED ADDRESS BIASED TO PAR6
011062  005004                  10$:    CLR     R4              ;WORD IN BUFFER
011064  005003                          CLR     R3              ;CLEAR ERROR SEEN FLAG
011066  010205                          MOV     R2,R5           ;GET EXPD BUFFER ADDRESS
011070  011264  002266          15$:    MOV     (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
011074  011164  002432                  MOV     (R1),RECMSG(R4) ;SAVE RECV FOR ERROR REPORT
011100  022221                          CMP     (R2)+,(R1)+     ;EXPD EQUAL RECV?
011102  001401                          BEQ     25$             ;BR IF YES
011104  005203                          INC     R3              ;SET ERROR SEEN FLAG
011106  062704  000002          25$:    ADD     #2,R4           ;POINT TO NEXT WORD ADDRESS
011112  020427  000014                  CMP     R4,#14          ;DONE FIRST 7 WORDS?
011116  003764                          BLE     15$             ;BR IF NO
011120  032765  000200  000012          BIT     #X2.EXTF,XST2(R5);IS EXTENDED FEATURES SET IN EXPD?
011126  001403                          BEQ     50$             ;BR IF NO
011130  020427  000016                  CMP     R4,#16          ;DONE EXTENDED FEATURES WORD?
011134  003755                          BLE     15$             ;BR IF NO
011136  005703                  50$:    TST     R3              ;ANY ERRORS SEEN?
011140  001402                          BEQ     55$             ;BR IF NO
011142  000241                          CLC                     ;SET FAILURE
011144  000401                          BR      60$             ;
011146  000261                  55$:    SEC                     ;SET SUCCESS
011150  000207                  60$:    RTS     PC              ;RETURN
```

```
                                        .SBTTL  CKMSG2  - COMPARE EXPD RECV MESSAGE BUFFERS
                                ;*
                                ;
                                ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
                                ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
                                ;ERROR PRINT ROUTINES.
                                ;
                                ;INPUT:
                                ;
                                ;       R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
                                ;       R2      EXPD MESSAGE BUFFER ADDRESS
                                ;       R3      NUMBER OF BYTES TO COMPARE
                                ;
                                ;OUTPUT:
                                ;
                                ;       CARRY   SET - MESSAGE BUFFERS MATCH
                                ;               CLR - MESSAGE BUFFERS DON'T MATCH
                                ;
                                ;IMPLICIT OUTPUT:
                                ;
                                ;       EXPMSG          BUFFER IS SET TO EXPD DATA
                                ;       RECMSG          BUFFER IS SET TO RECV DATA
                                ;       RCVHIADD        SET TO HIGH ORDER ADDRESS OF RECV
                                ;       RCVLOADD        SET TO LOW ORDER ADDRESS OF RECV
                                ;
                                ;-
011152                          CKMSG2::
011152                                  SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
011156  020327  000144                  CMP     R3,#RECMSG-EXPMSG;@@D IS COUNT ABOVE MAX ALLOWED?
011162  003412                          BLE     5$                      ;@@D BR IF NO
011164  012703  000144                  MOV     #RECMSG-EXPMSG,R3;@@D
011170                                  PRINTF  #DEBUGMSG               ;@@D
011170  012746  011304                  MOV     #DEBUGMSG,-(SP)
011174  012746  000001                  MOV     #1,-(SP)
011200  010600                          MOV     SP,R0
011202  104417                          TRAP    C$PNTF
011204  062706  000004                  ADD     #4,SP
011210  010037  002250          5$:     MOV     R0,RCVHIADD             ;SAVE RECV HIGH ADDRESS
011214  010137  002252                  MOV     R1,RCVLOAD              ;SAVE RECV LOW ADDRESS
011220  005737  003102                  TST     KTENABLE                ;TESTING ABOVE 28K?
011224  001403                          BEQ     10$                     ;BR IF NO
011226  004737  020112                  JSR     PC,SETMAP               ;RETURN ADDRESS BIASED TO PAR6 IN R0
011232  010001                          MOV     R0,R1                   ;GET RETURNED ADDRESS BIASED TO PAR6
011234  005004          10$:    CLR     R4                      ;WORD IN BUFFER
011236  005005                  CLR     R5                      ;CLEAR ERROR SEEN FLAG
011240  111264  002266  15$:    MOVB    (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
011244  111164  002432          MOVB    (R1),RECMSG(R4) ;SAVE RECV FOR ERROR REPORT
011250  122221                  CMPB    (R2)+,(R1)+     ;EXPD EQUAL RECV?
011252  001401                  BEQ     25$             ;BR IF YES
011254  005205                  INC     R5              ;SET ERROR SEEN FLAG
011256  062704  000001  25$:    ADD     #1,R4           ;POINT TO NEXT BYTE
011262  020403                  CMP     R4,R3           ;DONE ALL BYTES?
011264  002001                  BGE     50$             ;BR IF YES
011266  000764                  BR      15$             ;DO NEXT BYTE
011270  005705          50$:    TST     R5              ;ANY ERRORS SEEN?
011272  001402                  BEQ     55$             ;BR IF NO
```

CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

```
 2282 011274  000241                    CLC                     ;SET FAILURE
 2283 011276  000401                    BR      60$             ;
 2284 011300  000261              55$:  SEC                     ;SET SUCCESS
 2285 011302  000207              60$:  RTS     PC              ;RETURN
 2286
 2287 011304     120     122     117  DEBUGMSG:       .ASCIZ  'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED-';@@D
 2288 011374     045     116     045  FERCM:  .ASCII  /%N%A  %%%/
 2289 011405     040     040     124  ERCM:   .ASCIZ  /  TSSR ERROR CODE REC'D = /
 2290 011440     056     056     056  SIMSG:  .ASCIZ  /.... AFTER DOING SOFT INIT/
 2291 011473     124     105     123  TINERR: .ASCIZ  /TEST: .../
 2292                                  .EVEN
```

```
                                        ;+
                                        ;
                                        ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
                                        ;
                                        ;INPUT:
                                        ;
                                        ;       R1      CONTENTS OF TSSR AT ERROR
                                        ;
                                        ;SIDE EFFECTS:
                                        ;
                                        ;       EXECUTES DROP UNIT TO CEASE TESTING
                                        ;
                                        ;-

011506                                          BGNMSG  SFIMSG
011506                                  SFIMSG::
011506  004737  005264                          JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
011512  004737  017776                          JSR     PC,CKDROP       ;DROP UNIT, IF ALLOWED
011516                                          ENDMSG
011516                                  L10003:
011516  104423                                  TRAP    C$MSG

                                        ;+
                                        ;PRINT ROUTINE TO PRINT THE CONTENTS OF
                                        ;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET.
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      TSSR CONTENTS
                                        ;       R4      ADDRESS OF COMMAND PACKET
                                        ;
                                        ;-

011520                                          BGNMSG  PKTSSR
011520                                  PKTSSR::
011520  004737  005264                          JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
011524  012700  000004                          MOV     #4,R0           ;NO. OF WORDS IN PACKET
011530  004737  006712                          JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
011534  013700  002716                          MOV     MESBFA,R0       ;ADDRESS OF MESSAGE BUFFER
011540  005001                                  CLR     R1              ;ASSUME NO HIGH MEMORY
011542  004737  013702                          JSR     PC,PRMESS       ;PRINT THE MESSAGE BUFFER ALSO
011546                                          ENDMSG
011546                                  L10004:
011546  104423                                  TRAP    C$MSG

                                        ;+
                                        ;PRINT ROUTINE TO PRINT THE CONTENTS OF
                                        ;TSSR AND A GET STATUS COMMAND PACKET.
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      TSSR CONTENTS
                                        ;       R4      ADDRESS OF COMMAND PACKET
                                        ;
                                        ;-
```

```
2345
2346 011550                                    BGNMSG  PKTGETS
     011550                                PKTGETS::
2347 011550  004737  005264                       JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
2348 011554  012700  000002                       MOV     #2,R0           ;NO. OF WORDS IN GET STATUS PACKET
2349 011560  004737  006712                       JSR     PC,PRIPKT       ;PRINT THE CONTENTS OF COMMAND PACKET
2350 011564                                       ENDMSG
     011564                                L10005:
     011564  104423                              TRAP    C$MSG
2351
2352
2353                                       ;+
2354                                       ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
2355                                       ;
2356                                       ;INPUTS:
2357                                       ;
2358                                       ;       R1      TSSR CONTENTS
2359                                       ;       R4      ADDRESS OF COMMAND PACKET
2360                                       ;-
2361
2362 011566                                    BGNMSG  SFFMSG
     011566                                SFFMSG::
2363 011566  004737  005264                       JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
2364 011572                                       ENDMSG
     011572                                L10006:
     011572  104423                              TRAP    C$MSG
2365
2366
2367                                              .SBTTL  PKTMES  - PRINT TSSR AND MESSAGE BUFFER
2368                                       ;+
2369                                       ;
2370                                       ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
2371                                       ;BUFFER FOR ERROR REPORTS
2372                                       ;
2373                                       ;INPUTS:
2374                                       ;
2375                                       ;       R1      CONTENTS OF TSSR
2376                                       ;       R2      LOW ORDER MESSAGE BUFFER
2377                                       ;       R3      HIGH ORDER MESSAGE BUFFER ADDRESS
2378                                       ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
2379                                       ;-
2380 011574                                    BGNMSG  PKTMES
     011574                                PKTMES::
2381 011574  004737  005264                       JSR     PC,PRITSSR      ;PRINT CONTENTS OF TSSR
2382 011600  010200                              MOV     R2,R0           ;LOW ORDER ADDRESS
2383 011602  010301                              MOV     R3,R1           ;HIGH ORDER ADDRESS
2384 011604  004737  013702                       JSR     PC,PRMESS       ;PRINT THE MESSAGE BUFFER
2385 011610                                       ENDMSG
     011610                                L10007:
     011610  104423                              TRAP    C$MSG
2386
```

```
                                        .SBTTL  ADDSSR  - PRINT TEST ADDRESS AND TSSR
                                ;+
                                ;PRINT ROUTINE TO PRINT THE CONTENTS OF
                                ;TSSR AND A MEMORY TEST ADDRESS
                                ;
                                ;INPUTS:
                                ;
                                ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                ;       ERRHI   HIGH ORDER MEMORY TEST ADDRESS
                                ;       ERRLO   LOW ORDER MEMORY TEST ADDRESS
                                ;-
                                        BGNMSG  ADDSSR
                                ADDSSR::
                                        JSR     PC,PRITADD      ;PRINT MEMORY TEST ADDRESS
                                        MOV     TSSR(R5),R1     ;GET CURRENT TSSR
                                        JSR     PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
                                        ENDMSG
                                L10010:
                                        TRAP    C$MSG


                                        .SBTTL  MSGEXP - PRINT WRITE CHAR. EXPD RECV MESSAGE BUFFERS
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
                                        BGNMSG  MSGEXP
                                MSGEXP::
                                        MOV     #7,R0           ;ASSUME NO EXT FEATURES
                                5$:     JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
                                        ENDMSG
                                L10011:
                                        TRAP    C$MSG
```

```
                                        .SBTTL  FIFEXP  - PRINT FIFO EXP/RECV DATA
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT FIFO EXP/RECV DATA
                                ;
                                ;       R1      - BYTE COUNT
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
                                ;-
                                        BGNMSG  FIFEXP
011642                          FIFEXP::
                                        PRINTX  #FIF1MSG,R1     ;PRINT BYTES TRANSFERRED
011642  010146                          MOV     R1,-(SP)
011644  012746  011714                  MOV     #FIF1MSG,-(SP)
011650  012746  000002                  MOV     #2,-(SP)
011654  010600                          MOV     SP,R0
011656  104415                          TRAP    C$PNTX
011660  062706  000006                  ADD     #6,SP
                                        PRINTX  #FIF2MSG        ;PRINT HEADER MSG
011664  012746  011763                  MOV     #FIF2MSG,-(SP)
011670  012746  000001                  MOV     #1,-(SP)
011674  010600                          MOV     SP,R0
011676  104415                          TRAP    C$PNTX
011700  062706  000004                  ADD     #4,SP
011704  010100                          MOV     R1,R0           ;GET BYTE COUNT
011706  004737  015616                  JSR     PC,PRBYTEXP     ;PRINT FIFO BYTES IN ERROR
011712                                  ENDMSG
011712                          L10012:
011712  104423                          TRAP    C$MSG
011714    045     116     045   FIF1MSG:        .ASCIZ  '<N><A NUMBER OF BYTES TRANSFERRED = <D2'
011763    045     116     045   FIF2MSG:        .ASCIZ  '<N><A FIFO DATA BYTES IN ERROR:'
                                        .EVEN
```

```
                                        .SBTTL  MSGSTAT - PRINT STATUS HEADER AND MESSAGE BUFFERS
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
                                ;
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
012022                                  BGNMSG  MSGSTAT
012022                          MSGSTAT::
012022  012701  012064                  MOV     #STATCOD,R1     ;ASCII ADDRESS TABLE
012026  012100                  10$:    MOV     (R1)+,R0        ;DONE ALL MSG LINES?
012030  001410                          BEQ     20$             ;BR IF YES
012032                                  PRINTX  R0              ;PRINT STATUS BIT NAMES
012032  010046                          MOV     R0,-(SP)
012034  012746  000001                  MOV     #1,-(SP)
012040  010600                          MOV     SP,R0
012042  104415                          TRAP    C$PNTX
012044  062706  000004                  ADD     #4,SP
012050  000766                          BR      10$             ;DO ANOTHER MSG LINE
012052  012700  000012          20$:    MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
012056  004737  015246                  JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
012062                                  ENDMSG
012062                          L10013:
012062  104423                          TRAP    C$MSG

012064  012102  012144  012235  STATCOD:       .WORD   1$,2$,3$,4$,5$,6$,0
012102     045     116     045  1$:.ASCIZ  '%N%A Tape Bus Signals in Word #8:'
012144     045     116     045  2$:.ASCIZ  '%N%A        PARERR<15>  IEOT  <12>  IFMK  <9>  IRDY<6>  IRWD<2>'
012235     045     116     045  3$:.ASCIZ  '%N%A        IRESV2<14>  IIDENT<11>  IHER  <8>  IONL<5>  IFBY<1>'
012326     045     116     045  4$:.ASCIZ  '%N%A        IRESV1<13>  ICER  <10>  ISPEED<7>  ILDP<4>  IFPT<0>'
012417     045     116     045  5$:.ASCIZ  '%N%A Tape Bus Signals in Word #9:'
012461     045     116     045  6$:.ASCIZ  '%N%A        DATMIS<7>  ILW<6>  OUTRDY<5>  INRDY<4>'
                                        .EVEN


                                        .SBTTL  MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
012536                                  BGNMSG  MSGLOOP
012536                          MSGLOOP::
012536  012701  012600                  MOV     #LOOPCOD,R1     ;ASCII ADDRESS TABLE
```

MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS

```
2497 012542  012100                  10$:   MOV     (R1)+,R0        ;DONE ALL MSG LINES?
2498 012544  001410                         BEQ     20$             ;BR IF YES
2499 012546                                 PRINTX  R0              ;PRINT STATUS BIT NAMES
     012546  010046                         MOV     R0,-(SP)
     012550  012746  000001                 MOV     #1,-(SP)
     012554  010600                         MOV     SP,R0
     012556  104415                         TRAP    C$PNTX
     012560  062706  000004                 ADD     #4,SP
2500 012564  000766                         BR      10$             ;DO ANOTHER MSG LINE
2501 012566  012700  000012           20$:   MOV     #10.,R0         ;NUMBER OF WORDS IN A READ STATUS BUFFER
2502 012572  004737  015246                 JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
2503 012576                                 ENDMSG
     012576                           L10014:
     012576  104423                         TRAP    C$MSG
2504
2505 012600  012620  012673  012772  LOOPCOD:       .WORD   1$,2$,3$,4$,5$,6$,7$,0
2506 012620    045     116     045  1$:.ASCIZ  '%N%A Tape Bus Loopback Signals in Word #8:'
2507 012673    045     116     045  2$:.ASCIZ  '%N%A      PARERR<15>          IRESV2<14>              IRESV1<13>'
2508 012772    045     116     045  3$:.ASCIZ  '%N%A IHISP=>IEOT<12>    IWRT=>IIDENT<11>    IREV  =>ICER  <10>'
2509 013071    045     116     045  4$:.ASCIZ  '%N%A IWFM =>IFMK<09>    IEDIT=>IHER <08>    IFAD  =>ISPEED<07>'
2510 013170    045     116     045  5$:.ASCIZ  '%N%A ITADO=>IRDY<06>    ITAD1=>IONL <05>    IERASE=>ILDP  <04>'
2511 013267    045     116     045  6$:.ASCIZ  '%N%A IREW =>IDBY<03>    IRWU =>IRWD <02>    IFEN  =>IFBY  <01>'
2512 013366    045     116     045  7$:.ASCIZ  '%N%A IGO  =>IFPT<00>'
2513                                        .EVEN
2514
```

```
                                        .SBTTL  MSGSUB - PRINT WRITE SUBSYSTEM MESSAGE BUFFER
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
                                ;
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
                                        BGNMSG  MSGSUB
013414                          MSGSUB::
013414  012700  000012                  MOV     #10.,R0         ;SIZE OF WRITE SUBSYSTEM BUFFER
013420  004737  015246                  JSR     PC,PRMSGEXP     ;PRINT EXPD/RECV MESSAGE BUFFERS
013424                                  ENDMSG
013424                          L10015:
013424  104423                          TRAP    C$MSG
                                        .SBTTL  MEMADD - PRINT MEMORY ADDRESS DATA ERROR
                                ;+
                                ;
                                ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       ERRHI   - MEMORY ERROR HIGH ORDER ADDRESS
                                ;       ERRLO   - MEMORY ERROR LOW ORDER ADDRESS
                                ;       EXP     - EXPECTED DATA
                                ;       RECV    - RECEIVED DATA
                                ;-
                                        BGNMSG  MEMADD
013426                          MEMADD::
013426  004737  007532                  JSR     PC,PRIADD       ;PRINT MEMORY ADDRESS IN ERROR
013432  013701  002176                  MOV     EXPD,R1         ;GET EXPD DATA
013436  013702  002200                  MOV     RECV,R2         ;GET RECEIVED DATA
013442  004737  007314                  JSR     PC,PRIXOR       ;PRINT EXPD/RECV
013446                                  ENDMSG
013446                          L10016:
013446  104423                          TRAP    C$MSG
```

PRAMPKT - PRINT RAM AND PACKET DATA

```
                                        .SBTTL  PRAMPKT - PRINT RAM AND PACKET DATA
                                ;+
                                ;
                                ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
                                ;WHEN THE RAM DATA DOES NOT MATCH.
                                ;
                                ;INPUTS:
                                ;
                                ;       R4      POINTER TO COMMAND PACKET
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       RAMDATA         DATA AS READ FROM THE RAM
                                ;       RAMSIZ          NUMBER OF BYTES IN PACKET
                                ;                       IF RAMSIZ=0 THEN DEFAULT TO 8.
                                ;
                                ;IMPLICIT OUTPUTS:
                                ;
                                ;       RAMSIZ  SET TO 0
                                ;-
                                PRAMPKT:
                                        SAVREG                          ;SAVE R1-R5 UNTIL NEXT RETURN
013454  012701  002206                  MOV     #RAMDATA,R1             ;DATA FROM THE RAM
013460  005002                          CLR     R2                      ;INIT BYTE NUMBER
013462  122124                  5$:     CMPB    (R1)+,(R4)+             ;COMPARE EXPECTED, RECEIVED
013464  001000                          BNE     7$                      ;BR IF NO MATCH
013466  116105  177777          7$:     MOVB    -1(R1),R5               ;GET RECV RAM DATA
013472  116403  177777                  MOVB    -1(R4),R3               ;GET EXPD PACKET DATA
013476                                  XOR     R5,R3                   ;XOR EXPD/RECV
013506  042703  177400                  BIC     #177400,R3              ;LOW BYTE ONLY
013512  116137  177777  002200          MOVB    -1(R1),RECV             ;GET RECEIVED RAM DATA
013520  116437  177777  002176          MOVB    -1(R4),EXPD             ;GET EXPECTED RAM DATA
013526                                  PRINTB  #RAMASC,R2,RECV,EXPD,R3
013526  010346                          MOV     R3,-(SP)
013530  013746  002176                  MOV     EXPD,-(SP)
013534  013746  002200                  MOV     RECV,-(SP)
013540  010246                          MOV     R2,-(SP)
013542  012746  013616                  MOV     #RAMASC,-(SP)
013546  012746  000005                  MOV     #5,-(SP)
013552  010600                          MOV     SP,R0
013554  104414                          TRAP    C$PNTB
013556  062706  000014                  ADD     #14,SP
013562  005202                  10$:    INC     R2                      ;UPDATE BYTE COUNT
013564  005737  002246                  TST     RAMSIZ                  ;DEFAULT TO 8.?
013570  001404                          BEQ     15$                     ;BR IF YES
013572  020237  002246                  CMP     R2,RAMSIZ               ;DONE ALL BYTES?
013576  003731                          BLE     5$                      ;BR IF NO
013600  000403                          BR      25$                     ;
013602  020227  000010          15$:    CMP     R2,#8.                  ;DONE DEFAULT NUMBER OF BYTES?
013606  002725                  20$:    BLT     5$                      ;BR IF NO
013610  005037  002246          25$:    CLR     RAMSIZ                  ;SET DEFAULT RAMSIZ
013614  000207                          RTS     PC                      ;RETURN

013616  045  116  045  RAMASC: .ASCIZ  '%N%A  BYTE: %D2%A RAM: %03%A Packet: %03%A XOR:%03'
                                .EVEN
```

```
                                        .SBTTL  PRMESS  - PRINT CONTENTS OF MESSAGE BUFFER
                                ;+
                                ;
                                ;THIS ROUTINE PRINTS THE CONTENTS OF
                                ;THE 7 WORD MESSAGE BUFFER RETURNED BY THE
                                ;TK-25.
                                ;
                                ;INPUT:
                                ;
                                ;       R0      LOW ORDER ADDRESS OF MESSAGE BUFFER
                                ;       R1      HIGH ORDER ADDRESS OF MESSAGE BUFFER
                                ;       NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
                                ;
                                ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
                                ;
                                ;-

013702                          PRMESS:
013702                                  SAVREG                          ;SAVE THE REGISTERS
013706  010537  010642                  MOV     R5,RAMR5H               ;SAVE DEVICE REGISTER POINTER
013712  010005                          MOV     R0,R5                   ;SAVE LOW ORDER ADDRESS
013714  005737  003102                  TST     KTENABLE                ;ADDRESS ABOVE 28K?
013720  001001                          BNE     10$                     ;BR IF YES
013722  005001                          CLR     R1                      ;SET HIGH ORDER ADDRESS TO 0
013724  010103                  10$:    MOV     R1,R3                   ;SAVE HIGH ORDER ADDRESS
013726  006100                          ROL     R0                      ;SHIFT BIT15 TO C BIT
013730  006101                          ROL     R1                      ;SHIFT TO HIGH ORDER FOR PRINTOUT
013732                                  PRINTX  #PROASC,R1,R5           ;PRINT MESSAGE BUFFER ADDRESS
013732  010546                          MOV     R5,-(SP)
013734  010146                          MOV     R1,-(SP)
013736  012746  014540                  MOV     #PROASC,-(SP)
013742  012746  000003                  MOV     #3,-(SP)
013746  010600                          MOV     SP,R0
013750  104415                          TRAP    C$PNTX
013752  062706  000010                  ADD     #10,SP
013756  022715  177777                  CMP     #177777,(R5)            ;MESSAGE BUFFER FULL OF ONES
013762  001010                          BNE     15$                     ;BR IF BUFFER IS PROBABLY OKAY
013764                                  PRINTX  #MESBFN                 ;"MESSAGE BUFFER PROBABLY NOT VALID"
013764  012746  014460                  MOV     #MESBFN,-(SP)
013770  012746  000001                  MOV     #1,-(SP)
013774  010600                          MOV     SP,R0
013776  104415                          TRAP    C$PNTX
014000  062706  000004                  ADD     #4,SP
014004                          15$:    PRINTX  #PR1ASC                 ;PRINT HEADER FOR CONTENTS
014004  012746  014605                  MOV     #PR1ASC,-(SP)
014010  012746  000001                  MOV     #1,-(SP)
014014  010600                          MOV     SP,R0
014016  104415                          TRAP    C$PNTX
014020  062706  000004                  ADD     #4,SP
014024  005004                          CLR     R4                      ;NUMBER OF THE NEXT WORD
014026  010501                          MOV     R5,R1                   ;COPY LOW ORDER ADDRESS
014030  010300                          MOV     R3,R0                   ;COPY HIGH ORDER ADDRESS
014032  001403                          BEQ     20$                     ;BR IF NOT ABOVE 28K
014034  004737  020112                  JSR     PC,SETMAP               ;SETUP PAR ADDRESS IN R0
014040  010005                          MOV     R0,R5                   ;GET PAR FORMAT ADDRESS ABOVE 28K
014042                          20$:
014042                                  PRINTX  #MESHEA,(R5)+           ;PRINT "MESSAGE BUFFER HEADER ="
```

PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

```
        014042  012546                          MOV     (R5)+,-(SP)
        014044  012746  014643                  MOV     #MESHEA,-(SP)
        014050  012746  000002                  MOV     #2,-(SP)
        014054  010600                          MOV     SP,R0
        014056  104415                          TRAP    C$PNTX
        014060  062706  000006                  ADD     #6,SP
  2646  014064                                  PRINTX  #DATAFL,(R5)+   ;PRINT "DATA FIELD LENGTH     ="
        014064  012546                          MOV     (R5)+,-(SP)
        014066  012746  014710                  MOV     #DATAFL,-(SP)
        014072  012746  000002                  MOV     #2,-(SP)
        014076  010600                          MOV     SP,R0
        014100  104415                          TRAP    C$PNTX
        014102  062706  000006                  ADD     #6,SP
  2647  014106                                  PRINTX  #RBPCRA,(R5)+   ;PRINT "RESIDUAL BYTE COUNTER ="
        014106  012546                          MOV     (R5)+,-(SP)
        014110  012746  014755                  MOV     #RBPCRA,-(SP)
        014114  012746  000002                  MOV     #2,-(SP)
        014120  010600                          MOV     SP,R0
        014122  104415                          TRAP    C$PNTX
        014124  062706  000006                  ADD     #6,SP
  2648  014130                                  PRINTX  #XSOCON,(R5)+   ;PRINT "XSTAT0 CONTENTS       ="
        014130  012546                          MOV     (R5)+,-(SP)
        014132  012746  015022                  MOV     #XSOCON,-(SP)
        014136  012746  000002                  MOV     #2,-(SP)
        014142  010600                          MOV     SP,R0
        014144  104415                          TRAP    C$PNTX
        014146  062706  000006                  ADD     #6,SP
  2649  014152                                  PRINTX  #XS1CON,(R5)+   ;PRINT "XSTAT1 CONTENTS       ="
        014152  012546                          MOV     (R5)+,-(SP)
        014154  012746  015067                  MOV     #XS1CON,-(SP)
        014160  012746  000002                  MOV     #2,-(SP)
        014164  010600                          MOV     SP,R0
        014166  104415                          TRAP    C$PNTX
        014170  062706  000006                  ADD     #6,SP
  2650  014174                                  PRINTX  #XS2CON,(R5)+   ;PRINT "XSTAT2 CONTENTS       ="
        014174  012546                          MOV     (R5)+,-(SP)
        014176  012746  015134                  MOV     #XS2CON,-(SP)
        014202  012746  000002                  MOV     #2,-(SP)
        014206  010600                          MOV     SP,R0
        014210  104415                          TRAP    C$PNTX
        014212  062706  000006                  ADD     #6,SP
  2651  014216                                  PRINTX  #XS3CON,(R5)+   ;PRINT "XSTAT3 CONTENTS       ="
        014216  012546                          MOV     (R5)+,-(SP)
        014220  012746  015201                  MOV     #XS3CON,-(SP)
        014224  012746  000002                  MOV     #2,-(SP)
        014230  010600                          MOV     SP,R0
        014232  104415                          TRAP    C$PNTX
        014234  062706  000006                  ADD     #6,SP
  2652  014240  022737  000001  002134          CMP     #1,TRANSTST     ;CHECK SOFTWARE P-TABLE
  2653  014246  001402                          BEQ     40$             ;DO DUMP
  2654  014250  000137  014360                  JMP     50$             ;DON'T DO THE DUMP
  2655  014254                          40$:    PRINTX  #RAMFHR
        014254  012746  014362                  MOV     #RAMFHR,-(SP)
        014260  012746  000001                  MOV     #1,-(SP)
        014264  010600                          MOV     SP,R0
        014266  104415                          TRAP    C$PNTX
        014270  062706  000004                  ADD     #4,SP
```

PRMESS - PRINT CONTENTS OF MESSAGE BUFFER

```
 2656 014274  012737  000010  002246          MOV     #8.,RAMSIZ      ;RAM FIELD IS 8 BYTES LONG
 2657 014302  012737  000020  010640          MOV     #20,RAMHLD      ;FIELD STARTS AT 20 OCTAL (10 HEX)
 2658 014310  004737  010456                  JSR     PC,RAMER        ;READ AND PRINT THEM
 2659 014314  012737  000040  010640          MOV     #40,RAMHLD      ;FIELD STARTS AT 40 OCTAL (20 HEX)
 2660 014322  004737  010456                  JSR     PC,RAMER        ;READ AND PRINT THEM
 2661 014326  012737  000060  010640          MOV     #60,RAMHLD      ;FIELD STARTS AT 60 OCTAL (30 HEX)
 2662 014334  004737  010456                  JSR     PC,RAMER        ;READ AND PRINT THEM
 2663 014340  012737  000020  002246          MOV     #16.,RAMSIZ     ;RAM FIELD IS SIXTEEN BYTES LONG
 2664 014346  012737  000100  010640          MOV     #100,RAMHLD     ;FIELD STARTS AT 100 OCTAL (40 HEX)
 2665 014354  004737  010456                  JSR     PC,RAMER        ;READ AND PRINT THEM
 2666 014360  000207                  50$:    RTS     PC              ;RETURN
 2667 014362     045     116     045  RAMFHR: .ASCIZ  '<N><A ********** SPECIAL CONTROLLER RAM MEMORY DUMP **********'
 2668 014460     045     116     045  MESBFN: .ASCIZ  '<N><A MESSAGE BUFFER CONTENTS PROBABLY NOT VALID'
 2669 014540     045     116     045  PROASC: .ASCIZ  '<N><A Message Buffer Address = <01><05>'
 2670 014605     045     116     045  PR1ASC: .ASCIZ  '<N><A Message Buffer Contents:'
 2671
 2672 014643     045     116     045  MESHEA: .ASCIZ  '<N><A Message Buffer Header     = <06>'
 2673 014710     045     116     045  DATAFL: .ASCIZ  '<N><A Data Field Length         = <06>'
 2674 014755     045     116     045  RBPCRA: .ASCIZ  '<N><A Residual Byte Counter     = <06>'
 2675 015022     045     116     045  XS0CON: .ASCIZ  '<N><A XSTAT0 Contents           = <06>'
 2676 015067     045     116     045  XS1CON: .ASCIZ  '<N><A XSTAT1 Contents           = <06>'
 2677 015134     045     116     045  XS2CON: .ASCIZ  '<N><A XSTAT2 Contents           = <06>'
 2678 015201     045     116     045  XS3CON: .ASCIZ  '<N><A XSTAT3 Contents           = <06>'
 2679                                  .EVEN
```

```
                                        .SBTTL  PRMSGEXP - PRINT EXPD/RECV MESSAGE BUFFERS
                                ;+[B
                                ;
                                ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
                                ;
                                ;       R0      - NUMBER OF WORDS IN BUFFER
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;       RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
                                ;       RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
                                ;-
015246                          PRMSGEXP::
015246                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
015252  010005                          MOV     R0,R5           ;SAVE NUMBER OF WORDS
015254  013700  002252                  MOV     RCVLOADD,R0     ;GET RECV LOW ADDRESS
015260  010004                          MOV     R0,R4           ;COPY LOW ADDRESS
015262  013701  002250                  MOV     RCVHIADD,R1     ;GET RECV HIGH ADDRESS
015266  006100                          ROL     R0              ;SHIFT BIT15 TO C BIT
015270  006101                          ROL     R1              ;SHIFT TO HIGH ORDER FOR PRINTOUT
015272                                  PRINTX  #PRMSG0,R1,R4   ;PRINT MESSAGE BUFFER ADDRESS
015272  010446                          MOV     R4,-(SP)
015274  010146                          MOV     R1,-(SP)
015276  012746  015426                  MOV     #PRMSG0,-(SP)
015302  012746  000003                  MOV     #3,-(SP)
015306  010600                          MOV     SP,R0
015310  104415                          TRAP    C$PNTX
015312  062706  000010                  ADD     #10,SP
015316                                  PRINTX  #PRMSG1         ;PRINT HEADER FOR CONTENTS
015316  012746  015473                  MOV     #PRMSG1,-(SP)
015322  012746  000001                  MOV     #1,-(SP)
015326  010600                          MOV     SP,R0
015330  104415                          TRAP    C$PNTX
015332  062706  000004                  ADD     #4,SP
015336  005004                          CLR     R4              ;NUMBER OF THE CURRENT WORD
015340  012701  002266                  MOV     #EXPMSG,R1      ;GET EXPD BUFFER ADDRESS
015344  012702  002432                  MOV     #RECMSG,R2      ;GET RECV BUFFER ADDRESS
015350  011100                  20$:    MOV     (R1),R0         ;GET EXPD
015352  011203                          MOV     (R2),R3         ;GET RECV
015354                                  XOR     R0,R3           ;XOR EXPD/RECV
015364                                  PRINTX  #PRMSG2,R4,(R1)+,(R2)+,R3
015364  010346                          MOV     R3,-(SP)
015366  012246                          MOV     (R2)+,-(SP)
015370  012146                          MOV     (R1)+,-(SP)
015372  010446                          MOV     R4,-(SP)
015374  012746  015531                  MOV     #PRMSG2,-(SP)
015400  012746  000005                  MOV     #5,-(SP)
015404  010600                          MOV     SP,R0
015406  104415                          TRAP    C$PNTX
015410  062706  000014                  ADD     #14,SP
015414  005204                          INC     R4              ;NUMBER OF THE NEXT
015416  020405                          CMP     R4,R5           ;DONE ALL YET?
015420  002001                          BGE     50$             ;BR IF YES
015422  000752                          BR      20$             ;DO ANOTHER
015424  000207                  50$:    RTS     PC              ;RETURN
```

```
 2717
 2718 015426     045     116     045  PRMSG0: .ASCIZ  '%N%A Message Buffer Address = %01%05'
 2719 015473     045     116     045  PRMSG1: .ASCIZ  '%N%A Message Buffer Contents:'
 2720 015531     045     116     045  PRMSG2: .ASCIZ  '%N%A   WORD 0%D2%A  EXPD: %06%A  RECV: %06%A  XOR: %06'
 2721                                         .EVEN
 2722
```

```
                                        .SBTTL  PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
                                ;+
                                ;
                                ;ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS
                                ;       ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
                                ;
                                ;       R0      - NUMBER OF BYTES IN BUFFER
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       EXPMSG  - EXPECTED MESSAGE BUFFER
                                ;       RECMSG  - RECEIVED MESSAGE BUFFER
                                ;-
015616                          PRBYTEXP::
015616                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
015622  010005                          MOV     R0,R5           ;SAVE NUMBER OF BYTES
015624  005037  002264                  CLR     PRMNO           ;INIT ERROR COUNT
015630  005004                          CLR     R4              ;NUMBER OF THE CURRENT BYTE
015632  012701  002266                  MOV     #EXPMSG,R1      ;GET EXPD BUFFER ADDRESS
015636  012702  002432                  MOV     #RECMSG,R2      ;GET RECV BUFFER ADDRESS
015642  111100                  20$:    MOVB    (R1),R0         ;GET EXPD BYTE
015644  042700  177400                  BIC     #^C<377>,R0     ;CLEAR UPPER BYTE
015650  110037  016164                  MOVB    R0,PRBEXP       ;SAVE FOR ERROR REPORT
015654  111203                          MOVB    (R2),R3         ;GET RECV BYTE
015656  042703  177400                  BIC     #^C<377>,R3     ;CLEAR UPPER BYTE
015662  110337  016166                  MOVB    R3,PRBREC       ;FOR ERROR REPORT
015666                                  XOR     R0,R3           ;XOR EXPD/RECV
015676  122122                          CMPB    (R1)+,(R2)+     ;EXPD = RECV?
015700  001431                          BEQ     30$             ;BR IF YES
015702  005237  002264                  INC     PRMNO           ;UPDATE ERROR COUNT
015706  023727  002264  000010          CMP     PRMNO,#8.       ;PRINTED 8?
015714  101023                          BHI     30$             ;BR IF YES
015716                          27$:    PRINTX  #PRBMSG,R4,PRBEXP,PRBREC,R3
015716  010346                          MOV     R3,-(SP)
015720  013746  016166                  MOV     PRBREC,-(SP)
015724  013746  016164                  MOV     PRBEXP,-(SP)
015730  010446                          MOV     R4,-(SP)
015732  012746  016032                  MOV     #PRBMSG,-(SP)
015736  012746  000005                  MOV     #5,-(SP)
015742  010600                          MOV     SP,R0
015744  104415                          TRAP    C$PNTX
015746  062706  000014                  ADD     #14,SP
015752                                  FORCEXIT        50$     ;@@D
015762  000404                          BR      35$             ;@@D
015764                          30$:
015764                                  FORCERROR  27$,NOTSSR   ;@@D
015774                          35$:                            ;@@D
015774  005204                          INC     R4              ;NUMBER OF THE NEXT
015776  020405                          CMP     R4,R5           ;DONE ALL YET?
016000  002001                          BGE     50$             ;BR IF YES
016002  000717                          BR      20$             ;DO ANOTHER
016004                          50$:    PRINTX  #PRBTOT,PRMNO   ;PRINT TOTAL ERROR COUNT
016004  013746  002264                  MOV     PRMNO,-(SP)
016010  012746  016117                  MOV     #PRBTOT,-(SP)
016014  012746  000002                  MOV     #2,-(SP)
016020  010600                          MOV     SP,R0
016022  104415                          TRAP    C$PNTX
```

```
      016024  062706  000006                  ADD     #6,SP
 2767 016030  000207                          RTS     PC              ;RETURN
 2768
 2769 016032     045     116     045  PRBMSG: .ASCIZ  '%N%A   BYTE #%D2%A  EXPD: %03%A  RECV: %03%A  XOR: %03'
 2770 016117     045     116     045  PRBTOT: .ASCIZ  '%N%A NUMBER OF BYTES IN ERROR = %D2'
 2771                                          .EVEN
 2772 016164  000000                  PRBEXP: .WORD   0               ;EXPD
 2773 016166  000000                  PRBREC: .WORD   0               ;RECV
 2774
```

```
                                        .SBTTL  EXPREC  - PRINT EXPD/RECV WORD DATA
                                ;+
                                ;
                                ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      RECEIVED DATA
                                ;       R2      EXPECTED DATA
                                ;
                                ;-

016170                                  BGNMSG  EXPREC
016170                          EXPREC::
016170  004737  007314                  JSR     PC,PRIXOR       ;PRINT THE DATA
016174                                  ENDMSG
016174                          L10017:
016174  104423                          TRAP    C$MSG
```

```
                                        .SBTTL  EXPBREC - PRINT EXPD/RECV BYTE DATA
                                ;+
                                ;
                                ;PRINT ROUTINE TO DISPLAY BYTE EXPD/RECV DATA
                                ;
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      RECEIVED DATA BYTE
                                ;       R2      EXPECTED DATA BYTE
                                ;
                                ;-

016176                                  BGNMSG  EXPBREC
016176                          EXPBREC::
016176  004737  007164                  JSR     PC,PRIBXOR      ;PRINT THE DATA
016202                                  ENDMSG
016202                          L10020:
016202  104423                          TRAP    C$MSG


                                        .SBTTL  RAMERR  - PRINT RAM AND PACKET DATA
                                ;+
                                ;
                                ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
                                ;
                                ;INPUTS:
                                ;
                                ;       R4      POINTER TO COMMAND PACKET
                                ;
                                ;IMPLICIT INPUTS:
                                ;
                                ;       RAMDATA         DATA AS READ FROM THE RAM
                                ;       RAMSIZ          NUMBER OF BYTES IN PACKET
                                ;                       IF RAMSIZ=0 THEN DEFAULT TO 8.
                                ;
                                ;IMPLICIT OUTPUTS:
                                ;
                                ;       RAMSIZ  SET TO 0
                                ;-

016204                                  BGNMSG  RAMERR
016204                          RAMERR::
016204  004737  013450                  JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
016210                                  ENDMSG
016210                          L10021:
016210  104423                          TRAP    C$MSG


                                        .SBTTL  RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA
                                ;+
                                ;
                                ;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
                                ;
                                ;INPUTS:
```

```
RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA

                                        ;
                                        ;       R4      POINTER TO COMMAND PACKET
                                        ;
                                        ;IMPLICIT INPUTS:
                                        ;
                                        ;       RAMDATA         DATA AS READ FROM THE RAM
                                        ;       RAMSIZ          NUMBER OF BYTES IN PACKET
                                        ;                       IF RAMSIZ=0 THEN DEFAULT TO 8.
                                        ;       ERRHI           HIGH ORDER TEST ADDRESS
                                        ;       ERRLO           LOW ORDER TEST ADDRESS
                                        ;
                                        ;IMPLICIT OUTPUTS:
                                        ;
                                        ;       RAMSIZ  SET TO 0
                                        ;-
                                        
016212                                  BGNMSG  RAMTADD
016212                                  RAMTADD::
016212  004737  007646                  JSR     PC,PRITADD      ;PRINT TEST ADDRESS
016216  004737  013450                  JSR     PC,PRAMPKT      ;PRINT RAM/PACKET DATA
016222                                  ENDMSG
016222                                  L10022:
016222  104423                          TRAP    C$MSG

                                        .SBTTL  RAMEXP  - PRINT RAM EXPD/RECV DATA
                                        ;+
                                        ;
                                        ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      RECEIVED DATA
                                        ;       R2      EXPECTED DATA
                                        ;       R4      CONTROLLER RAM ADDRESS
                                        ;-

016224                                  BGNMSG  RAMEXP
016224                                  RAMEXP::
016224  042701  177400                  BIC     #^C<377>,R1     ;SAVE EXPD RAM DATA BYTE
016230  042702  177400                  BIC     #^C<377>,R2     ;SAVE EXPD RAM DATA BYTE
016234  004737  007440                  JSR     PC,PRIRAM       ;PRINT THE RAM ADDRESS
016240  004737  007314                  JSR     PC,PRIXOR       ;PRINT THE DATA
016244                                  ENDMSG
016244                                  L10023:
016244  104423                          TRAP    C$MSG

                                        .SBTTL  TIMEXP  - PRINT TIMER A,B AND EXP/REC
                                        ;+
                                        ;
                                        ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
                                        ;AND TIMER A,B HEADER MESSAGE
                                        ;
                                        ;INPUTS:
                                        ;
                                        ;       R1      RECEIVED DATA
                                        ;       R2      EXPECTED DATA
```

```
                                        ;-
                                        
                                                BGNMSG  TIMEXP
016246                                  TIMEXP::
                                                PRINTX  #TIMSG0         ;PRINT HEADER
016246  012746  016274                          MOV     #TIMSG0,-(SP)
016252  012746  000001                          MOV     #1,-(SP)
016256  010600                                  MOV     SP,R0
016260  104415                                  TRAP    C$PNTX
016262  062706  000004                          ADD     #4,SP
016266  004737  007314                          JSR     PC,PRIXOR       ;PRINT THE DATA
016272                                          ENDMSG
016272                                  L10024:
016272  104423                                  TRAP    C$MSG

016274     045     116     045  TIMSG0: .ASCIZ  '<N><A TIMER A STATUS IS IN BIT 3<N><A TIMER B STATUS IS IN BIT 2'
                                        .EVEN
```

```
                                        .SBTTL  BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS

                                ;+
                                ;
                                ;PRINT ROUTINE FOR TSSR ERRORS ON DATA TRANSFERS
                                ;
                                ;INPUTS:
                                ;
                                ;       R1      CONTENTS OF TSSR
                                ;       R2      DATA WRITTEN (8 BITS)
                                ;
                                ;-

                                        BGNMSG  BADSSR
016374                          BADSSR::
016374  010246                          MOV     R2,-(SP)        ;SAVE DATA TRANSFERRED
016376  042702  177400                  BIC     #177400,R2      ;GET JUST ONE BYTE
016402                                  PRINTB  #XFERASC,R2
016402  010246                          MOV     R2,-(SP)
016404  012746  016434                  MOV     #XFERASC,-(SP)
016410  012746  000002                  MOV     #2,-(SP)
016414  010600                          MOV     SP,R0
016416  104414                          TRAP    C$PNTB
016420  062706  000006                  ADD     #6,SP
016424  012602                          MOV     (SP)+,R2        ;RESTORE R2
016426  004737  005264                  JSR     PC,PRITSSR      ;DECODE TSSR CONTENTS
016432                                  ENDMSG
016432                          L10025:
016432  104423                          TRAP    C$MSG
016434    045     116     045   XFERASC:        .ASCIZ  '%N%A Data Transferred = %03'
```

```
                                        .SBTTL  GLOBAL SUBROUTINES SECTION
                                ;++
                                ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
                                ; THAT ARE USED IN MORE THAN ONE TEST.
                                ;--
                                        .SBTTL  SOFINIT - SOFT INITIALIZE OF CONTROLLER

                                ;+
                                ;
                                ;ROUTINE TO DO A SOFT INITIALIZE OF THE CONTROLLER
                                ;BY WRITING INTO THE TSSR REGISTER. AFTER THE INIT,
                                ;THE TSSR REGISTER IS TESTED FOR ERRORS. ANY ERRORS
                                ;DETECTED SHOULD BE TREATED AS DEVICE FATAL ERRORS.
                                ;
                                ;INPUTS:
                                ;
                                ;       R5      ADDRESS OF FIRST REGISTER
                                ;
                                ;OUTPUTS:
                                ;
                                ;       R0      CONTENTS OF TSSR, IF ERROR
                                ;       CARRY   SET IF INIT WAS OKAY
                                ;               CLEAR IF FATAL ERROR
                                ;
                                ;CALLING SEQUENCE:
                                ;
                                ;       MOV     #ADDRESS,R5
                                ;       JSR     PC,SOFINIT
                                ;       BCS     CONTINUE
                                ;       ERRDF                   ;REPORT FATAL ERROR
                                ;
                                ;-

016470                          SOFINIT::
016470                                  SAVREG                  ; SAVE THE REGISTERS
016474  012765  000000  000000          MOV     #0,TSSR(R5)     ; DO THE INIT.
016502  004737  016744                  JSR     PC,WAITF        ; WAIT FOR SSR
016506  016500  000000                  MOV     TSSR(R5),R0     ;GET THE TSSR REGISTER
016512  010004                          MOV     R0,R4           ;START SETUP OF EXPECTED TSSR
016514  042704  176277                  BIC     #^C<HIADDR!OFL>,R4 ;CLEAR OUT UNUSED BITS
016520  052704  002200                  BIS     #SSR!NBA,R4     ;R4 HAS EXPECTED CONTENTS
016524  020400                          CMP     R4,R0           ;ONLY EXPECTED BITS SET ?
016526  001402                          BEQ     5$              ;BRANCH IF OKAY
016530  000241                          CLC                     ;CLEAR THE CARRY FOR ERROR
016532  000401                          BR      10$             ;GO TO EXIT
016534  000261                  5$:     SEC                     ;SET THE CARRY BIT
016536  000207                  10$:    RTS     PC              ;RETURN TO CALLER
```

```
                                        .SBTTL  CHKAMB  - CHECK TSSR FOR AMBIGUITY

                                ;+
                                ;
                                ;THIS ROUTINE TESTS THE CONTENTS OF THE TSSR REGISTER
                                ;FOR AMBIGUITY
                                ;
                                ;INPUT:
                                ;
                                ;       R0      CONTENTS OF TSSR
                                ;
                                ;OUTPUT:
                                ;
                                ;       R0      CONTENTS OF TSSR
                                ;
                                ;       CARRY   SET - NO AMBIGUITY
                                ;               CLR - AMBIGUOUS CONTENTS
                                ;
                                ;-

016540                          CHKAMB:
016540                                  SAVREG                          ;SAVE THE GENERAL REGISTERS
016544  010004                          MOV     R0,R4                   ;CONTENTS OF TSSR
016546  032700  100000                  BIT     #SC,R0                  ;IS BIT 15 SET ?
016552  001004                          BNE     5$                      ;BRANCH IF YES
016554  032700  174077                  BIT     #^C<NBA!OFL!SSR!HIADDR>,R0      ;ANY OTHER BITS SET ?
016560  001023                          BNE     40$                     ;MUST BE AN ERROR
016562  000424                          BR      45$                     ;RETURN WITH SUCCESS
016564  032700  000200          5$:     BIT     #SSR,R0                 ;IS READY BIT SET ?
016570  001011                          BNE     10$                     ;BRANCH IF READY BIT IS SET.
016572  032700  000040                  BIT     #BIT5,R0                ;IS FATAL ERROR BIT SET ?
016576  001414                          BEQ     40$                     ;ERROR IF NOT
016600  042704  177761                  BIC     #^CTERCLS,R4            ;CLEAR ALL BUT TERMINATION CODE
016604  020427  000016                  CMP     R4,#16                  ;ALL THREE BITS MUST BE SET
016610  001007                          BNE     40$                     ;ERROR IF NOT SET
016612  000410                          BR      45$                     ;OK IF ALL ARE SET
016614  032700  000040          10$:    BIT     #BIT5,R0                ;IS FATAL ERROR BIT SET ?
016620  001405                          BEQ     45$                     ;ERROR IF BIT IS SET WITH SSR
016622  032700  000006                  BIT     #BIT2!BIT1,R0           ;IS THIS A FUNCTION REJECT
016626  001002                          BNE     45$                     ;BR, IF TSSR IS OK
016630  000241                  40$:    CLC                             ;AMBIGUOUS CONTENTS
016632  000401                          BR      50$
016634  000261                  45$:    SEC                             ;SHOW SUCCESS - NO AMBIGUITY
016636  000207                  50$:    RTS     PC                      ;RETURN TO CALLER
```

```
3024                                            .SBTTL  ENAINT,DSBINT - ENABLE/DISABLE INTERRUPTS
3025                                    ;
3026                                    ; DEFAULT DISPLAY INTERRUPT HANDLERS.
3027                                    ; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
3028                                    ; OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
3029                                    ;
3030
3031                                    ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
3032                                    ;
3033          000200                            IOKCKIN=BIT7            ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
3034          000001                            IOKSTP=BIT0             ; EXPECT "STOP" INTERRUPT.
3035                                    ;
3036                                    ;INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
3037 016640      000                   INTMASK:        .BYTE   0
3038                                    ;INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
3039 016641      000                   INTFLAG:        .BYTE   0
3040
3041                                    ;SAVED INTERRUPT VECTOR:
3042 016642  000000                    INTVEC: .WORD   0
3043                                    ;SAVE CPU PC
3044 016644  000000                    INTCPC: .WORD   0
3045
3046                                    ;SUBROUTINE TO ENABLE INTERRUPTS:
3047 016646  010046                    ENAINT: MOV     R0,-(SP)        ;SAVE R0
3048 016650  013700  002156                    MOV     IVEC,R0         ;GET POINTER TO VECTORS
3049 016654  012720  016712                    MOV     #INTR,(R0)+     ;SET UP INTERRUPT VECTOR
3050 016660  012720  000340                    MOV     #PRI07,(R0)+
3051 016664  012600                            MOV     (SP)+,R0        ;RESTORE R0
3052 016666  011646                            MOV     (SP),-(SP)
3053 016670  012766  000000  000002            MOV     #0,2(SP)        ;SET CPU TO LEVEL 0
3054 016676  000002                            RTI
3055
3056                                    ;SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
3057 016700  011646                    DSBINT: MOV     (SP),-(SP)
3058 016702  012766  000340  000002            MOV     #PRI07,2(SP)
3059 016710  000002                            RTI
3060
```

```
                                        .SBTTL  INTR    - INTERRUPT HANDLERS

016712                                  BGNSRV  INTR            ;DEFINE INTERRUPT ENTRY
016712                          INTR::
016712  012737  000001  002172          MOV     #1,INTRECV      ;SET FLAG TO SHOW INTERRUPT RECEIVED
016720  105037  016641                  CLRB    INTFLAG         ;CLEAR FLAG TO SAY WE GOT INTERRUPT
016724  132737  000001  016640          BITB    #IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
016732  001003                          BNE     1$              ;BR IF YES
016734  152737  000001  016641          BISB    #IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.

                                ;SAVE REGISTERS, MSG BUFFER, ETC.
016742                          1$:
016742                                  ENDSRV
016742                          L10026:
016742  000002                          RTI
```

WAITF - WAIT FOR SUBSYSTEM READY

```
                                        .SBTTL  WAITF   - WAIT FOR SUBSYSTEM READY
                                ;
                                ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
                                ;
                                ;INPUTS:
                                ;
                                ;       R5      ADDRESS OF FIRST DEVICE REGISTER
                                ;
                                ;OUTPUTS:
                                ;
                                ;       R0      CONTENTS OF LAST TSSR READ
                                ;       CARRY   SET - READY BIT SET
                                ;               CLR - TIMEOUT WAITING FOR READY
                                ;
016744                          WAITF:: BREAK                   ; DO A SUPVSR BREAK FIRST.
016744  104422                          TRAP    C$BRK
016746  012746  010000                  MOV     #10000.,-(SP)   ;BIG MSEC TIMER
016752                                  DELAY   1               ;DELAY 100US
016752  012727  000001                  MOV     #1,(PC)+
016756  000000                          .WORD   0
016760  013727  002116                  MOV     L$DLY,(PC)+
016764  000000                          .WORD   0
016766  005367  177772                  DEC     -6(PC)
016772  001375                          BNE     .-4
016774  005367  177756                  DEC     -22(PC)
017000  001367                          BNE     .-20
017002  016500  000000          2$:     MOV     TSSR(R5),R0     ;READ THE TSSR REGISTER
017006  105700                          TSTB    R0              ;TEST FOR READY BIT SET
                                
017010  100420                          BMI     3$              ; EXIT ON STOP FLAG.
017012                                  DELAY   1               ; WAIT 100 USEC
017012  012727  000001                  MOV     #1,(PC)+
017016  000000                          .WORD   0
017020  013727  002116                  MOV     L$DLY,(PC)+
017024  000000                          .WORD   0
017026  005367  177772                  DEC     -6(PC)
017032  001375                          BNE     .-4
017034  005367  177756                  DEC     -22(PC)
017040  001367                          BNE     .-20
017042  005316                          DEC     (SP)            ;REDUCE DELAY COUNT
017044  001356                          BNE     2$              ;RETRY UNTIL TIMER EXPIRES
017046  000241                          CLC                     ; C = 0, CONTROLLER STILL RUNNING...
017050  000401                          BR      4$              ;...OR HUNG UP AFTER 300 MSEC.
017052  000261                  3$:     SEC                     ; C = 1, CONTROLLER IS STOPPED.
017054  005326                  4$:     DEC     (SP)+           ;RESTORE STACK WITHOUT CHANGING CARRY BIT
017056  000207                          RTS     PC
```

CHKTSSR - CHECK TSSR FOR READY

```
                                .SBTTL  CHKTSSR - CHECK TSSR FOR READY

                        ;*
                        ;
                        ;THIS ROUTINE WAITS FOR READY IN THE TSSR
                        ;AND TESTS FOR AMBIGUOUS BIT SETTINGS IN TSSR.
                        ;
                        ;INPUT:
                        ;
                        ;       R5      ADDRESS OF CSR REGISTERS
                        ;
                        ;OUTPUT:
                        ;
                        ;       R0      CONTENTS OF TSSR
                        ;       CARRY   SET - OKAY
                        ;               CLR - NOT READY AMBIGUOUS, OR SC SET
                        ;
                        ;

017060                  CHKTSSR:
017060  004737  016744          JSR     PC,WAITF        ;WAIT FOR READY
017064  103014                 BCC     20$             ;BRANCH IF TIME OUT
017066  004737  016540          JSR     PC,CHKAMB       ;TSSR AMBIGOUS?
017072  103006                 BCC     10$             ;BR IF YES
017074  032700  100000          BIT     #SC,R0          ;SPECIAL CONDITION SET?
017100  001405                 BEQ     15$             ;BR IF NO
017102  032700  074000          BIT     #<SCE!BIE!RMR!NXM>,R0   ;ANY ERROR BITS SET?
017106  001402                 BEQ     15$             ;BR IF NO
017110  000241          10$:    CLC                     ;SET FAILURE
017112  000401                 BR      20$             ;
017114  000261          15$:    SEC                     ;SET SUCCESS
017116  000207          20$:    RTS     PC              ;RETURN TO CALLER
```

```
3140                                            .SBTTL  XNXM    - CHECK FOR NONEXISTENT MEMORY
3141                                    ;+
3142                                    ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
3143                                    ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
3144                                    ;               "C" = 0, ALL ADDRESSES OK.
3145                                    ;
3146                                    ;CALL:  MOV ADR1,R1
3147                                    ;       MOV ADR2,R2
3148                                    ;       JSR PC,NXM
3149                                    ;       RETURN                  ;TEST "C" AND PROCEED.
3150                                    ;
3151 017120  012737  017152  000004    XNXM:   MOV     #2$,@#4         ; SET BUSERR VECTOR.
3152 017126  012737  000200  000006            MOV     #PRI04,@#6
3153 017134  005003                            CLR     R3              ;FLAG.
3154 017136  005711                    1$:     TST     (R1)            ;TEST THE ADDRESS(ES).
3155                                                                    ;IF ANY TRAP, CONTINUE AT 2$.
3156 017140  020102                            CMP     R1,R2           ;OTHERWISE, CONTINUE HERE.
3157 017142  001407                            BEQ     3$              ;BR IF FINISHED (NO NEXM'S).
3158 017144  062701  000002                    ADD     #2,R1           ;SET NEXT ADDRESS...
3159 017150  000772                            BR      1$              ;...AND CONTINUE.
3160
3161 017152  005103                    2$:     COM     R3              ;GOT ONE, SET FLAG...
3162 017154  012716  017162                    MOV     #3$,(SP)
3163 017160  000002                            RTI                     ;...AND DISMISS INTERRUPT...
3164 017162                            3$:     CLRVEC  #4              ;...AND GIVE BACK THE VECTOR.
     017162  012700  000004                    MOV     #4,R0
     017166  104436                            TRAP    C$CVEC
3165 017170  005703                            TST     R3              ;DID WE CATCH ONE ??
3166 017172  001401                            BEQ     .+4             ;NO, "C" = 0, SKIP NEXT.
3167 017174  000261                            SEC                     ;YES, "C" = 1, (R1) = NEXM ADDR.
3168 017176  000207                            RTS     PC
3169
3170
3171
3172
3173                                            .SBTTL  TSTLOOP - CHECK ITERATION COUNT
3174                                    ;+
3175                                    ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
3176                                    ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
3177                                    ; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
3178                                    ;
3179                                    ; CALL: LOOPTO  ARG
3180                                    ;
3181 017200                            TSTLOOP::
3182 017200  005737  002136                    TST     NOITS           ; ITERATIONS INHIBITED?
3183 017204  001006                            BNE     1$              ; YES.
3184 017206  005737  002152                    TST     QVP             ; NO.
3185 017212  100403                            BMI     1$              ;LOOPS DISALLOWED IN QUICK PASS.
3186 017214  005337  002164                    DEC     LOOPCNT         ; BUMP LOOP COUNTER.
3187 017220  001002                            BNE     2$
3188 017222  000241                    1$:     CLC                     ;LOOP DISALLOWED, OR DONE.
3189 017224  000401                            BR      3$
3190 017226  000261                    2$:     SEC                     ;LOOP ENABLED.
3191 017230  000207                    3$:     RTS     PC
```

TSTLOOP - CHECK ITERATION COUNT

```
                                        .SBTTL  TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
                                ;+
                                ; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
                                ; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
                                ; IN THE CURRENT RUN SEQUENCE.
                                ; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
                                ;
                                ;INPUT:
                                ;
                                ;       R0      POINTER TO TEST ID ASCIZ STRING
                                ;
                                ;OUTPUT:
                                ;
                                ;       R5      ADDRESS OF FIRST DEVICE REGISTER
                                ;
                                ;IMPLICIT OUTPUTS:
                                ;
                                ;       TSTCNT  UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;       INTERRUPT LEVEL IS RASIED TO LEVEL OF
                                ;       THE DEVICE UNDER TEST
                                ;
                                ;

017232                          TSTSETUP::
017232  010046                          MOV     R0,-(SP)        ;SAVE THE TEST ID MESSAGE
017234  005037  003106                  CLR     SIFLAG          ; CLEAR "SOFT INIT" FLAG
017240  005037  017500                  CLR     ERRK            ; CLEAR LOCAL ERROR COUNTER.
017244  005037  005232                  CLR     EXTA            ; CLEAR ERROR EXTENSION FLAG.
017250  105037  016640                  CLRB    INTMASK         ; CLEAR INTERRUPT MASK (CHECK ERROR)
017254  013700  002150                  MOV     UNITN,R0        ; GET THE UNIT NUMBER,
017260  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
017262  005737  003062                  TST     NODEV           ; DID STARTUP FIND THE DEVICE?
017266  001430                          BEQ     4$              ; BR IF YES
017270  100010                          BPL     3$              ; BR IF NOT IDLE
017272  052760  160000  003130          BIS     #160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
017300                          ERRDF   1,NXR,NXRERR    ; NO DEVICE HERE  - PRINT IT
017300  104455                  TRAP    C$ERDF
017302  000001                  .WORD   1
017304  003636                  .WORD   NXR
017306  005176                  .WORD   NXRERR
017310  000407                  BR      2$
017312  052760  160001  003130  3$:     BIS     #160001,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
017320                          ERRDF   2,NOINIT        ; DEVICE NOT IDLE
017320  104455                  TRAP    C$ERDF
017322  000002                  .WORD   2
017324  004233                  .WORD   NOINIT
017326  000000                  .WORD   0
017330  012737  177777  003060  2$:     MOV     #-1,DUFLG       ; DROP THE UNIT
017336                          DODU    UNITN
017336  013700  002150                  MOV     UNITN,R0
017342  104451                  TRAP    C$DODU
017344                          DOCLN                   ; ABORT THE PASS
```

TSTSETUP PRINT TEST NAME AND INIT ERROR COUNTS

```
      017344  104444                          TRAP    C$DCLN
3240  017346  000423                          BR      5$
3241
3242  017350                          4$:     RFLAGS  R0              ; GET THE OPERATOR FLAGS.
      017350  104421                          TRAP    C$RFLA
3243  017352  032700  001000                  BIT     #PNT,R0         ; PRINT THE TEST NUMBERS?
3244  017356  001412                          BEQ     1$              ; BR IF NO
3245  017360  011600                          MOV     (SP),R0         ;GET THE ID MESSAGE
3246  017362                                  PRINTF  #TNAM,R0        ;DISPLAY THE TEST ID
      017362  010046                          MOV     R0,-(SP)
      017364  012746  017426                  MOV     #TNAM,-(SP)
      017370  012746  000002                  MOV     #2,-(SP)
      017374  010600                          MOV     SP,R0
      017376  104417                          TRAP    C$PNTF
      017400  062706  000006                  ADD     #6,SP
3247  017404  005237  002162          1$:     INC     TSTCNT          ; BUMP TEST COUNTER.
3248  017410                                  SETPRI  IPRI            ;PRIORITY THAT OF DEVICE
      017410  013700  002160                  MOV     IPRI,R0
      017414  104441                          TRAP    C$SPRI
3249  017416  005726                  5$:     TST     (SP)+           ;FIX UP THE STACK
3250  017420  013705  002154                  MOV     CSRADDR,R5      ; ADDRESS OF TSV REGISTERS ON UNIBUS
3251  017424  000207                          RTS     PC
3252  017426     045     123     045  TNAM:   .ASCIZ  '%S%T%A Test'
3253                                          .EVEN
```

```
                                        .SBTTL  TSTEND  - PRINT ERRORS RECEIVED
                                ;
                                ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
                                ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
                                ;
017442                          TSTEND: RFLAGS  R0
017442  104421                         TRAP    C$RFLA
017444  030027  020000                 BIT     R0,#IER
017450  001412                         BEQ     1$              ; BR IF "IER" NOT SET.
017452                                  PRINTF  #ESUM,ERRK      ; PRINT ERROR COUNT.
017452  013746  017500                 MOV     ERRK,-(SP)
017456  012746  017502                 MOV     #ESUM,-(SP)
017462  012746  000002                 MOV     #2,-(SP)
017466  010600                         MOV     SP,R0
017470  104417                         TRAP    C$PNTF
017472  062706  000006                 ADD     #6,SP
017476  000207                 1$:     RTS     PC

017500  000000                 ERRK:   0                       ; LOCAL ERROR COUNT.
017502     045     101     040 ESUM:   .ASCIZ  /%A %D%A ERRORS/
017521     105     122     122 EMAXDU: .ASCIZ  /ERROR LIMIT REACHED -- DROPPING UNIT/
                                        .EVEN

                                        .SBTTL  INCERK  - INCREMENT LOCAL ERROR COUNT
                                ;+
                                ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
                                ;-
017566  005237  017500         INCERK: INC     ERRK            ; INCREMENT LOCAL ERROR COUNT
017572  010046                         MOV     R0,-(SP)        ; SAVE R0
017574  013700  002150                 MOV     UNITN,R0        ; GET UNIT NUMBER,
017600  006300                         ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
017602  062700  003130                 ADD     #ERTABL,R0      ; R0 GETS ADDRESS OF ERROR TABLE ENTRY.
017606  005210                         INC     (R0)            ; INCREMENT THE DEVICE ERROR COUNT
017610  032710  007777                 BIT     #7777,(R0)      ; DID WE OVERFLOW THE FIELD?
017614  001001                         BNE     1$              ; BR IF NO.
017616  005310                         DEC     (R0)            ; YES -- BACK IT UP TO 7777.
017620  012600                 1$:     MOV     (SP)+,R0        ; RESTORE R0
017622  000207                         RTS     PC              ; RETURN TO CALLER.

017624  010046                 CKEMAX: MOV     R0,-(SP)        ; SAVE R0
017626  013700  002150                 MOV     UNITN,R0        ; GET UNIT NUMBER
017632  006300                         ASL     R0              ; ... AND MAKE IT A WORD OFFSET
017634  016000  003130                 MOV     ERTABL(R0),R0   ; GET ERROR TABLE ENTRY
017640  042700  170000                 BIC     #170000,R0      ; EXTRACT ERROR COUNT FIELD
017644  020037  002142                 CMP     R0,GERRMAX      ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
017650  103004                         BHIS    1$              ; BR IF YES
017652  023737  017500  002140         CMP     ERRK,LERRMAX    ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
017660  103417                         BLO     2$              ; BR IF NO
017662                          1$:     RFLAGS  R0              ; GET OPERATOR FLAGS
017662  104421                         TRAP    C$RFLA
017664  032700  000040                 BIT     #IDU,R0         ; IS DROPPING INHIBITED?
017670  001013                         BNE     2$              ; BR IF YES.
017672  012737  177777  003060         MOV     #-1,DUFLG       ; NO -- DROP THE UNIT
017700                                  ERRDF   4,EMAXDU
017700  104455                         TRAP    C$ERDF
017702  000004                         .WORD   4
017704  017521                         .WORD   EMAXDU
```

INCERK - INCREMENT LOCAL ERROR COUNT

```
     017706  000000                          .WORD   0
3301 017710                                  DODU    UNITN
     017710  013700  002150                  MOV     UNITN,R0
     017714  104451                          TRAP    C$DODU
3302 017716                                  DOCLN
     017716  104444                          TRAP    C$DCLN
3303 017720  012600                  2$:     MOV     (SP)+,R0        ; RESTORE R0
3304 017722  000207                          RTS     PC              ; RETURN TO CALLER
3305                                         .SBTTL  FATCHK - INC FATAL ERRORS AND CHECK FOR LIMIT
3306                                 ;+
3307                                 ;
3308                                 ;       CHECK FATAL COUNTER, AFTER INC, FOR MORE THAN 25
3309                                 ;       ERRORS AND IF OVER CALL UNIT DROP ROUTINE
3310                                 ;
3311                                 ;-
3312 017724                          FATCHK:
3313 017724                                  SAVREG                  ;BETTER SAVE THE REGISTERS
3314 017730  013701  002150                  MOV     UNITN,R1        ;PICK UP THE UNIT NUMBER
3315 017734  006301                          ASL     R1              ;MAKE IT INTO A BYTE OFFSET
3316 017736  062761  000001  003130          ADD     #1,ERTABL(R1)   ;ADD 1 TO THE PROPER UNIT'S ERROR COUNTER
3317 017744  005237  002170                  INC     FATFLG          ;BUMP FATAL ERROR COUNTER
3318 017750  023727  002170  000031          CMP     FATFLG,#25.     ;CHECK AGAINST 25
3319 017756  002406                          BLT     9$              ;BR, IF LESS THAN 25 ERRORS
3320 017760                                  RFLAGS  R0              ;READ THE FLAGS INTO R0
     017760  104421                          TRAP    C$RFLA
3321 017762  032700  040000                  BIT     #BIT14,R0       ;BR, IF LOOP ON ERROR IS SET
3322 017766  001002                          BNE     9$              ;OTHERWISE NEVER BE ABLE TO SCOPE ETC.
3323 017770  004737  017776                  JSR     PC,CKDROP       ;DROP UNIT IF ALLOWED
3324 017774  000207                  9$:     RTS     PC              ;RETURN ETC.
3325                                 ;
3326                                 ;
3327                                 ;
```

```
                                        .SBTTL  CKDROP  - CHECK IF UNIT SHOULD BE DROPPED
                                ;+
                                ; CHECK IF UNIT SHOULD BE DROPPED
                                ;-
017776  010046                  CKDROP: MOV     R0,-(SP)
020000                                  FORCERROR       1$,NOTSSR
020010                                  RFLAGS  R0
020010  104421                          TRAP    C$RFLA
020012  032700  000040                  BIT     #IDU,R0
020016  001010                          BNE     1$
020020  011600                          MOV     (SP),R0
020022  012737  177777  003060          MOV     #-1,DUFLG
020030                                  DODU    UNITN
020030  013700  002150                  MOV     UNITN,R0
020034  104451                          TRAP    C$DODU
020036                                  DOCLN                   ;ABORT THE PASS
020036  104444                          TRAP    C$DCLN
020040  012600                  1$:     MOV     (SP)+,R0
020042  000207                          RTS     PC


                                        .SBTTL  CONFIG  - DETERMINE CONFIGURATION OF SYSTEM
                                ;
                                ; SUBROUTINE - DETERMINE CONFIGURATION OF TK-25 SYSTEM.
                                ;
020044                          CONFIG:
020044  004737  016470                  JSR     PC,SOFINIT
020050  000207                          RTS     PC
```

```
3359                                            .SBTTL  KTON,KTOFF      - ENABLE/DISABLE MEMORY MANAGEMENT
3360                                    ;
3361                                    ; SUBROUTINE - ENABLE MEM MGT.
3362                                    ;
3363 020052  005737  003100            KTON:   TST     KTFLG           ; GOT KT?
3364 020056  001403                            BEQ     1$              ; NO.
3365 020060  012737  000001  177572            MOV     #1,SR0          ; YES. ENABLE KT11.
3366 020066  000207                    1$:     RTS     PC
3367
3368
3369
3370                                    ;
3371                                    ; SUBROUTINE - DISABLE MEM MGT.
3372                                    ;
3373 020070  005737  003100            KTOFF:  TST     KTFLG           ; GOT KT11?
3374 020074  001405                            BEQ     1$              ; NO.
3375 020076  000240                            NOP
3376 020100  000240                            NOP
3377 020102  012737  000000  177572            MOV     #0,SR0          ; DISABLE KT.
3378 020110  000207                    1$:     RTS     PC
3379
3380
```

```
                                        .SBTTL  SETMAP  - SETUP PAR6 MAPPING

                                ;+
                                ;
                                ;THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
                                ;AN 18 BIT ADDRESS. THE OFFSET INTO THE PAGE
                                ;IS RETURNED BIASED TO PAR6.
                                ;
                                ;INPUTS:
                                ;
                                ;       R0      HIGH ORDER ADDRESS BITS
                                ;       R1      LOW ORDER ADDRESS BITS
                                ;
                                ;OUTPUTS:
                                ;
                                ;       R0      OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
                                ;       CARRY   SET IF SUCCESS
                                ;               CLR IF ERROR
                                ;-
020112                          SETMAP:
020112                                  SAVREG                  ;SAVE R1-R4 UNTIL NEXT RETURN
020116  005737  003100                  TST     KTFLG           ;SYSTEM HAVE ABOVE 28K?
020122  001433                          BEQ     10$             ;BR IF NO
020124  010102                          MOV     R1,R2           ;SAVE LOW ORDER BITS
        000006                          .REPT   6
                                        ASR     R0              ;CONVERT WORD ADDRESS TO 32W BLOCKS
                                        ROR     R1              ;MAKE IT DOUBLE PRECISION
                                        .ENDR
020156  042701  000177                  BIC     #177,R1         ;ALINE FOR LOWER 4K BOUNDARY
020162  020137  003100                  CMP     R1,KTFLG        ;HIGHER THAN EXISTING MEMORY?
020166  103011                          BHIS    10$             ;BR IF YES
020170  010137  172354                  MOV     R1,@#KIPAR6     ;SETUP MAPPING REGISTER PAR6
020174  042702  160000                  BIC     #160000,R2      ;SETUP DISPLACEMENT IN PAGE
020200  062702  140000                  ADD     #140000,R2      ;ADD IN PAR6 BIAS
020204  010200                          MOV     R2,R0           ;RETURN IN R0
020206  000261                          SEC                     ;SET SUCCESS
020210  000401                          BR      15$             ;
020212  000241                  10$:    CLC                     ;SET FAILURE
020214  000207                  15$:    RTS     PC              ;RETURN
```

```
                                        .SBTTL  FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN
                                ;+
                                ;  FILL MEMORY WITH A BACKGROUND PATTERN
                                ;
                                ; INPUTS:
                                ;
                                ;       R0 = BACKGROUND PATTERN
                                ;       FREE    = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
                                ;       KTFLG   = SET TO HIGHEST MEMORY LOCATION IF > 28K.
                                ;
                                ; OUTPUTS:
                                ;
                                ;       NONE
                                ;-
                                ;
020216                          FILLMEM:
020216                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
020222  004737  020070                  JSR     PC,KTOFF        ;DISABLE KT.
020226  010003                          MOV     R0,R3           ;COPY TEST PATTERN
020230  013701  003072                  MOV     FREE,R1         ;GET FIRST FREE LOCATION
020234  013702  003074                  MOV     FRESIZ,R2       ;SIZE OF FREE SPACE BELOW 28K.
020240  010321                  10$:    MOV     R3,(R1)+        ;STORE A BACKGROUND WORD
020242  005302                          DEC     R2              ;DONE ALL MEMORY IN FREE SPACE?
020244  003375                          BGT     10$             ;BR IF NO
020246  005737  003100                  TST     KTFLG           ; GOT KT?
020252  001452                          BEQ     55$             ; NO. GET OUT.
020254  004737  020052                  JSR     PC,KTON         ; YES. ENABLE KT.
020260  005000                          CLR     R0              ;HIGH ORDER ADDRESS START
020262  013701  003104                  MOV     PST32W,R1       ;GET >28K START ADDRESS (IN 32W BLOCKS)
        000006                          .REPT   6
                                        CLC                     ;CLEAR C BIT
                                        ROL     R1              ;CONVERT BLOCKS TO WORDS
                                        ROL     R0              ;MAKE IT DOUBLE PRECISION
                                        .ENDR
020332  004737  020112                  JSR     PC,SETMAP       ;SETUP PAR6 MAPPING REGISTER
020336  010320                  30$:    MOV     R3,(R0)+        ;STORE TEST PATTERN IN >28K ADDRESS
020340  020027  160000                  CMP     R0,#160000      ;END OF PAR6 MAPPING AREA?
020344  103774                          BLO     30$             ;BR IF NO
020346  162700  020000                  SUB     #20000,R0       ;BACKUP INTO PAR6 MAPPING BEGIN
020352  062737  000200  172354          ADD     #200,@#KIPAR6   ;POINT TO NEXT 4K BLOCK >28K.
020360  023737  172354  003100          CMP     @#KIPAR6,KTFLG  ;END OF MEMORY?
020366  001402                          BEQ     50$             ;BR IF YES
020370  000137  020336                  JMP     30$             ;KEEP GOING ON ETC.
020374  004737  020070          50$:    JSR     PC,KTOFF        ; DISABLE KT.
020400  000207                  55$:    RTS     PC
```

```
                                        .SBTTL  CMPMEM  - COMPARE MEMORY TO BACKGROUND PATTERN
                                ;+
                                ; COMPARE MEMORY WITH A BACKGROUND PATTERN
                                ;
                                ; INPUTS:
                                ;
                                ;       R0 = BACKGROUND PATTERN
                                ;       FREE    = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
                                ;       KTFLG   = SET TO HIGHEST MEMORY LOCATION IF > 28K.
                                ;
                                ; OUTPUTS:
                                ;
                                ;       CARRY   - SET IF NO ERROR
                                ;       CARRY   - CLR IF ERROR
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;
                                ;       ERRHI   - ERROR HIGH ADDRESS
                                ;       ERRLO   - ERROR LOW ADDRESS
                                ;       EXPD    - EXPECTED DATA
                                ;       RECV    - RECEIVED DATA
                                ;-
020402                          CMPMEM:
020402                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
020406  010003                          MOV     R0,R3           ;COPY TEST PATTERN
020410  004737  020070                  JSR     PC,KTOFF        ;DISABLE KT.
020414  013701  003072                  MOV     FREE,R1         ;GET FIRST FREE LOCATION
020420  013702  003074                  MOV     FRESIZ,R2       ;SIZE OF FREE SPACE BELOW 28K.
020424  020311                  10$:    CMP     R3,(R1)         ;FREE SPACE LOCATION EQUAL TO EXPD?
020426  001411                          BEQ     15$             ;BR IF YES
020430  010137  002204                  MOV     R1,ERRLO        ;SAVE ADDRESS IN ERROR
020434  005037  002202                  CLR     ERRHI           ;NO HIGH ADDRESS
020440  010337  002176                  MOV     R3,EXPD         ;SAVE EXPD FOR ERROR REPORT
020444  011137  002200                  MOV     (R1),RECV       ;SAVE RECV FOR ERROR REPORT
020450  000474                          BR      50$             ;
020452  005721                  15$:    TST     (R1)+           ;POINT TO NEXT ADDRESS
020454  005302                          DEC     R2              ;DONE ALL MEMORY IN FREE SPACE?
020456  003362                          BGT     10$             ;BR IF NO
020460  005737  003100                  TST     KTFLG           ; GOT KT?
020464  001472                          BEQ     55$             ; NO. GET OUT.
020466  004737  020052                  JSR     PC,KTON         ; YES. ENABLE KT.
020472  005000                          CLR     R0              ;HIGH ORDER ADDRESS START
020474  013701  003104                  MOV     PST32W,R1       ;GET >28K START ADDRESS (IN 32W BLOCKS)
        000006                          .REPT   6
                                        ROL     R1              ;CONVERT BLOCKS TO WORDS
                                        ROL     R0              ;MAKE IT DOUBLE PRECISION
                                        .ENDR
020530  042701  000177                  BIC     #177,R1         ;ALIGNE 4K BOUNDARY
020534  010046                          MOV     R0,-(SP)        ;SAVE HIGH ORDER
020536  010146                          MOV     R1,-(SP)        ;SAVE LOW ORDER
020540  004737  020112                  JSR     PC,SETMAP       ;SETUP PAR6 MAPPING REGISTER
020544  010004                          MOV     R0,R4           ;COPY ADDRESS BIASED TO PAR6
020546  012601                          MOV     (SP)+,R1        ;RESTORE LOW ORDER IN NON PAR6 FORMAT
020550  012600                          MOV     (SP)+,R0        ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
020552  020314                  30$:    CMP     R3,(R4)         ;ABOVE 28K LOCATION EQUAL EXPD?
020554  001411                          BEQ     32$             ;BR IF YES
020556  010037  002202                  MOV     R0,ERRHI        ;SAVE HIGH ORDER IN ERROR
```

CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN

```
3528 020562  010137  002204                  MOV     R1,ERRLO        ;SAVE LOW ORDER IN ERROR
3529 020566  010337  002176                  MOV     R3,EXPD         ;SAVE EXPD FOR ERROR REPORT
3530 020572  011437  002200                  MOV     (R4),RECV       ;SAVE RECV FOR ERROR REPORT
3531 020576  000421                          BR      50$             ;
3532 020600  062701  000002          32$:    ADD     #2,R1           ;UPDATE NON PAR6 ADDRESS
3533 020604  005500                          ADC     R0              ;MAKE IT DOUBLE PRECISION ADD
3534 020606  062704  000002                  ADD     #2,R4           ;UPDATE PAR FORMAT ADDRESS
3535 020612  020427  160000                  CMP     R4,#160000      ;END OF PAR6 MAPPING AREA?
3536 020616  103755                          BLO     30$             ;BR IF NO
3537 020620  162704  020000                  SUB     #20000,R4       ;BACKUP INTO PAR6 MAPPING BEGIN
3538 020624  062737  000200  172354          ADD     #200,@#KIPAR6   ;POINT TO NEXT 4K BLOCK >28K.
3539 020632  023737  172354  003100          CMP     @#KIPAR6,KTFLG  ;END OF MEMORY?
3540 020640  101744                          BLOS    30$             ;BR IF NO
3541 020642  004737  020070          50$:    JSR     PC,KTOFF        ;TURN OFF MEMORY MAPPING
3542 020646  000241                          CLC                     ;SET FAILURE
3543 020650  000403                          BR      60$             ;
3544 020652  004737  020070          55$:    JSR     PC,KTOFF        ;TURN OFF MEMORY MAPPING
3545 020656  000261                          SEC                     ;SET SUCCESS
3546 020660  000207                  60$:    RTS     PC
3547
```

```
                                        .SBTTL  REGSAV  - SAVE R1-R5 ON STACK
                                ;*
                                ;
                                ;ROUTINE TO
                                ;SAVE R1 THROUGH R5 ON THE STACK
                                ;
                                ;CALLING SEQUENCE:
                                ;
                                ;       JSR     R5,REGSAV
                                ;
                                ;THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
                                ;THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
                                ;THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
                                ;REGISTERS.
                                ;
                                ;THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
                                ;CALLED VIA A JSR PC INSTRUCTION
                                ;
                                ;-

020662                          REGSAV:
020662                                  BREAK                   ;LOOK FOR CNTL C
020662  104422                          TRAP    C$BRK
020664  010446                          MOV     R4,-(SP)
020666  010346                          MOV     R3,-(SP)
020670  010246                          MOV     R2,-(SP)
020672  010146                          MOV     R1,-(SP)
020674  010546                          MOV     R5,-(SP)
020676  016605  000012                  MOV     10.(SP),R5
020702  004736                          JSR     PC,@(SP)+
020704  012601                          MOV     (SP)+,R1
020706  012602                          MOV     (SP)+,R2
020710  012603                          MOV     (SP)+,R3
020712  012604                          MOV     (SP)+,R4
020714  012605                          MOV     (SP)+,R5
020716                                  BREAK                   ;LOOK FOR CNTL C
020716  104422                          TRAP    C$BRK
020720  000207                          RTS     PC
```

GETPAT - GET 8 BIT PATTERN FROM OPERATOR

```
                                        .SBTTL  GETPAT  - GET 8 BIT PATTERN FROM OPERATOR
                                ;+
                                ;
                                ;ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
                                ;
                                ;INPUTS:
                                ;
                                ;       NONE.
                                ;
                                ;OUTPUTS:
                                ;
                                ;       R0      OCTAL NUMBER FROM THE OPERATOR
                                ;
                                ;CALLING SEQUENCE:
                                ;
                                ;       JSR     PC,GETPAT
                                ;
                                ;-
                                GETPAT::
                                        SAVREG                          ;SAVE THE GENERAL REGISTERS
020722                          1$:     GMANID  DATASC,PATDAT,0,377,0,377,NO
020726  104443                          TRAP    C$GMAN
020730  000406                          BR      10000$
020732  020756                          .WORD   PATDAT
020734  000022                          .WORD   T$CODE
020736  020760                          .WORD   DATASC
020740  000377                          .WORD   377
020742  000000                          .WORD   T$LOLIM
020744  000377                          .WORD   T$HILIM
020746                          10000$:
020746                                  BNCOMPLETE      1$      ;RETRY IF ERROR
020746  103367                          BCC     1$
020750  013700  020756                  MOV     PATDAT,R0       ;DATA PATTERN FROM OPERATOR
020754  000207                          RTS     PC              ;RETURN TO CALLER

                                ;+
                                ;LOCAL DATA AREA
                                ;-

020756  000000                  PATDAT: .WORD   0               ;TEMPORARY STORAGE FOR DATA
020760    105     116     124   DATASC: .ASCIZ  'ENTER DATA PATTERN'
                                        .EVEN
```

```
                                        .SBTTL  GETSEL  - ISSUE MENU AND GET OPERATOR RESPONSE
                                ;+
                                ;ROUTINE TO ISSUE A MENU AND GET
                                ;THE OPERATOR'S RESPONSE.
                                ;
                                ;INPUTS:
                                ;
                                ;       R0      ADDRESS OF ASCIZ STRING OF MENU
                                ;       R1      MAXIMUM ALLOWABLE OPERATOR RESPONSE
                                ;
                                ;OUTPUTS·
                                ;
                                ;       R0      NUMBER OF THE OPERATOR'S SELECTION
                                ;
021004                          GETSEL::
021004                                  SAVREG                          ;SAVE GENERAL REGISTERS
021010  010002                         MOV     R0,R2                   ;SAVE THE MENU ADDRESS
021012  010203                 1$:     MOV     R2,R3                   ;START OF MENU STRING
021014  005713                 2$:     TST     (R3)                    ;END OF ASCII ?
021016  001412                         BEQ     3$                      ;BRANCH IF ALL LINES DISPLAYED
021020                                  PRINTF  #SELASC,(R3)+           ;DISPLAY THE MENU
021020  012346                         MOV     (R3)+,-(SP)
021022  012746  021170                 MOV     #SELASC,-(SP)
021026  012746  000002                 MOV     #2,-(SP)
021032  010600                         MOV     SP,R0
021034  104417                         TRAP    C$PNTF
021036  062706  000006                 ADD     #6,SP
021042  000764                         BR      2$
021044                          3$:     GMANID  MENASC,MENRES,D,-1,0,-1,NO
021044  104443                         TRAP    C$GMAN
021046  000406                         BR      10001$
021050  021224                         .WORD   MENRES
021052  000042                         .WORD   T$CODE
021054  021175                         .WORD   MENASC
021056  177777                         .WORD   -1
021060  000000                         .WORD   T$LOLIM
021062  177777                         .WORD   T$HILIM
021064                          10001$:
021064                                  BNCOMPLETE      1$      ;RETRY IF ERROR
021064  103352                         BCC     1$
021066  013700  021224                 MOV     MENRES,R0               ;GET THE OPERATOR'S REPLY
021072  020001                         CMP     R0,R1                   ;COMPARE TO MAXIMUM ALLOWED
021074  101411                         BLOS    5$                      ;BRANCH IF OK
021076                                  PRINTF  #MENERR                 ;DISPLAY ERROR MESSAGE
021076  012746  021122                 MOV     #MENERR,-(SP)
021102  012746  000001                 MOV     #1,-(SP)
021106  010600                         MOV     SP,R0
021110  104417                         TRAP    C$PNTF
021112  062706  000004                 ADD     #4,SP
021116  000735                         BR      1$                      ;RETRY
021120  000207                 5$:     RTS     PC                      ;RETURN TO CALLER
021122     045     116     045 MENERR: .ASCIZ   '%N%A *** Menu Selection Too Large ***'
021170     045     116     045 SELASC: .ASCIZ   '%N%T'
021175     105     156     164 MENASC: .ASCIZ   'Enter Menu Selection: '
                                        .EVEN
021224  000000                 MENRES:         .WORD   0
```

```
                                        .SBTTL  CHKMAN  - CHECK MANUAL INTERVENTION LEGALITY
                                ;+
                                ;
                                ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
                                ;
                                ;INPUT:
                                ;
                                ;       NONE.
                                ;
                                ;OUTPUT:
                                ;
                                ;       CARRY   0       MANUAL INTERVENTION NOT ALLOWED
                                ;               1       MANUAL INTERVENTION IS OK
                                ;
                                ;SIDE EFFECTS:
                                ;
                                ;       A MESSAGE IS DISPLAYED WARNING THAT TEST IS
                                ;       NOT EXECUTED IF MANUAL INTERVENTION IS NOT
                                ;       ALLOWED.
                                ;
                                ;-

021226                          CHKMAN::
                                        SAVREG                  ;SAVE THE REGISTERS
021226                                  MANUAL                  ;SEE IF MANUAL INTERVENTION OK
021232  104450                          TRAP    C$MANI
                                        BCOMPLETE 1$            ;BRANCH IF ALLOWED
021234                                  BCS     1$
021234  103411                          PRINTF  #NOMAN          ;PRINT THE WARNING MESSAGE
021236                                  MOV     #NOMAN,-(SP)
021236  012746  021262                  MOV     #1,-(SP)
021242  012746  000001                  MOV     SP,R0
021246  010600                          TRAP    C$PNTF
021250  104417                          ADD     #4,SP
021252  062706  000004                  CLC                     ;CLEAR CARRY FOR ERROR
021256  000241                  1$:     RTS     PC              ;RETURN
021260  000207
                                NOMAN:  .ASCIZ  '<N><A *** Manual Intervention not Allowed - Test Aborted ***'
021262  045  116  045                   .even
```

```
ENVIRN  - SETUP FREE DIAGNOSTIC SPACE

                                                .SBTTL  ENVIRN  - SETUP FREE DIAGNOSTIC SPACE
                                        ;
                                        ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
                                        ;
021356                                  ENVIRN: MEMORY  R0
021356  104431                                  TRAP    C$MEM
021360  010037  003072                          MOV     R0,FREE         ; GET 1ST FREE ADDRESS...
021364  062737  000002  003072                  ADD     #2,FREE
021372  011037  003074                          MOV     (R0),FRESIZ     ;...AND WORD COUNT.
021376  162737  000004  003074                  SUB     #4,FRESIZ
021404  013702  002012                          MOV     L$UNIT,R2       ; GET NUMBER OF UNITS
021410  162737  000007  003074  10$:            SUB     #7,FRESIZ       ; TAKE AWAY 7 WORDS PER UNIT
021416  005302                                  DEC     R2
021420  001373                                  BNE     10$
021422  013700  003072                          MOV     FREE,R0         ;GET FIRST FREE ADDRESS
021426  063700  003074                          ADD     FRESIZ,R0       ;POINT TO LAST FREE ADDRESS
021432  162700  000002                          SUB     #2,R0           ;BACKUP 1 WORD
021436  010037  003076                          MOV     R0,FREEHI       ;STORE LAST FREE ADDRESS
021442  000207                                  RTS     PC              ;RETURN
```

```
                                        .SBTTL  KTINIT  - SETUP KT11 MEMORY MANAGEMENT REGISTERS
                                ;+
                                ;
                                ;ROUTINE TO INIT KT-11
                                ;
                                ;-

021444                          KTINIT:
021444  005037  003100                  CLR     KTFLG           ; INIT >28K MEMORY FLAG
021450  005037  003102                  CLR     KTENABLE        ; INIT TEST >28K FLAG
021454  023727  002120  001577          CMP     L$HIME,#1577    ; GOT ENOUGH MEMORY (>28K)?
021462  101444                          BLOS    9$              ; NO.
021464  013700  000004                  MOV     @#ERRVEC,R0     ; SAVE OLD ERR VEC PTR.
021470  012737  021562  000004          MOV     #2$,@#ERRVEC    ; SET ERR VEC PTR.
021476  005737  177572                  TST     @#SR0           ; GOT KT11?
021502  000240                          NOP                     ;  (TRAP IF NO).
021504  013737  002120  003100          MOV     L$HIME,KTFLG    ; YES. SET KT FLAG.
021512  042737  000177  003100          BIC     #177,KTFLG      ;
021520  010037  000004                  MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
021524  005000                          CLR     R0              ; R0 = AR DATA.
021526  012701  172340                  MOV     #KIPAR0,R1      ; R1 = KI REGS PTR.
021532  012761  077406  177740  1$:     MOV     #77406,-40(R1)  ; SET DESCRIPTOR REG.
021540  010021                          MOV     R0,(R1)+        ; SET KIPAR REG.
021542  062700  000200                  ADD     #200,R0         ; BUMP AR DATA BY "4K".
021546  020027  002000                  CMP     R0,#2000        ; AT "I/O"?
021552  001367                          BNE     1$              ; NO.
021554  012741  177600                  MOV     #177600,-(R1)   ; YES. SET KTPAR7 FOR I/O.
021560  000405                          BR      9$

021562  012716  021570          2$:     MOV     #6$,(SP)        ; SET UP RETURN
021566  000002                          RTI                     ; RTI TO NEXT LOCATION

021570  010037  000004          6$:     MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.

021574  000207                  9$:     RTS     PC
                                ;       .IIF DF ONEFILE,        .PAGE
```

PROTECTION TABLE

```
3763                                            .SBTTL ,PROTECTION TABLE
3764 021576                                     BGNPROT
     021576                             L$PROT::
3765 021576  177777  177777  177777             .WORD   -1, -1, -1, -1  ;NO DEVICE PROTECTION REQUIRED.
3766 021606                                     ENDPROT
3767
```

INITIALIZE SECTION

```
                                        .SBTTL  INITIALIZE SECTION

                                ;++
                                ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
                                ;AT THE BEGINNING OF EACH PASS.
                                ;
                                ;IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
                                ;IF "CONTINUE", NOTHING IS REQUIRED.
                                ;
                                ; -
                                ;+
                                ;INSERT TEMPORARY JUMP TO ODT
                                ;-
021606                                  BGNINIT
021606                          L$INIT::
021606                          40$:
021606  012737  005672  002146          MOV     #EPRT1,EPRTSW   ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
021614  005037  003106                  CLR     SIFLAG          ;CLEAR "SOFT INIT" FLAG
021620  005037  003102                  CLR     KTENABLE        ;CLEAR TEST ABOVE 28K FLAG
021624  005037  002246                  CLR     RAMSIZ          ;CLEAR RAM SIZE FOR RAMERR ROUTINE
021630                                  READEF  #EF.CONTINUE
021630  012700  000036                  MOV     #EF.CONTINUE,R0
021634  104447                          TRAP    C$REFG
021636                                  BNCOMPLETE 1$
021636  103023                          BCC     1$
021640  023737  002150  002012          CMP     UNITN,L$UNIT    ;UNIT IN RANGE?
021646  103064                          BHIS    4$              ;BR IF NO.
021650  005737  003060                  TST     DUFLG           ;DROPPED UNIT?
021654  100466                          BMI     NXTU            ;BR IF YES
021656  013701  002150                  MOV     UNITN,R1
021662  006301                          ASL     R1
021664  005761  003130                  TST     ERTABL(R1)
021670  001512                          BEQ     SETU
021672  032761  040000  003130          BIT     #BIT14,ERTABL(R1) ;DROPPED?
021700  001054                          BNE     NXTU
021702                                  EXIT    INIT            ;DO NOTHING IF "CONTINUE .
021702  104432                          TRAP    C$EXIT
021704  000412                          .WORD   L10030-.
021706                          1$:     READEF  #EF.NEW
021706  012700  000035                  MOV     #EF.NEW,R0
021712  104447                          TRAP    C$REFG
021714                                  BNCOMPLETE NXTU         ;TAKE NEXT UNIT IF NOT NEW PASS.
021714  103046                          BCC     NXTU
021716                                  READEF  #EF.START
021716  012700  000040                  MOV     #EF.START,R0
021722  104447                          TRAP    C$REFG
021724                                  BCOMPLETE 2$
021724  103404                          BCS     2$
021726                                  READEF  #EF.RESTART
021726  012700  000037                  MOV     #EF.RESTART,R0
021732  104447                          TRAP    C$REFG
021734                                  BNCOMPLETE 31$
021734  103025                          BCC     31$
021736                          2$:                             ;1ST PASS, BUS-INIT...
021736                                  BRESET                  ;BUS RESET.
021736  104433                          TRAP    C$RESET
021740  005037  002162                  CLR     TSTCNT          ;NUMBER OF TESTS RUN IN PASS
```

INITIALIZE SECTION

```
3810 021744  005037  002170                CLR     FATFLG          ;RESET FLAG TO ZERO "FATAL ERRORS"
3811 021750  005037  003332        19$:    CLR     SKIPT           ;CLEAR THE SUBTEST "SKIPPER"
3812 021754                        20$:
3813 021754  012737  177777  002152        MOV     #-1,QVP         ;...QUICK VERIFY...
3814 021762  004737  021356                JSR     PC,ENVIRN       ;SET ENVIRONMENT.
3815 021766  004737  021444                JSR     PC,KTINIT       ;INITIALIZE KT MEMORY MANAGEMENT
3816 021772  012700  003130                MOV     #ERTABL,R0
3817 021776  005020                30$:    CLR     (R0)+           ;CLEAR THE ERROR TABLE
3818 022000  020027  003330                CMP     R0,#ERTABE
3819 022004  103774                        BLO     30$
3820 022006  000404                        BR      4$
3821 022010  005037  002152        31$:    CLR     QVP
3822 022014  000137  022064                JMP     PASRPT          ;GO REPORT THE STATUS
3823
3824 022020                        4$:
3825 022020  012737  177777  002150  NEWPAS: MOV     #-1,UNITN       ;INIT UNIT NUMBER...
3826 022026  005037  002166                CLR     DEVCNT          ;CLEAR COUNT OF DEVICES RUNNING
3827 022032                        NXTU:   BREAK
     022032  104422                        TRAP    C$BRK
3828 022034  005237  002150                INC     UNITN           ;...AND SET NEXT UNIT NUMBER.
3829 022040  023737  002150  002012        CMP     UNITN,L$UNIT
3830 022046  103423                        BLO     SETU
3831 022050  012737  177777  003060        MOV     #-1,DUFLG
3832 022056  000401                        BR      11$
3833 022060                                DOCLN                   ;ABORT, NO MORE UNITS.
     022060  104444                        TRAP    C$DCLN
3834 022062  000240                11$:    NOP
3835 022064                        PASRPT:
3836 022064  023727  002012  000001        CMP     L$UNIT,#1       ;HOW MANY UNITS SELECTED?
3837 022072  101752                        BLOS    NEWPAS          ;BR IF ONLY 1
3838 022074  005737  002166                TST     DEVCNT          ;ARE ANY STILL RUNNING?
3839 022100  001747                        BEQ     NEWPAS          ;BR IF NO
3840 022102                                RFLAGS  R0
     022102  104421                        TRAP    C$RFLA
3841 022104  032700  000100                BIT     #ISR,R0         ;SHOULD WE PRINT STATISTICS
3842 022110  001343                        BNE     NEWPAS          ;BR IF NO
3843
3844 022112                                DORPT
     022112  104424                        TRAP    C$DRPT
3845 022114  000741                        BR      NEWPAS
3846 022116                        10$:
3847
3848 022116                        SETU:   GPHARD  UNITN,R0        ;GET UNIT N P-TABLE POINTER.
     022116  013700  002150                MOV     UNITN,R0
     022122  104442                        TRAP    C$GPHRD
3849 022124                                BNCOMPLETE NXTU         ;BR IF UNIT NOT AVAILABLE.
     022124  103342                        BCC     NXTU
3850 022126  005037  003060                CLR     DUFLG           ;CLEAR "DROPPED" FLAG.
3851 022132  005237  002166                INC     DEVCNT
3852 022136  012001                        MOV     (R0)+,R1        ;GET 1ST REGISTER ADDRESS.
3853 022140  010137  002154                MOV     R1,CSRADDR      ;ADDRESS OF REGISTERS OF UNIT UNDER TEST
3854
3855 022144  012001                        MOV     (R0)+,R1        ;GET VECTOR ADDRESS.
3856 022146  011002                        MOV     (R0),R2         ;GET INTERRUPT PRIORITY
3857 022150  010237  002160                MOV     R2,IPRI         ;SET INTERRUPT PRIORITY.
3858 022154  010137  002156                MOV     R1,IVEC         ;SET INTERRUPT VECTOR POINTER...
3859 022160  012721  016712                MOV     #INTR,(R1)+     ;...VECTOR...
```

INITIALIZE SECTION

```
3860 022164  010221                        MOV     R2,(R1)+        ;...AND PRIORITY.
3861
3862 022166                          1$:
3863                                 ;       TST     QVP             ;1ST PASS ??
3864                                 ;       BEQ     5$              ;NO, SKIP THE PASS 1 STUFF.
3865
3866                                 ;
3867                                 ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
3868                                 ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
3869                                 ;
3870 022166  013701  002150                  MOV     UNITN,R1
3871 022172  006301                        ASL     R1
3872 022174  052761  100000  003130        BIS     #BIT15,ERTABL(R1) ;SAY DEVICE RUNNING
3873 022202  005037  005232                CLR     EXTA            ;CLEAR ERROR EXTENSION FLAG.
3874 022206  023727  002012  000001        CMP     L$UNIT,#1       ;ARE WE TESTING MULTIPLE UNITS?
3875 022214  101416                        BLOS    10$             ;BR IF NO.
3876 022216                                RFLAGS  R0              ;YES -- GET OPERATOR FLAGS.
     022216  104421                        TRAP    C$RFLA
3877 022220  032700  001000                BIT     #PNT,R0         ;SHOULD WE PRINT UNIT #?
3878 022224  001412                        BEQ     10$             ;BR IF NOT.
3879 022226                                PRINTF  #PUNIT,UNITN    ;PRINT THE UNIT #
     022226  013746  002150                MOV     UNITN,-(SP)
     022232  012746  022320                MOV     #PUNIT,-(SP)
     022236  012746  000002                MOV     #2,-(SP)
     022242  010600                        MOV     SP,R0
     022244  104417                        TRAP    C$PNTF
     022246  062706  000006                ADD     #6,SP
3880 022252                          10$:
3881 022252  005037  003062                CLR     NODEV
3882 022256  013701  002154                MOV     CSRADDR,R1      ;ADDRESS OF FIRST REGISTER
3883 022262  010102                        MOV     R1,R2           ;START OF REGISTERS\
3884 022264  062702  000000                ADD     #TSSR,R2        ;ADDRESS OF TSSR REGISTER
3885 022270  004737  017120                JSR     PC,XNXM         ;TEST BOTH CONTROLLER REGISTERS...
3886 022274  103005                        BCC     2$              ;...AND BR IF ALL OK,
3887 022276  010137  003062                MOV     R1,NODEV        ;FLAG DEVICE AS NON-EXISTENT
3888 022302  012737  177777  003060        MOV     #-1,DUFLG       ;DROP THIS UNIT.
3889 022310                          2$:
3890                                 ;
3891                                 ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
3892                                 ;
3893 022310                          5$:     SETPRI  #PRI00          ;ENABLE INTERRUPTS.
     022310  012700  000000                MOV     #PRI00,R0
     022314  104441                        TRAP    C$SPRI
3894 022316                                ENDINIT
     022316                          L10030:
     022316  104411                        TRAP    C$INIT
3895
3896 022320     045     116     045  PUNIT:  .ASCIZ  /%N%N%A***** TESTING UNIT %D2%A *****/
3897                                        .EVEN
```

```
                                        .SBTTL  ADD AND DROP UNITS SECTIONS

                                ;++
                                ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                ; TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
                                ;    OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
                                ;--
022366                                  BGNAU
022366                          L$AU::
022366  010001                          MOV     R0,R1           ; GET UNIT TO BE ADDED (R0)
022370  006301                          ASL     R1              ; MAKE IT A WORD INDEX
022372  052761  100000  003130          BIS     #100000,ERTABL(R1) ; SET THE "ACTIVE" BIT
022400  042761  040000  003130          BIC     #40000,ERTABL(R1)  ; CLEAR THE "DROPPED" BIT
022406                                  PRINTF  #1$,R0
022406  010046                          MOV     R0,-(SP)
022410  012746  022434                  MOV     #1$,-(SP)
022414  012746  000002                  MOV     #2,-(SP)
022420  010600                          MOV     SP,R0
022422  104417                          TRAP    C$PNTF
022424  062706  000006                  ADD     #6,SP
022430                                  EXIT    AU
022430  000167                          .WORD   J$JMP
022432  000026                          .WORD   L10031-2-.
022434     045     116     045  1$:     .ASCIZ  /%N%A  UNIT %D%A ADDED/
                                        .EVEN

022462                                  ENDAU                   ; UNUSED.
022462                          L10031:
022462  104452                          TRAP    C$AU
                                ;++
                                ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                ; TO BE REMOVED FROM THE TEST LIST.
                                ;
                                ; SUPVSR DOES THE "DROPPING".  THIS IS JUST TO TELL THE MAN.
                                ; "DROPPED" UNITS ARE RE-SELECTED ON OPERATOR "STA" OR "ADD"
                                ; COMMAND, OTHERWISE REMAIN INACTIVE.  THE "DISPLAY" COMMAND
                                ; WILL PRINT ALL DROPPED UNITS, AND THE P-TABLES OF THOSE
                                ; WHICH ARE STILL ACTIVE.
                                ; UPON ENTRY, R0 CONTAINS THE UNIT TO BE DROPPED.

022464                                  BGNDU
022464                          L$DU::
022464  012737  177777  003060          MOV     #-1,DUFLG
022472  010001                          MOV     R0,R1
022474  006301                          ASL     R1
022476  052761  140000  003130          BIS     #140000,ERTABL(R1)      ; SAY DROPPED
022504  000240  000240  000240          240,240,240                     ; ?????????
022512                                  PRINTF  #1$,R0 _
022512  010046                          MOV     R0,-(SP)
022514  012746  022540                  MOV     #1$,-(SP)
022520  012746  000002                  MOV     #2,-(SP)
022524  010600                          MOV     SP,R0
022526  104417                          TRAP    C$PNTF
022530  062706  000006                  ADD     #6,SP
022534                                  EXIT    DU
022534  000167                          .WORD   J$JMP
022536  000030                          .WORD   L10032-2-.
```

```
3936 022540     045     116     045  1$:     .ASCIZ  /%N%A  UNIT %D%A DROPPED/
3937                                         .EVEN
3938 022570                                  ENDDU
     022570                             L10032:
     022570  104453                          TRAP    C$DU
3939                                         ;++
3940                                         ; AUTO-DROP CODE SECTION.
3941                                         ;--
3942 022572                                  BGNAUTO
     022572                             L$AUTO::
3943 022572  012703  000550                 MOV     #360.,R3        ;ENOUGH TIME FOR 2400' REEL TO REWIND
3944 022576  004737  016744            10$: JSR     PC,WAITF        ;WAIT FOR SSR TO SET
3945 022602  103420                        BCS     20$             ;LEAVE WHEN SSR IS SET
3946 022604                                 DELAY   250.            ;WAIT FOR .25 SECONDS
     022604  012727  000372                MOV     #250.,(PC)+
     022610  000000                        .WORD   0
     022612  013727  002116                MOV     L$DLY,(PC)+
     022616  000000                        .WORD   0
     022620  005367  177772                DEC     -6(PC)
     022624  001375                        BNE     .-4
     022626  005367  177756                DEC     -22(PC)
     022632  001367                        BNE     .-20
3947 022634  005303                        DEC     R3              ;BUMP COUNTER DOWN
3948 022636  001357                        BNE     10$             ;KEEP GOING
3949 022640  004737  017776                JSR     PC,CKDROP       ;TRY AND DROP UNIT
3950 022644                             20$:
3951 022644                                 ENDAUTO                 ; UNUSED.
     022644                             L10033:
     022644  104461                          TRAP    C$AUTO
```

```
                                        .SBTTL  CLEAN-UP AND REPORT CODING SECTIONS

                                ;++
                                ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS
                                ; EXECUTED AT THE END OF EACH PASS (OR SUB-PASS).
                                ; USE TO RETURN DEVICE UNDER TEST TO A NEUTRAL STATE.
                                ;--
022646                                  BGNCLN
022646                          L$CLEAN::
022646  005737  003060                  TST     DUFLG           ;"DROPPED" FLAG IS SET ON...
022652  100407                          BMI     1$              ;...AND GROSS CONTROLLER FAULT...
                                                                ;...DON'T TRY TO XCT CLEANUP CODE.

022654  013705  002154                  MOV     CSRADDR,R5      ; ADDRESS OF TSV REGISTERS ON UNIBUS
022660  012765  000000  000000          MOV     #0,TSSR(R5)     ;DO SOFT INIT
022666  004737  016744                  JSR     PC,WAITF
022672                          1$:
022672                          2$:     ENDCLN
022672                          L10034:
022672  104412                          TRAP    C$CLEAN

                                ;++
                                ; THE REPORT CODING SECTION CONTAINS THE
                                ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                                ;--
022674                                  BGNRPT
022674                          L$RPT::
022674                                  PRINTS  #DEVSUM
022674  012746  023136                  MOV     #DEVSUM,-(SP)
022700  012746  000001                  MOV     #1,-(SP)
022704  010600                          MOV     SP,R0
022706  104416                          TRAP    C$PNTS
022710  062706  000004                  ADD     #4,SP
022714  010246                          MOV     R2,-(SP)
022716  010346                          MOV     R3,-(SP)
022720  010446                          MOV     R4,-(SP)
022722  012704  003130                  MOV     #ERTABL,R4      ; GET START OF ERROR TABLE.
022726  005003                          CLR     R3              ; CLEAR UNIT NUMBER
022730  011402                  1$:     MOV     (R4),R2         ; GET ERROR TABLE ENTRY & TEST IT.
022732  001407                          BEQ     4$              ; ZERO IF UNIT NOT RUN
022734  100066                          BPL     4$
022736  032702  040000                  BIT     #BIT14,R2       ; WAS UNIT DROPPED?
022742  001015                          BNE     2$              ; BR IF YES
022744  042702  170000                  BIC     #^C7777,R2      ; GET ERROR COUNT FIELD
022750                                  PRINTS  #DEVONL,R3,R2   ; PRINT
022750  010246                          MOV     R2,-(SP)
022752  010346                          MOV     R3,-(SP)
022754  012746  023173                  MOV     #DEVONL,-(SP)
022760  012746  000003                  MOV     #3,-(SP)
022764  010600                          MOV     SP,R0
022766  104416                          TRAP    C$PNTS
022770  062706  000010                  ADD     #10,SP
022774  000446                          BR      4$
022776  020227  160000          2$:     CMP     R2,#160000      ; WAS UNIT NON-EXISTENT?
023002  001012                          BNE     3$              ; BR IF NO
023004                                  PRINTS  #DEVNXR,R3
023004  010346                          MOV     R3,-(SP)
023006  012746  023255                  MOV     #DEVNXR,-(SP)
```

```
                023012  012746  000002                  MOV     #2,-(SP)
                023016  010600                          MOV     SP,R0
                023020  104416                          TRAP    C$PNTS
                023022  062706  000006                  ADD     #6,SP
          3992  023026  000431                          BR      4$
          3993  023030  020227  160001          3$:     CMP     R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
          3994  023034  001012                          BNE     30$             ; BR IF NO.
          3995  023036                                  PRINTS  #DEVNRD,R3
                023036  010346                          MOV     R3,-(SP)
                023040  012746  023337                  MOV     #DEVNRD,-(SP)
                023044  012746  000002                  MOV     #2,-(SP)
                023050  010600                          MOV     SP,R0
                023052  104416                          TRAP    C$PNTS
                023054  062706  000006                  ADD     #6,SP
          3996  023060  000414                          BR      4$
          3997  023062  042702  170000          30$:    BIC     #^C7777,R2
          3998  023066                                  PRINTS  #DEVDRO,R3,R2
                023066  010246                          MOV     R2,-(SP)
                023070  010346                          MOV     R3,-(SP)
                023072  012746  023420                  MOV     #DEVDRO,-(SP)
                023076  012746  000003                  MOV     #3,-(SP)
                023102  010600                          MOV     SP,R0
                023104  104416                          TRAP    C$PNTS
                023106  062706  000010                  ADD     #10,SP
          3999  023112  062704  000002          4$:     ADD     #2,R4
          4000  023116  005203                          INC     R3
          4001  023120  020427  003330                  CMP     R4,#ERTABE
          4002  023124  103701                          BLO     1$
          4003  023126  012604                          MOV     (SP)+,R4
          4004  023130  012603                          MOV     (SP)+,R3
          4005  023132  012602                          MOV     (SP)+,R2
          4006  023134                                  ENDRPT                  ; UNUSED.
                023134                          L10035:
                023134  104425                          TRAP    C$RPT
          4007
          4008
          4009  023136      045     116     045 DEVSUM: .ASCIZ   /%N%ADEVICE STATUS SUMMARY:%N/
          4010  023173      045     101     040 DEVONL: .ASCIZ   /%A  UNIT %D3%A   CONTROLLER READY,  ERRORS = %D%N/
          4011  023255      045     101     040 DEVNXR: .ASCIZ   /%A  UNIT %D3%A   DROPPED, NON-EXISTENT REGISTER%N/
          4012  023337      045     101     040 DEVNRD: .ASCIZ   /%A  UNIT %D3%A   DROPPED, NOT READY AT STARTUP%N/
          4013  023420      045     101     040 DEVDRO: .ASCIZ   /%A  UNIT %D3%A   DROPPED, ERRORS = %D%N/
          4014                                        .EVEN
          4017
          4024
          4030
          4038
```

```
                                .SBTTL  TEST  1: BUS RESET TEST


                        ;       THIS TEST VERIFIES THAT THE MODULE'S DEVICE REGISTERS ARE
                        ;       ACCESSIBLE ON THE BUS (SUBTEST 1) AND THEN CHECKS THAT THE
                        ;       BUILT-IN INITIALIZATION SELF-TEST MICRODIAGNOSTIC DID NOT FIND
                        ;       ANY BASIC PROBLEMS IN THE MODULE.  AREAS OF LOGIC TESTED BY THE
                        ;       SELF-TEST SEQUENCE ARE AS FOLLOWS:  ROM AND PIPELINE REGISTER,
                        ;       SEQUENCER, INTERNAL BUSES, 2901 MICROPROCESSOR, AND, RAM.  THIS
                        ;       TEST INITIALIZES THE CONTROLLER BY ISSUING THE BUS INIT SIGNAL
                        ;       VIA A RESET INSTRUCTION, OR BY WRITING INTO THE TSSR REGISTER,
                        ;       WAITS A PERIOD OF TIME (TO ALLOW THE CONTROLLER'S INITIALIZATION
                        ;       MICRODIAGNOSTIC SEQUENCE TO BE COMPLETED), AND THEN CHECKS THE
                        ;       CONTENTS OF THE TSSR REGISTER.  SUCCESSFUL INITIALIZATION IS
                        ;       INDICATED BY SUBSYSTEM READY (SSR) AND NEED BUFFER ADDRESS (NBA)
                        ;       BITS BEING SET (1) AND ALL OTHER BITS (EXCEPT A17 AND A16 AND
                        ;       OFL, WHICH ARE IGNORED FOR THIS TEST) BEING CLEAR (0).  IF THE
                        ;       CONTENTS OF TSSR ARE NOT AS EXPECTED, AN ERROR REPORT IS ISSUED
                        ;       LISTING THE EXPECTED DATA, ACTUAL DATA, AND THE DISCREPANCIES.
                        ;       THE ERROR REPORT ANALYZES THE TSSR CONTENTS AND DISCERNS AND
                        ;       REPORTS ONE OF THREE POSSIBILITIES:
                        ;
                        ;
                        ;            1.  TSSR CONTENTS ARE AMBIGUOUS (ANY OF BITS 11-14 ARE SET,
                        ;                OR STATES OF SSR AND SC BITS DO NOT CORRESPOND TO THE
                        ;                APPARENT ERROR CODE IN BITS 0-5):  INDICATES THAT THE
                        ;                TSSR CONTENT CANNOT BE TRUSTED.  INDICATES A
                        ;                CATASTROPHIC CONTROLLER MALFUNCTION.  THIS IS A FATAL
                        ;                ERROR (EXECUTION IS ABORTED).  FIELD ACTION WOULD BE TO
                        ;                REPLACE THE CONTROLLER.  IF THE CONTROLLER ITSELF IS BEING
                        ;                DEBUGGED, THE PROGRAM SHOULD BE RESTARTED WITH LOOP ON
                        ;                ERROR ENABLED IN ORDER TO PROBE FOR THE PROBLEM.
                        ;
                        ;            2.  SSR = 0, SC = 0 AND THE ERROR CODE IN BITS 0-5 IS IN
                        ;                THE RANGE 17-13:  THIS IS A FATAL ERROR.  THE ERROR
                        ;                CODE IS DECODED AND THE APPROPRIATE DESCRIPTION GIVEN.
                        ;                INDICATES THAT A SERIOUS PROBLEM EXISTS.
                        ;

023470                          BGNTST
023470                                                          T1::
023470  005037  002170          CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
023474  012737  005672  002146  MOV     #EPRT1,EPRTSW           ;SET UP ERROR MESSAGE SWITCH
023502  005037  003100          CLR     KTFLG           ;HOLD OFF KT11
023506  012700  023704          MOV     #TST1ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
023512  004737  017232          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
023516  012737  000005  002164  MOV     #5.,LOOPCNT             ;PERFORM 5 ITERATIONS
023524                  T1LOOP:
023524  005003                  CLR     R3                      ;USE R3 AS FATAL ERROR FLAG

023526                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
023526                                                                  T1.1:
023526  104402                                                                  TRAP    C$BSUB

023530                          BRESET                          ;ISSUE A BUS RESET
023530  104433                                                                  TRAP    C$RESET
023532  004737  016744          JSR     PC,WAITF                ;WAIT FOR READY
```

```
TEST  1: BUS RESET TEST

 4097 023536  016501  000000          MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 4098 023542  010102                  MOV     R1,R2           ;START SETUP OF EXPECTED TSSR
 4099 023544  042702  176277          BIC     #^C<HIADDR!OFL>,R2 ;CLEAR OUT UNUSED BITS
 4100 023550  052702  002200          BIS     #SSR!NBA,R2     ;R4 HAS EXPECTED CONTENTS
 4101 023554  020102                  CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
 4102 023556  001405                  BEQ     10$                     ;BRANCH IF COMPARE
 4106 023560                          ERRDF   ERRNO,SFHERR,SFFMSG     ;REPORT A FATAL ERROR
      023560  104455                                                          TRAP    C$ERDF
      023562  000145                                                          .WORD   101
      023564  003603                                                          .WORD   SFHERR
      023566  011566                                                          .WORD   SFFMSG
 4107 023570  005203                  INC     R3                      ;SET THE FATAL ERROR FLAG
 4108 023572                  10$:
 4109 023572                          ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
      023572                                                                L10037:
      023572  104403                                                          TRAP    C$ESUB
 4110
```

TEST 1: BUS RESET TEST

```
4112 023574  005703                    TST     R3                  ;DID WE HAVE FATAL ERROR ?
4113 023576  001402                    BEQ     20$                 ;BRANCH IF NOT
4114 023600  004737  017776            JSR     PC,CKDROP           ;GO DROP THIS UNIT, IF ALLOWED
4115 023604  005003              20$:  CLR     R3                  ;RESET FATAL ERROR FLAG
4116
4117
4118 023606                            BGNSUB                      ;////////////// BEGIN SUBTEST //////////////
     023606                                                                      T1.2:
     023606  104402                                                                TRAP    C$BSUB
4119
4120 023610  005065  000000            CLR     TSSR(R5)            ;WRITE TO ISSUE A SOFT RESET
4121 023614  004737  016744            JSR     PC,WAITF            ;WAIT FOR READY TO SET
4122 023620  016501  000000            MOV     TSSR(R5),R1         ;GET REGISTER TSSR DATA
4123 023624  010102                    MOV     R1,R2               ;START SETUP OF EXPECTED TSSR
4124 023626  042702  176277            BIC     #^C<HIADDR!OFL>,R2  ;CLEAR OUT UNUSED BITS
4125 023632  052702  002200            BIS     #SSR!NBA,R2         ;R4 HAS EXPECTED CONTENTS
4126 023636  020102                    CMP     R1,R2               ;COMPARE EXPECTED TO RECEIVED
4127 023640  001405                    BEQ     10$                 ;BRANCH IF COMPARE
4131 023642                            ERRDF   ERRNO,SFIERR,SFFMSG ;REPORT A FATAL ERROR
     023642  104455                                                                TRAP    C$ERDF
     023644  000146                                                                .WORD   102
     023646  003550                                                                .WORD   SFIERR
     023650  011566                                                                .WORD   SFFMSG
4132 023652  005203                    INC     R3                  ;SET THE ERROR FLAG
4133 023654                      10$:
4134 023654                            ENDSUB                      ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     023654                                                                      L10040:
     023654  104403                                                                TRAP    C$ESUB
4135
4136
4137 023656  005703                    TST     R3                  ;FATAL ERROR DETECTED ?
4138 023660  001402                    BEQ     20$                 ;BRANCH IF NOT
4139 023662  004737  017776            JSR     PC,CKDROP           ;SEE IF TIME TO DROP UNIT
4140 023666  004737  017200      20$:  JSR     PC,TSTLOOP          ;SHOULD WE DO ITERATIONS ?
4141 023672  103002                    BCC     40$                 ;BRANCH IF NOT
4142 023674  000137  023524            JMP     T1LOOP              ;LOOP UNTIL COUNT EXPIRED
4143 023700                      40$:  EXIT    TST                 ;ALL DONE THIS TEST
     023700  104432                                                                TRAP    C$EXIT
     023702  000022                                                                .WORD   L10036-.
4144
4145                                   ;+
4146                                   ;LOCAL TEXT MESSAGES FOR TEST
4147                                   ;-
4148
4149 023704     111     156     151   TST1ID: .ASCIZ  'Initialization'
4150                                           .EVEN
4151 023724                                    ENDTST
     023724                                                                L10036:
     023724  104401                                                                TRAP    C$ETST
4152
4153
```

```
                                        .SBTTL  TEST  2: RAM TEST

                                ;       THIS TEST VERIFIES THAT ALL LOCATIONS OF THE RAM ON THE CONTROLLER
                                ;       CAN PROPERLY STORE AND READ BACK ALL DATA PATTERNS, AND THAT
                                ;       EACH RAM LOCATION IS UNIQUELY ADDRESSED (I.E., THAT ONE AND ONLY
                                ;       ONE LOCATION IS ACCESSED BY ANY PARTICULAR ADDRESS).  THESE
                                ;       TESTS ARE PERFORMED BY THREE SUBTESTS, DESCRIBED BELOW.  A
                                ;       BYPRODUCT OF THESE TESTS IS A VERIFICATION OF TWO REGISTERS IN
                                ;       THE 2901 AND THE CAPABLITY OF THE 2901 TO CORRECTLY PERFORM AN
                                ;       ADD.
                                ;
                                ;
                                ;       TEST 2, SUBTEST 1: -
                                ;
                                ;       THIS SUBTEST VERIFIES EACH RAM LOCATION BY FIRST PLACING THE
                                ;       CONTROLLER INTO MAINTENANCE MODE BY WRITING INTO THE LOW BYTE OF TSDB
                                ;       AND THEN PERFORMING THE FOLLOWING SEQUENCE FOR EACH ADDRESS
                                ;       0-7777 (OCTAL):
                                ;
                                ;            1.  THE ADDRESS TO BE TESTED IS LOADED INTO THE TSDB (VIA A
                                ;                WORD WRITE).
                                ;
                                ;            2.  THE ADDRESSED RAM LOCATION IS WRITTEN, THEN READ INTO
                                ;                THE LOW BYTE OF TSBA, BY WRITING A DATA BYTE INTO THE
                                ;                LOW BYTE OF TSDB.
                                ;
                                ;            3.  THE LOW BYTE OF TSBA IS CHECKED TO SEE IF IT CONTAINS
                                ;                THE DATA PATTERN ORIGINALLY WRITTEN;  A DISCREPANCY IS
                                ;                REPORTED AS AN ERROR.
                                ;
                                ;            4.  THE ADDRESS OF THE LOCATION BEING TESTED IS AGAIN
                                ;                WRITTEN INTO TSDB (WORD WRITE), TO CAUSE THE LOCATION
                                ;                UNDER TEST TO AGAIN BE READ INTO THE LOW BYTE OF TSBA.
                                ;                THE LOW BYTE OF TSBA IS AGAIN CHECKED AND DISCREPANCIES
                                ;                REPORTED.
                                ;
                                ;            5.  THE HIGH BYTE OF TSBA IS CHECKED;  IT SHOULD CONTAIN
                                ;                THE SUM OF THE HIGH AND LOW BYTES LAST WRITTEN INTO
                                ;                TSDB AS A WORD.  A DISCREPANCY IS REPORTED AS A 2901
                                ;                PROBLEM.
                                ;
                                ;            6.  THE CONTENT OF TSSR IS CHECKED;  SETTING OF THE SC BIT
                                ;                IS IGNORED.  OTHER DISCREPANCIES IN TSSR ARE REPORTED.
                                ;

                                BGNTST
023726                                                          T2::
023726
023726  005037  002170          CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
023732  012737  005672  002146  MOV     #EPRT1,EPRTSW           ;SET UP ERROR MESSAGE SWITCH
023740  005037  003100          CLR     KTFLG           ;HOLD OFF KT11

023744                          BGNSUB                  ;////////////// BEGIN SUBTEST ////////////
023744                                                          T2.1:
023744  104402                                                          TRAP    C$BSUB
```

```
 4214 023746  012700  024672                MOV     #TST2ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
 4215 023752  004737  017232                JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
 4216 023756  012737  000002  002164        MOV     #2,LOOPCNT              ;PERFORM 2 ITERATIONS
 4217 023764                        T2LOOP:
 4218 023764  004737  016470                JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
 4219 023770  103405                        BCS     20$                     ;BR IF INIT WAS OK
 4223 023772  010001                        MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
 4224 023774                                ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
      023774  104455                                                                TRAP    C$ERDF
      023776  000311                                                                .WORD   201
      024000  003550                                                                .WORD   SFIERR
      024002  011506                                                                .WORD   SFIMSG
 4225 024004  012704  000002        20$:    MOV     #2,R4                   ;SET RAM ADDRESS AT TWO
 4226 024010                        25$:
 4227
 4228 024010  110402                        MOVB    R4,R2                   ;EXPECTED DATA FROM WRAP-AROUND
 4229 024012  110465  177777                MOVB    R4,TSDBH(R5)            ;LOAD ADDRESS INTO TSDB
 4230 024016  110265  177776                MOVB    R2,TSDBL(R5)            ;LOADS DATA INTO RAM LOCATION
 4231 024022  116501  177776                MOVB    TSBAL(R5),R1            ;READS WRAP DATA
 4232 024026  120102                        CMPB    R1,R2                   ;DOES WRITTEN(WRAP) = READ
 4233 024030  001404                        BEQ     30$                     ;BR IF OK, THEY ARE EQUAL
 4237 024032                                ERRHRD  ERRNO,TSBAM2,EXPREC     ;DATA NOT WRAPPED CORRECTLY
      024032  104456                                                                TRAP    C$ERHRD
      024034  000312                                                                .WORD   202
      024036  024530                                                                .WORD   TSBAM2
      024040  016170                                                                .WORD   EXPREC
 4238 024042                        30$:
 4239
 4240 024042  005204                        INC     R4                      ;NEXT ADDRESS
 4241 024044  020427  000400                CMP     R4,#400                 ;END OF RAM MEMORY CHECK
 4242 024050  001357                        BNE     25$                     ;LOOP TILL ALL RAM WRITTEN
 4243 024052  005002                        CLR     R2                      ;CLEAR OUT R2 HIGH BITS
 4244 024054  005304                        DEC     R4                      ;SET BACK TO 377
 4245 024056  110402                40$:    MOVB    R4,R2                   ;GET DATA PATTERN BACK IN SHAPE
 4246 024060  110465  177777                MOVB    R4,TSDBH(R5)            ;LOAD UP RAM ADDRESS POINTER
 4247 024064  116501  177776                MOVB    TSBAL(R5),R1            ;READ RAM CONTENTS BACK
 4248 024070  120102                        CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
 4249 024072  001404                        BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
 4253 024074                                ERRHRD  ERRNO,TSBAM2,EXPREC     ;WRITTEN DATA NOT = TO READ
      024074  104456                                                                TRAP    C$ERHRD
      024076  000313                                                                .WORD   203
      024100  024530                                                                .WORD   TSBAM2
      024102  016170                                                                .WORD   EXPREC
 4254 024104                        45$:    CKLOOP                          ;SCOPE LOOP
      024104  104406                                                                TRAP    C$CLP1
 4255 024106  005304                        DEC     R4                      ;DROP DATA COUNTER (PATTERN)
 4256 024110  022704  000002                CMP     #2,R4                   ;AT LOC TWO YET
 4257 024114  001360                        BNE     40$                     ;BR, IF NOT AT TWO YET
 4258
 4259 024116                                ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
      024116                                                                L10042:
      024116  104403                                                                TRAP    C$ESUB
 4260
```

```
4262
4263 024120                            BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     024120                                                                            T2.2:
     024120  104402                                                                      TRAP    C$BSUB
4264                            ;       TEST 2, SUBTEST 2
4265
4266                            ;
4267                            ;       THIS SUBTFST WRITES RAM WITH ALL ZEROS
4268                            ;       THEN WALKS AN ALL ONES WORD DOWN THROUGH MEMORY
4269                            ;
4270 024122  004737  016470             JSR     PC,SOFINIT              ;DO INITIALIZE ON CONTROLLER
4271 024126  103405                     BCS     20$                     ;BR IF INIT WAS OK
4275 024130  010001                     MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
4276 024132                             ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
     024132  104455                                                                      TRAP    C$ERDF
     024134  000314                                                                      .WORD   204
     024136  003550                                                                      .WORD   SFIERR
     024140  011506                                                                      .WORD   SFIMSG
4277 024142  005002             20$:    CLR     R2                      ;TEST DATA = 0
4278 024144  012704  000002             MOV     #2,R4                   ;STARTING RAM ADDRESS = 2
4279 024150                     25$:
4280
4281 024150  110465  177777             MOVB    R4,TSDBH(R5)            ;LOAD ADDRESS INTO TSDB
4282 024154  110265  177776             MOVB    R2,TSDBL(R5)            ;LOADS DATA INTO RAM LOCATION
4283 024160  116501  177776             MOVB    TSBAL(R5),R1            ;READS WRAP DATA
4284 024164  120102                     CMPB    R1,R2                   ;DOES WRITTEN(WRAP) = READ ?
4285 024166  001404                     BEQ     30$                     ;BR IF OK, THEY ARE EQUAL
4289 024170                             ERRHRD  ERRNO,TSBAM2,EXPREC     ;DATA NOT WRAPPED CORRECTLY
     024170  104456                                                                      TRAP    C$ERHRD
     024172  000315                                                                      .WORD   205
     024174  024530                                                                      .WORD   TSBAM2
     024176  016170                                                                      .WORD   EXPREC
4290 024200                     30$:
4291
4292 024200  005204                     INC     R4                      ;NEXT ADDRESS
4293 024202  020427  000400             CMP     R4,#400                 ;END OF RAM MEMORY CHECK
4294 024206  001360                     BNE     25$                     ;BR, MORE RAM TO GO
4295
4296 024210  005304             35$:    DEC     R4                      ;SET BACK TO 377
4297 024212  005002                     CLR     R2                      ;SET TO ALL ZEROS
4298 024214                     40$:
4299 024214  110465  177777             MOVB    R4,TSDBH(R5)            ;LOAD UP THE ADDRESS FOR RAM
4300 024220  116501  177776             MOVB    TSBAL(R5),R1            ;READ THE RAM CONTENTS BACK
4301 024224  005002                     CLR     R2                      ;LOOKING FOR 000000 (EXPECTED)
4302 024226  120102                     CMPB    R1,R2                   ;BOTH SHOULD BE 00000000 BINARY
4303 024230  001404                     BEQ     43$                     ;BR, IF DATA IS GOOD
4307 024232                             ERRHRD  ERRNO,TSBAM3,EXPREC     ;CHARACTERISTICS DATA NOT CORRECT
     024232  104456                                                                      TRAP    C$ERHRD
     024234  000316                                                                      .WORD   206
     024236  024612                                                                      .WORD   TSBAM3
     024240  016170                                                                      .WORD   EXPREC
4308 024242  012702  000377     43$:    MOV     #000377,R2              ;SET ALL ONES WORD
4309 024246  110465  177777             MOVB    R4,TSDBH(R5)            ;LOAD UP RAM ADDRESS POINTER
4310 024252  110265  177776             MOVB    R2,TSDBL(R5)            ;WRITE DATA INTO RAM
4311 024256  116501  177776             MOVB    TSBAL(R5),R1            ;READ RAM CONTENTS BACK
4312 024262  120102                     CMPB    R1,R2                   ;CHECK WITH DATA WRITTEN
4313 024264  001404                     BEQ     45$                     ;BR IF OK, DATA IN = DATA OUT
```

```
 4317 024266                                ERRHRD  ERRNO,TSBAM2,EXPREC     ;WRITTEN DATA NOT = TO READ
      024266  104456                                                                TRAP    C$ERHRD
      024270  000317                                                                .WORD   207
      024272  024530                                                                .WORD   TSBAM2
      024274  016170                                                                .WORD   EXPREC
 4318 024276                          45$:    CKLOOP                          ;SCOPE LOOP
      024276  104406                                                                TRAP    C$CLP1
 4319 024300  005304                         DEC     R4                      ;DROP RAM ADDRESS POINTER
 4320 024302  022704  000002                CMP     #2,R4                   ;AT LOC 2 YET
 4321 024306  001342                         BNE     40$                     ;BR, IF NOT AT TWO YET
 4322
 4323 024310                                 ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
      024310                                                                L10043:
      024310  104403                                                                TRAP    C$ESUB
 4324
```

```
 4326
 4327 024312                                BGNSUB                  ;////////////// BEGIN SUBTEST //////////////
      024312                                                                       T2.3:
      024312  104402                                                                    TRAP    C$BSUB
 4328
 4329                               ;       TEST 2, SUBTEST 3
 4330                               ;
 4331                               ;       THIS SUBTEST WRITES RAM WITH ALL ONES
 4332                               ;       THEN WALKS A ZERO WORD DOWN THROUGH MEMORY
 4333                               ;
 4334 024314  004737  016470                JSR     PC,$OFINIT              ;DO INITIALIZE ON CONTROLLER
 4335 024320  103405                        BCS     20$                     ;BR IF INIT WAS OK
 4339 024322  010001                        MOV     R0,R1                   ;CONTENTS OF TSSR REGISTER
 4340 024324                                ERRDF   ERRNO,SFIERR,SFIMSG     ;FATAL ERROR TSSR WAS NOT OK
      024324  104455                                                                   TRAP    C$ERDF
      024326  000320                                                                   .WORD   208
      024330  003550                                                                   .WORD   SFIERR
      024332  011506                                                                   .WORD   SFIMSG
 4341 024334  012702  177777        20$:    MOV     #177777,R2              ;SET DATA AT ALL ONES
 4342 024340  012704  000002                MOV     #2,R4                   ;SET RAM ADDRESS AT TWO
 4343 024344                        25$:
 4344 024344                                BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      024344  104404                                                                   TRAP    C$BSEG
 4345 024346  110465  177777                MOVB    R4,TSDBH(R5)            ;LOAD ADDRESS INTO TSDB
 4346 024352  110265  177776                MOVB    R2,TSDBL(R5)            ;LOADS DATA INTO RAM LOCATION
 4347 024356  116501  177776                MOVB    TSBAL(R5),R1            ;READS WRAP DATA
 4348 024362  120102                        CMPB    R1,R2                   ;DOES WRITTEN(WRAP) = READ ?
 4349 024364  001404                        BEQ     30$                     ;BR IF OK, THEY ARE EQUAL
 4353 024366                                ERRHRD  ERRNO,TSBAM2,EXPREC     ;DATA NOT WRAPPED CORRECTLY
      024366  104456                                                                   TRAP    C$ERHRD
      024370  000321                                                                   .WORD   209
      024372  024530                                                                   .WORD   TSBAM2
      024374  016170                                                                   .WORD   EXPREC
 4354 024376                        30$:
 4355 024376                                ENDSEG                          ;<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<
      024376                                                                       10000$:
      024376  104405                                                                    TRAP    C$ESEG
 4356
 4357 024400  005204                        INC     R4                      ;NEXT ADDRESS
 4358 024402  020427  000400                CMP     R4,#400                 ;END OF RAM MEMORY CHECK
 4359 024406  001356                        BNE     25$                     ;BR, MORE RAM TO GO
 4360
 4361 024410  005304                35$:    DEC     R4                      ;SET BACK TO 377
 4362 024412                        40$:
 4363 024412  012702  000377                MOV     #000377,R2              ;SET UP EXPECTED DATA REGISTER
 4364 024416  005001                        CLR     R1                      ;CLEAN OUT REGISTER
 4365 024420  110465  177777                MOVB    R4,TSDBH(R5)            ;SELECT ADDRESS IN RAM
 4366 024424  116501  177776                MOVB    TSBAL(R5),R1            ;PICK UP RAM CONTENTS
 4367 024430  120102                        CMPB    R1,R2                   ;IS MEMORY STILL ALL ONES
 4368 024432  001404                        BEQ     43$                     ;BR, IF OK (ALL ONES)
 4372 024434                                ERRHRD  ERRNO,TSBAM3,EXPREC     ;MEMORY CHANGED AFTER ALL ONES WRITE
      024434  104456                                                                   TRAP    C$ERHRD
      024436  000322                                                                   .WORD   210
      024440  024612                                                                   .WORD   TSBAM3
      024442  016170                                                                   .WORD   EXPREC
 4373 024444  005002                43$:    CLR     R2                      ;SET UP NEW EXPECTED
 4374 024446  110465  177777                MOVB    R4,TSDBH(R5)            ;LOAD UP RAM ADDRESS POINTER
```

```
4375 024452  116265  177776            MOVB    R2,TSDBL(R5)          ;WRITE DATA INTO RAM
4376 024456  116501  177776            MOVB    TSBAL(R5),R1          ;READ RAM CONTENTS BACK
4377 024462  120102                    CMPB    R1,R2                 ;CHECK WITH DATA WRITTEN
4378 024464  001404                    BEQ     45$                   ;BR IF OK, DATA IN = DATA OUT
4382 024466                            ERRHRD  ERRNO,TSBAM2,EXPREC   ;WRITTEN DATA NOT = TO READ
     024466  104456                                                          TRAP    C$ERHRD
     024470  000323                                                          .WORD   211
     024472  024530                                                          .WORD   TSBAM2
     024474  016170                                                          .WORD   EXPREC
4383 024476                    45$:    CKLOOP                        ;SCOPE LOOP
     024476  104406                                                          TRAP    C$CLP1
4384 024500  005304                    DEC     R4                    ;DROP RAM ADDRESS POINTER
4385 024502  022704  000002            CMP     #2,R4                 ;CHECK LOC TWO
4386 024506  001341                    BNE     40$                   ;BR, IF NOT AT LOC 2 YET
4387
4388 024510                            ENDSUB                        ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     024510                                                                  L10044:
     024510  104403                                                          TRAP    C$ESUB
4389
4390 024512  004737  017200            JSR     PC,TSTLOOP            ;DO WE NEED TO ITERATE TEST ?
4391 024516  103002                    BCC     63$                   ;BRANCH IF NOT
4392 024520  000137  023764            JMP     T2LOOP                ;EXECUTE AGAIN
4393 024524                    63$:    EXIT    TST                   ;ALL DONE THIS TEST
     024524  104432                                                          TRAP    C$EXIT
     024526  000150                                                          .WORD   L10041-.
4394                           ;+
4395                           ;LOCAL TEXT MESSAGES FOR TEST
4396                           ;-
4397
4398 024530    040    127    162  TSBAM2: .ASCIZ  ' Write to TSDB Not Equal to Read of TSBA Low Byte'
4399 024612    127    162    151  TSBAM3: .ASCIZ  'Write To RAM Location Modified Another Location'
4400 024672    122    141    155  TST2ID: .ASCIZ  'Ram'
4401                                  .EVEN
4402 024676                           ENDTST
     024676                                                  L10041:
     024676  104401                                                  TRAP    C$ETST
```

```
                        .SBTTL  TEST  3: COMMAND REJECT


                ;       THIS TEST VERIFIES THAT ALL COMMANDS OTHER THAN WRITE
                ;       CHARACTERISTICS ARE REJECTED DUE TO THE NEED BUFFER ADDRESS
                ;       (NBA) BIT BEING SET IN TSSR, AND THAT THE TSBA AND TSSR
                ;       REGISTERS ARE LEFT IN THE PROPER STATE AFTER EACH COMMAND IS
                ;       REJECTED.  THIS TEST CHECKS MICROPROCESSOR SEQUENCING, BASIC
                ;       COMMAND DECODING AND DATI DMA HANDLING.  THIS TEST CONTAINS TWO
                ;       SUBTESTS:  SUBTEST 1 SEQUENCES THROUGH ALL COMMAND WORDS (OTHER
                ;       THAN WRITE CHARACTERISTICS) WITH THE INTERRUPT ENABLE (IE) BIT
                ;       CLEAR AND VERIFIES THAT AN INTERRUPT IS NOT GENERATED BY THE
                ;       REJECTED COMMAND;  SUBTEST 2 PERFORMS SIMILARLY TO SUBTEST 1 BUT
                ;       SETS THE IE BIT IN EACH COMMAND WORD AND VERIFIES THAT AN
                ;       INTERRUPT IS GENERATED WHEN THE COMMAND IS REJECTED.  SUBTEST 1
                ;       SETS UP THE INTERRUPT SERVICE ROUTINE TO FLAG UNEXPECTED
                ;       INTERRUPTS.  THE COMMAND WORD IN THE COMMAND BUFFER IS
                ;       INITIALIZED TO 100000 (OCTAL) AND THE REMAINING THREE WORDS IN
                ;       THE COMMAND BUFFER ARE SET TO KNOWN UNIQUE PATTERNS.  THEN THE
                ;       FOLLOWING SEQUENCE IS PERFORMED:
                ;
                ;
                ;            1.  INITIALIZE THE CONTROLLER BY WRITING INTO THE TSSR;
                ;                PROPER INITIAL CONDITIONS ARE VERIFIED.
                ;
                ;            2.  TSDB IS WRITTEN WITH ADDRESS OF THE COMMAND BUFFER TO
                ;                START PROCESSING.
                ;
                ;            3.  THE PROGRAM WAITS FOR SSR TO SET;  IF SSR DOES NOT SET,
                ;                AN ERROR REPORT IS ISSUED AND THE TEST IS ABORTED.
                ;
                ;            4.  THE CONTENTS OF TSSR ARE CHECKED.  TSSR IS CORRECT IF
                ;                IT CONTAINS EITHER OCTAL 102206 OR 102306 (BIT 6
                ;                DEPENDS UPON THE STATE OF THE TAPE TRANSPORT).
                ;
                ;            5.  THE CONTENTS OF TSBA ARE CHECKED.  TSBA SHOULD CONTAIN
                ;                THE INITIAL COMMAND BUFFER ADDRESS (LOADED IN STEP 2)
                ;                PLUS 10 (OCTAL);  I.E., TSBA SHOULD POINT TO THE WORD
                ;                JUST AFTER THE COMMAND PACKET (NOTE THAT 4 COMMAND
                ;                PACKET WORDS ARE ALWAYS FETCHED).
                ;
                ;            6.  USING THE MAINTENANCE MODE WRAPAROUND FUNCTIONS, THE
                ;                COMMAND IMAGE BLOCK IN THE CONTROLLER'S RAM (LOCATIONS
                ;                201-210 (OCTAL)) ARE CHECKED;  THE IMAGE SHOULD CONTAIN
                ;                A COPY OF THE FOUR COMMAND PACKET WORDS AS SET UP IN
                ;                CPU MEMORY.
                ;
                ;            7.  THE COMMAND WORD IN THE COMMAND BUFFER IS INCREMENTED
                ;                TO THE NEXT PATTERN NOT CONTAINING WRITE
                ;                CHARACTERISTICS OR IE.  THE REMAINING THREE WORD OF THE
                ;                COMMAND BUFFER ARE SEQUENCED WITH PSEUDO-RANDOM DATA.
                ;                IF THE COMMAND WORD HAS NOT REACHED ITS MAXIMUM VALUE
                ;                (177777+1), THE TEST SEQUENCE IS REPEATED.
                ;
                ;
                ;
                ;       SUBTEST 2 IS IDENTICAL TO SUBTEST 1, EXCEPT THAT THE PROGRAM
```

```
                                        ;       CAUSES THE IE BIT TO BE SET IN EACH COMMAND WORD AND THEN
                                        ;       VERIFIES THAT AN INTERRUPT OCCURS.
                                        ;
                                        BGNTST
                                                                                T3::
                                        CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
                                        MOV     #EPRT1,EPRTSW           ;SET UP ERROR MESSAGE SWITCH
                                        CLR     KTFLG           ;HOLD OFF KT11
                                        MOV     #TST3ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
                                        JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
                                        MOV     #2.,LOOPCNT             ;PERFORM 2 ITERATIONS
                                T3LOOF:
                                        BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
                                                                                T3.1:
                                                                                TRAP    C$BSUB

                                        SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
                                                                                MOV     #PRI00,R0
                                                                                TRAP    C$SPRI
                                        MOV     #T3PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
                                        MOV     #TSTBLK,R3              ;BLOCK OF TEST DATA
                                5$:     MOV     (R3)+,(R4)              ;INSERT THE NEXT TEST DATA WORD
                                        BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
                                                                                TRAP    C$BSEG
                                        JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
                                        BCS     10$                     ;BR IF SOFT INIT = OK
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
                                                                                TRAP    C$ERDF
                                                                                .WORD   301
                                                                                .WORD   SFIERR
                                                                                .WORD   SFIMSG
                                10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
                                        CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
                                        JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
                                        BIC     #BIT7,(R4)              ;DISABLE INTERRUPTS
                                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
                                        JSR     PC,WAITF                ;WAIT FOR SSR TO SET
                                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSP
                                        ERRDF   ERRNO,T3SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                                                                TRAP    C$ERDF
                                                                                .WORD   302
                                                                                .WORD   T3SSR
                                                                                .WORD   PKTSSR
                                        JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
                                15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                TRAP    C$CLP1
                                        ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
                                                                                TRAP    C$ESCAPE
                                                                                .WORD   L10046-.
                                        TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
                                        BEQ     22$                     ;BRANCH IF NOT
                                        ERRHRD  ERRNO,T3INT,PKTSSR
                                                                                TRAP    C$ERHRD
                                                                                .WORD   303
                                                                                .WORD   T3INT
```

```
                025072  011520                                                          .WORD   PKTSSR
4511            025074  012702  102206          22$:    MOV     @SC!NBA!SSR!TSREJ,R2    ;EXPECTED CONTENTS OF TSSR
4512            025100  004737  017060                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
4513            025104  016501  000000                  MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
4514            025110  032701  000100                  BIT     @OFL,R1                 ;IS OFF-LINE BIT SET ?
4515            025114  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
4516            025116  052702  000100                  BIS     @OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
4517            025122  020201                  25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
4518            025124  001404                          BEQ     30$                     ;DOES EXPECTED MATCH RECEIVED?
4522            025126                                  ERRHRD  ERRNO,T3NBA,PKTSSR      ;NBA NOT SET TO REJECT
                025126  104456                                                          TRAP    C$ERHRD
                025130  000460                                                          .WORD   304
                025132  025630                                                          .WORD   T3NBA
                025134  011520                                                          .WORD   PKTSSR
4523            025136                  30$:    CKLOOP                                  ;LOOP ON ERROR ?
                025136  104406                                                          TRAP    C$CLP1
4524            025140  004737  017060                  JSR     PC,CHKTSSR              ;WAIT FOR READY, NON-AMBIGUOUS
4525            025144  016501  177776                  MOV     TSBA(R5),R1             ;GET TSBA REGISTER CONTENTS
4526            025150  010402                          MOV     R4,R2                   ;START OF THE PACKET
4527            025152  020102                          CMP     R1,R2                   ;COMPARE EXPECTED TO RECEIVED
4528            025154  001404                          BEQ     35$                     ;ERROR IF NOT EQUAL
4532            025156                                  ERRHRD  ERRNO,T3TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
                025156  104456                                                          TRAP    C$ERHRD
                025160  000461                                                          .WORD   305
                025162  026071                                                          .WORD   T3TSBA
                025164  016170                                                          .WORD   EXPREC
4533
4534
4535            025166  004737  010354          35$:    JSR     PC,CKRAM                ;SEE IF DATA IN RAM IS CORRECT
4536            025172  103404                          BCS     40$                     ;BRANCH IF PACKET IN RAM IS CORRECT
4540            025174                                  ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
                025174  104456                                                          TRAP    C$ERHRD
                025176  000462                                                          .WORD   306
                025200  004643                                                          .WORD   PKTRAM
                025202  016204                                                          .WORD   RAMERR
4541            025204                  40$:    ENDSEG                                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
                025204                                                                  10000$:
                025204  104405                                                          TRAP    C$ESEG
4542            025206  011300                          MOV     (R3),R0                 ;PACKET COMMAND WORD
4543            025210  042700  177740                  BIC     #177740,R0              ;GET BITS 0-4
4544            025214  020027  000004                  CMP     R0,#4                   ;DON'T TEST WRITE CHARACTERISTICS
4545            025220  001002                          BNE     45$                     ;BRANCH IF OK
4546            025222  062703  000002                  ADD     #2,R3                   ;GET NEXT WORD FROM DATA TABLE
4547            025226  020327  003030          45$:    CMP     R3,#TBLEND              ;REACHED END OF TABLE ?
4548            025232  103002                          BHIS    50$                     ;BRANCH IF END OF TABLE
4549            025234  000137  024754                  JMP     5$                      ;CONTINUE TEST WITH NEW DATA
4550
4551            025240                  50$:    ENDSUB                                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
                025240                                                                  L10046:
                025240  104403                                                          TRAP    C$ESUB
4552
4553            025242  005737  002170                  TST     FATFLG                  ;ANY FATAL ERRORS ?
4554            025246  001402                          BEQ     60$                     ;BRANCH IF NOT
4555            025250  004737  017776                  JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
```

```
 4557 025254                    60$:    BGNSUB                  ;////////////// BEGIN SUBTEST //////////////
      025254                                                                          T3.2:
      025254  104402                                                                   TRAP    C$BSUB
 4558
 4559 025256                            SETPRI  #PRI00          ;LOWER PRIORITY TO ALLOW INTERRUPTS
      025256  012700  000000                                                           MOV     #PRI00,R0
      025262  104441                                                                   TRAP    C$SPRI
 4560 025264  012704  025620            MOV     #T3PACKET,R4    ;GET THE ADDRESS OF COMMAND PACKET
 4561 025270  012703  002720            MOV     #TSTBLK,R3      ;START OF TEST DATA
 4562 025274  012314            5$:     MOV     (R3)+,(R4)      ;PLACE NEXT DATA WORD IN PACKET
 4563 025276                            BGNSEG                  ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      025276  104404                                                                   TRAP    C$BSEG
 4564 025300  004737  016470            JSR     PC,SOFINIT      ;DO SOFT INIT OF CONTROLLER
 4565 025304  103405                    BCS     10$             ;BR IF SOFT INIT = OK
 4569 025306  010001                    MOV     R0,R1           ;SAVE CONTENTS OF TSSR
 4570 025310                            ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      025310  104455                                                                   TRAP    C$ERDF
      025312  000463                                                                   .WORD   307
      025314  003550                                                                   .WORD   SFIERR
      025316  011506                                                                   .WORD   SFIMSG
 4571 025320  005037  002170    10$:    CLR     FATFLG          ;CLEAR FATAL ERROR FLAG
 4572 025324  005037  002172            CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
 4573 025330  004737  017060            JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
 4574 025334  052714  000200            BIS     #BIT7,(R4)      ;ENABLE INTERRUPTS
 4575 025340  010465  177776            MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
 4576 025344  004737  016744            JSR     PC,WAITF        ;WAIT FOR SSR TO SET
 4577 025350  103407                    BCS     15$             ;BR IF CARRY SET (GOOD RETURN)
 4578 025352  010001                    MOV     R0,R1           ;SAVE CONTENTS OF TSSR
 4582 025354                            ERRDF   ERRNO,T3SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      025354  104455                                                                   TRAP    C$ERDF
      025356  000464                                                                   .WORD   308
      025360  025655                                                                   .WORD   T3SSR
      025362  011520                                                                   .WORD   PKTSSR
 4583 025364  004737  017724            JSR     PC,FATCHK       ;INC AND CHECK FOR MORE THAN 25 ERRORS
 4584 025370                    15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
      025370  104406                                                                   TRAP    C$CLP1
 4585 025372                            ESCAPE  SUB             ;BY-PASS SUBTEST IF FATAL ERROR
      025372  104410                                                                   TRAP    C$ESCAPE
      025374  000164                                                                   .WORD   L10047 .
 4586 025376  005737  002172            TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
 4587 025402  001004                    BNE     22$             ;BRANCH IF YES
 4591 025404                            ERRHRD  ERRNO,T3NINT,PKTSSR     ;REPORT ERROR IF NO INTERRUPT
      025404  104456                                                                   TRAP    C$ERHRD
      025406  000465                                                                   .WORD   309
      025410  026011                                                                   .WORD   T3NINT
      025412  011520                                                                   .WORD   PKTSSR
 4592 025414  012702  102206    22$:    MOV     #SC!NBA!SSR!TSREJ,R2    ;EXPECTED CONTENTS OF TSSR
 4593 025420  004737  017060            JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
 4594 025424  016501  000000            MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
 4595 025430  032701  000100            BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
 4596 025434  001402                    BEQ     25$             ;BRANCH IF NOT OFF-LINE
 4597 025436  052702  000100            BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
 4598 025442  020201    25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
 4599 025444  001404                    BEQ     30$             ;OKAY IF MATCH
 4603 025446                            ERRHRD  ERRNO,T3NBA,PKTSSR      ;NBA NOT SET TO REJECT
      025446  104456                                                                   TRAP    C$ERHRD
      025450  000466                                                                   .WORD   310
```

```
      025452  025630                                                          .WORD   T3NBA
      025454  011520                                                          .WORD   PKTSSR
4604  025456                          30$:    CKLOOP                  ;LOOP ON ERROR ?
      025456  104406                                                          TRAP    C$CLP1
4605  025460  004737  017060                  JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
4606  025464  016501  177776                  MOV     TSBA(R5),R1     ;GET TSBA REGISTER CONTENTS
4607  025470  010402                          MOV     R4,R2           ;START OF THE PACKET
4608  025472  020102                          CMP     R1,R2           ;COMPARE EXPECTED TO RECEIVED
4609  025474  001404                          BEQ     35$             ;ERROR IF NOT EQUAL
4613  025476                                  ERRHRD  ERRNO,T3TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
      025476  104456                                                          TRAP    C$ERHRD
      025500  000467                                                          .WORD   311
      025502  026071                                                          .WORD   T3TSBA
      025504  016170                                                          .WORD   EXPREC
4614
4615
4616  025506  004737  010354          35$:    JSR     PC,CKRAM        ;SEE IF DATA IN RAM IS CORRECT
4617  025512  103404                          BCS     40$             ;BRANCH IF PACKET IN RAM IS CORRECT
4621  025514                                  ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
      025514  104456                                                          TRAP    C$ERHRD
      025516  000470                                                          .WORD   312
      025520  004643                                                          .WORD   PKTRAM
      025522  016204                                                          .WORD   RAMERR
4622  025524                          40$:    ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      025524                                                                10000$:
      025524  104405                                                          TRAP    C$ESEG
4623  025526  011300                          MOV     (R3),R0         ;NEXT PACKET COMMAND WORD
4624  025530  042700  177740                  BIC     #177740,R0      ;GET BITS 0-4
4625  025534  020027  000004                  CMP     R0,#4           ;DON'T TEST WRITE CHARACTERISTICS
4626  025540  001002                          BNE     45$             ;BRANCH IF NOT WRITE CHARACTERISTICS
4627  025542  062703  000002                  ADD     #2,R3           ;BY-PASS WRITE CHARACTERISTCS
4628  025546  020327  003030          45$:    CMP     R3,#TBLEND      ;HAVE WE COMPLETED DATA TABLE ?
4629  025552  103002                          BHIS    50$             ;BRANCH IF ALL TESTED
4630  025554  000137  025274                  JMP     5$              ;TEST WITH NEXT DATA
4631
4632  025560                          50$:    ENDSUB                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
      025560                                                                L10047:
      025560  104403                                                          TRAP    C$ESUB
4633  025562  005737  002170                  TST     FATFLG          ;ANY FATAL ERRORS ?
4634  025566  001402                          BEQ     60$             ;BRANCH IF NOT
4635  025570  004737  017776                  JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
4636  025574  004737  017200          60$:    JSR     PC,TSTLOOP      ;SHOULD WE DO ITERATIONS ?
4637  025600  103002                          BCC     62$             ;BRANCH IF NOT
4638  025602  000137  024734                  JMP     T3LOOP          ;LOOP UNTIL COUNT EXPIRED
4639  025606                          62$:    EXIT    TST             ;ALL DONE THIS TEST
      025606  104432                                                          TRAP    C$EXIT
      025610  000352                                                          .WORD   L10045-.
4640
4641                                  ;+
4642                                  ;LOCAL STORAGE FOR THIS TEST
4643                                  ;-
4644
4646  025612                                  .BLKB   10-<.-TUV2A&7>
4648  025620                          T3PACKET:                       ;COMMAND PACKET FOR TEST
4649  025620  000000                          .WORD   0               ;WILL CONTAIN VARIABLE COMMANDS
4650  025622  052525                          .WORD   052525
4651  025624  125252                          .WORD   125252
```

```
4652 025626  052525                        .WORD   052525
4653
4654
4655                                ;+
4656                                ;LOCAL TEXT MESSAGES FOR TEST
4657                                ;-
4658
4659 025630     103     157     155  T3NBA:  .ASCIZ  'Command Not Rejected'
4660 025655     103     157     156  T3SSR:  .ASCIZ  'Contents of TSSR Incorrect After Write Packet'
4661 025733     125     156     145  T3INT:  .ASCIZ  'Unexpected Interrupt Received On Write Packet'
4662 026011     105     170     160  T3NINT: .ASCIZ  'Expected Interrupt Not Received On Write Packet'
4663 026071     111     156     143  T3TSBA: .ASCIZ  'Incorrect TSBA Address After Packet Write'
4664 026143     103     157     155  TST3ID: .ASCIZ  'Command Reject'
4665                                .EVEN
4666 026162                        ENDTST
     026162
     026162  104401                                              L10045:
                                                                        TRAP    C$ETST
```

```
                                .SBTTL  TEST  4: WRITE CHARACTERISTICS


                        ;       THIS TEST VERIFIES BASIC OPERATION OF THE WRITE CHARACTERISTICS
                        ;       COMMAND.  IT VERIFIES THAT THE COMMAND BLOCK AND CHARACTERISTICS
                        ;       DATA BLOCK ARE FETCHED PROPERLY FROM CPU MEMORY, THE NEED BUFFER
                        ;       ADDRESS (NBA) BIT IN TSSR IS HANDLED PROPERLY, AND THAT A PROPER
                        ;       MESSAGE PACKET IS STORED, WHERE APPROPRIATE.  THIS TEST DOES NOT
                        ;       CHECK THAT THE VARIOUS FUNCTIONS ENABLED BY CHARACTERISTIC MODE
                        ;       DATA BITS OPERATE PROPERLY;  THE FUNCTIONING OF THESE BITS IS
                        ;       VERIFIED IN SUBSEQUENT TESTS.  ALL COMMANDS EXECUTED IN THIS
                        ;       TEST HAVE THE INTERRUPT ENABLE (IE) BIT CLEARED TO ZERO, SO NO
                        ;       INTERRUPTS SHOULD BE GENERATED.  HOWEVER, THE PROGRAM RUNS AT
                        ;       PROCESSOR PRIORITY 0, WITH THE INTERRUPT SERVICE ROUTINE SET UP
                        ;       TO FLAG UNEXPECTED INTERRUPTS.  IF AN INTERRUPT OCCURS, A
                        ;       PROBLEM EXISTS IN EITHER THE LSI-11 BUS INTERFACE SECTION OR IN
                        ;       THE ROM OR PIPELINE.
                        ;
                        ;       THIS TEST CHECKS VARIOUS MICROPROGRAM SEQUENCES, COMMAND
                        ;       DECODING, DMA LOGIC, AND BASIC PACKET PROTOCOL HANDLING.  THIS
                        ;       IS THE FIRST TEST IN WHICH DATO DMA CYCLES (FOR STORING THE
                        ;       MESSAGE PACKET) ARE PERFORMED.  ANY ERRORS IN THE BODY OF THE
                        ;       TEST (I.E, ERRORS OTHER THAN INITIALIZATION ERRORS RELATED TO
                        ;       THE TRANSPORT BUS) DEFINITELY INDICATE A BAD CONTROLLER MODULE.
                        ;
                        ;
                        ;
                                BGNTST
                                                                                T4::
4697 026164  005037  002170             CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
4698 026170  012737  005672  002146     MOV     #EPRT1,EPRTSW           ;SET UP ERROR MESSAGE SWITCH
4699 026176  005037  003100             CLR     KTFLG           ;HOLD OFF KT11
4704 026202  012700  030117             MOV     #TST4ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
4705 026206  004737  017232             JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
4706 026212  012737  000002  002164     MOV     #2.,LOOPCNT             ;PERFORM 2 ITERATIONS
4707 026220                     T4LOOP:
4708 026220                             BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
     026220                                                                     T4.1:
     026220  104402                                                             TRAP    C$BSUB
4709 026222  004737  030146             JSR     PC,T4REST               ;SET PACKET TO START-UP VALUES
4710
4711 026226                             SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     026226  012700  000000                                                     MOV     #PRI00,R0
     026232  104441                                                             TRAP    C$SPRI
4712 026234  012703  002732             MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
4713 026240  012704  027250             MOV     #T4PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
4714 026244  012764  000010  000006     MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
4715 026252                     5$:
4716 026252                             BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     026252  104404                                                             TRAP    C$BSEG
4717
4718 026254  004737  016470             JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
4719 026260  103405                     BCS     10$                     ;BR IF SOFT INIT = OK
4723 026262  010001                     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
4724 026264                             ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     026264  104455                                                             TRAP    C$ERDF
```

```
                026266  000621                                                          .WORD   401
                026270  003550                                                          .WORD   SFIERR
                026272  011506                                                          .WORD   SFIMSG
4725            026274  005037  002170          10$:    CLR     FATFLG          ;CLEAR FATAL ERROR FLAG
4726            026300  005037  002172                  CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
4727            026304  010465  177776                  MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
4728            026310  004737  017060                  JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
4729            026314  103407                          BCS     15$             ;BR IF CARRY SET (GOOD RETURN)
4730            026316  010001                          MOV     R0,R1           ;SAVE CONTENTS OF TSSR
4734            026320                                  ERRDF   ERRNO,T4SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                026320  104455                                                          TRAP    C$ERDF
                026322  000622                                                          .WORD   402
                026324  027656                                                          .WORD   T4SSR
                026326  011520                                                          .WORD   PKTSSR
4735            026330  004737  017724                  JSR     PC,FATCHK       ;INC AND CHECK FOR MORE THAN 25 ERRORS
4736            026334                          15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
                026334  104406                                                          TRAP    C$CLP1
4737            026336                                  ESCAPE  SEG             ;BY-PASS SUBTEST IF FATAL ERROR
                026336  104410                                                          TRAP    C$ESCAPE
                026340  000126                                                          .WORD   10000$-.
4738            026342  005737  002172                  TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
4739            026346  001404                          BEQ     22$             ;BRANCH IF NOT
4743            026350                                  ERRHRD  ERRNO,T4INT,PKTSSR
                026350  104456                                                          TRAP    C$ERHRD
                026352  000623                                                          .WORD   403
                026354  027745                                                          .WORD   T4INT
                026356  011520                                                          .WORD   PKTSSR
4744            026360  016501  000000          22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
4745            026364  012702  000200                  MOV     #SSR,R2         ;EXPECTED CONTENTS OF TSSR
4746            026370  032701  000100                  BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
4747            026374  001402                          BEQ     25$             ;BRANCH IF NOT OFF-LINE
4748            026376  052702  000100                  BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
4749            026402  020201                  25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
4750            026404  001404                          BEQ     30$             ;OKAY IF MATCH
4754            026406                                  ERRHRD  ERRNO,T4NBA,PKTSSR      ;NBA NOT ZERO
                026406  104456                                                          TRAP    C$ERHRD
                026410  000624                                                          .WORD   404
                026412  027406                                                          .WORD   T4NBA
                026414  011520                                                          .WORD   PKTSSR
4755            026416                          30$:    CKLOOP                  ;LOOP ON ERROR ?
                026416  104406                                                          TRAP    C$CLP1
4756            026420  004737  017060                  JSR     PC,CHKTSSR      ;WAIT FOR READY, NON-AMBIGUOUS
4757            026424  016501  177776                  MOV     TSBA(R5),R1     ;GET TSBA REGISTER CONTENTS
4758            026430  012702  027250                  MOV     #T4PACKET,R2    ;START OF THE BUFFER
4759            026434  020102                          CMP     R1,R2           ;COMPARE EXPECTED TO RECEIVED
4760            026436  001404                          BEQ     35$             ;ERROR IF NOT EQUAL
4764            026440                                  ERRHRD  ERRNO,T4TSBA,EXPREC     ;PRINT THE ERROR & EXPD/RECV
                026440  104456                                                          TRAP    C$ERHRD
                026442  000625                                                          .WORD   405
                026444  030034                                                          .WORD   T4TSBA
                026446  016170                                                          .WORD   EXPREC
4765
4766
4767            026450  004737  010354          35$:    JSR     PC,CKRAM        ;SEE IF DATA IN RAM IS CORRECT
4768            026454  103404                          BCS     40$             ;BRANCH IF PACKET IN RAM IS CORRECT
4772            026456                                  ERRHRD  ERRNO,PKTRAM,RAMERR     ;REPORT THE RAM ERROR(S)
                026456  104456                                                          TRAP    C$ERHRD
```

```
      026460  000626                                                    .WORD   406
      026462  004643                                                    .WORD   PKTRAM
      026464  016204                                                    .WORD   RAMERR
 4773
 4774 026466                      40$:    ENDSEG                      ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      026466                                                                10000$:
      026466  104405                                                          TRAP    C$ESEG
 4775
 4776 026470  012364  000006              MOV     (R3)+,PKBCNT(R4)    ;SET THE TEST WORD
 4777 026474  020327  003030              CMP     R3,#TBLEND          ;HAS ALL DATA BEEN TESTED ?
 4778 026500  103002                      BHIS    55$                 ;BRANCH IF ALL DATA DONE
 4779 026502  000137  026252              JMP     5$                  ;BRANCH TILL BACK TO ZERO
 4780
 4781 026506                      55$:    ENDSUB                      ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
      026506                                                                L10051:
      026506  104403                                                          TRAP    C$ESUB
 4782
 4783 026510  005737  002170              TST     FATFLG              ;ANY FATAL ERRORS ?
 4784 026514  001402                      BEQ     60$                 ;BRANCH IF NOT
 4785 026516  004737  017776              JSR     PC,CKDROP           ;TRY TO DROP THE UNIT
 4786 026522                      60$:
```

```
                                ;+
                                ;
                                ;TEST 4, SUBTEST 2
                                ;
                                ;CHECK THAT UNUSED BITS BEING SET CAUSES
                                ;WRITE CHARACTERISTICS COMMAND TO BE REJECTED
                                ;
                                ;-

                                        BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
                                                                                        T4.2:
                                                                                                TRAP    C$BSUB

                                        SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
                                                                                                MOV     #PRI00,R0
                                                                                                TRAP    C$SPRI
                                        MOV     #T42DATA,R3             ;START OF TEST DATA FOR SUBTEST
                                5$:     MOV     #T4PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
                                        JSR     PC,T4REST               ;RESTORE PACKET TO STARTING VALUES

                                        BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
                                                                                                TRAP    C$BSEG

                                        JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
                                        BCS     10$                     ;BR IF SOFT INIT = OK
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
                                                                                                TRAP    C$ERDF
                                                                                                .WORD   407
                                                                                                .WORD   SFIERR
                                                                                                .WORD   SFIMSG
                                10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
                                        MOV     R4,R0                   ;START OF THE COMMAND PACKET
                                        ADD     (R3),R0                 ;OFFSET TO THE DATA WORD TO TEST
                                        BIS     2(R3),(R0)              ;SET THE DATA BITS TO BE TESTED
                                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
                                        JSR     PC,WAITF                ;WAIT FOR SSR TO SET
                                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        ERRDF   ERRNO,T4SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
                                                                                                TRAP    C$ERDF
                                                                                                .WORD   408
                                                                                                .WORD   T4SSR
                                                                                                .WORD   PKTSSR
                                15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
                                                                                                TRAP    C$CLP1
                                        ESCAPE  SEG                     ;BY-PASS CHECKS IF FATAL ERROR
                                                                                                TRAP    C$ESCAPE
                                                                                                .WORD   10000$-.
                                        TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
                                        BEQ     22$                     ;BRANCH IF NOT
                                        ERRHRD  ERRNO,T4INT,PKTSSR
                                                                                                TRAP    C$ERHRD
                                                                                                .WORD   409
                                                                                                .WORD   T4INT
                                                                                                .WORD   PKTSSR
                                22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
```

```
 4834 026660  012702  102206                MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
 4835 026664  032701  000100                 IT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 4836 026670  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
 4837 026672  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 4838 026676  020201                 25$:   CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 4839 026700  001414                        BEQ     30$                     ;OKAY IF MATCH
 4840 026702  010100                        MOV     R1,R0                   ;DATA FROM TSSR
 4841 026704                                XOR     R2,R0                   ;FIND BITS IN ERROR
 4842 026714  020027  002000                CMP     R0,#NBA                 ;IS NBA ONLY BIT IN ERROR ?
 4843 026720  001404                        BEQ     30$                     ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
 4847 026722                                ERRHRD  ERRNO,T42REJ,PKTSSR     ;COMMAND NOT REJECTED
      026722  104456                                                            TRAP    C$ERHRD
      026724  000632                                                            .WORD   410
      026726  027461                                                            .WORD   T42REJ
      026730  011520                                                            .WORD   PKTSSR
 4848 026732                         30$:   CKLOOP                          ;LOOP ON ERROR ?
      026732  104406                                                            TRAP    C$CLP1
 4849 026734  032701  002000                BIT     #NBA,R1                 ;IS NBA BIT SET ?
 4850 026740  001004                        BNE     35$                     ;OKAY IF NBA SET
 4854 026742                                ERRHRD  ERRNO,T42NBA,PKTSSR     ;NBA NOT SET
      026742  104456                                                            TRAP    C$ERHRD
      026744  000633                                                            .WORD   411
      026746  027330                                                            .WORD   T42NBA
      026750  011520                                                            .WORD   PKTSSR
 4855 026752                         35$:
 4856 026752                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      026752                                                                 10000$:
      026752  104405                                                            TRAP    C$ESEG
 4857
 4858 026754  062703  000004                ADD     #4,R3                   ;POINT TO NEXT DATA PAIR
 4859 026760  020327  027330                CMP     R3,#T42DONE             ;COMPARE TO END OF TEST DATA
 4860 026764  103002                        BHIS    57$                     ;BRANCH IF ALL DATA TESTED
 4861 026766  000137  026536                JMP     5$                      ;BRANCH TILL BACK TO ZERO
 4862
 4863 026772                         57$:   ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
      026772                                                                 L10052:
      026772  104403                                                            TRAP    C$ESUB
```

```
 4865
 4866                                   ;*
 4867                                   ;
 4868                                   ;TEST 4, SUBTEST 3
 4869                                   ;
 4870                                   ;SUBTEST TO VERIFY THAT A WRITE CHARACTERISTICS COMMAND IS
 4871                                   ;REJECTED IF AN ILLEGAL DATA BLOCK ADDRESS IS ISSUED.
 4872                                   ;
 4873                                   ;-
 4874
 4875 026774                                   BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
      026774                                                                                    T4.3:
      026774  104402                                                                             TRAP    C$BSUB
 4876
 4877 026776                                   SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      026776  012700  000000                                                                     MOV     #PRI00,R0
      027002  104441                                                                             TRAP    C$SPRI
 4878 027004  012703  027314                   MOV     #T42DATA,R3             ;START OF TEST DATA FOR SUBTEST
 4879 027010  012704  027250           5$:     MOV     #T4PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
 4880 027014  004737  030146                   JSR     PC,T4REST               ;RESTORE PACKET TO STARTING VALUES
 4881
 4882
 4883 027020  004737  016470                   JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 4884 027024  103405                           BCS     10$                     ;BR IF SOFT INIT = OK
 4888 027026  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4889 027030                                   ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      027030  104455                                                                             TRAP    C$ERDF
      027032  000634                                                                             .WORD   412
      027034  003550                                                                             .WORD   SFIERR
      027036  011506                                                                             .WORD   SFIMSG
 4890 027040  005037  002172           10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 4891 027044  052737  000001  027260           BIS     #1,T4DATA               ;MAKE ADDRESS ODD
 4892 027052  010465  177776                   MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 4893 027056  004737  016744                   JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 4894 027062  103405                           BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
 4895 027064  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 4899 027066                                   ERRDF   ERRNO,T4SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      027066  104455                                                                             TRAP    C$ERDF
      027070  000635                                                                             .WORD   413
      027072  027656                                                                             .WORD   T4SSR
      027074  011520                                                                             .WORD   PKTSSR
 4900 027076                           15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      027076  104406                                                                             TRAP    C$CLP1
 4901 027100                                   ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
      027100  104410                                                                             TRAP    C$ESCAPE
      027102  000116                                                                             .WORD   L10053-.
 4902 027104  005737  002172                   TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 4903 027110  001404                           BEQ     22$                     ;BRANCH IF NOT
 4907 027112                                   ERRHRD  ERRNO,T4INT,PKTSSR
      027112  104456                                                                             TRAP    C$ERHRD
      027114  000636                                                                             .WORD   414
      027116  027745                                                                             .WORD   T4INT
      027120  011520                                                                             .WORD   PKTSSR
 4908 027122  016501  000000           22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
 4909 027126  012702  102206                   MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
 4910 027132  032701  000100                   BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 4911 027136  001402                           BEQ     25$                     ;BRANCH IF NOT OFF-LINE
```

```
4912 027140  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
4913 027144  020201                 25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
4914 027146  001414                         BEQ     30$                     ;OKAY IF MATCH
4915 027150  010100                         MOV     R1,R0                   ;DATA FROM TSSR
4916 027152                                 XOR     R2,R0                   ;FIND BITS IN ERROR
4917 027162  020027  002000                CMP     R0,#NBA                 ;IS NBA ONLY BIT IN ERROR ?
4918 027166  001404                         BEQ     30$                     ;DON'T PRINT ERROR IF NBA ONLY BAD BIT
4922 027170                                 ERRHRD  ERRNO,T44REJ,PKTSSR     ;COMMAND NOT REJECTED
     027170  104456                                                                 TRAP    C$ERHRD
     027172  000637                                                                 .WORD   415
     027174  027560                                                                 .WORD   T44REJ
     027176  011520                                                                 .WORD   PKTSSR
4923 027200                         30$:    CKLOOP                          ;LOOP ON ERROR ?
     027200  104406                                                                 TRAP    C$CLP1
4924 027202  032701  002000                BIT     #NBA,R1                 ;IS NBA BIT SET ?
4925 027206  001004                         BNE     35$                     ;OKAY IF NBA SET
4929 027210                                 ERRHRD  ERRNO,T42NBA,PKTSSR     ;NBA NOT SET
     027210  104456                                                                 TRAP    C$ERHRD
     027212  000640                                                                 .WORD   416
     027214  027330                                                                 .WORD   T42NBA
     027216  011520                                                                 .WORD   PKTSSR
4930
4931 027220                         35$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     027220                                                                      L10053:
     027220  104403                                                                 TRAP    C$ESUB
4932
4933 027222  005737  002170                TST     FATFLG                  ;ANY FATAL ERRORS ?
4934 027226  001402                         BEQ     60$                     ;BRANCH IF NOT
4935 027230  004737  017776                JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
4936 027234                         60$:
4937 027234                                 EXIT    TST                     ;ALL DONE THIS TEST
     027234  104432                                                                 TRAP    C$EXIT
     027236  000756                                                                 .WORD   L10050-.
4938
4939                                ;+
4940                                ;LOCAL STORAGE FOR THIS TEST
4941                                ;-
4942
4944 027240                                 .BLKB   10-<.-TUV2A&7>
4946 027250                         T4PACKET:                               ;COMMAND PACKET FOR TEST
4947 027250  100004                         .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
4948 027252  027260                         .WORD   T4DATA                  ;ADDRESS OF CHARACTERISTICS BLOCK
4949 027254  000000                         .WORD   0
4950 027256  000010                         .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
4951
4952 027260                         T4DATA:                                 ;CHARACTERISTICS DATA BLOCK
4953 027260  027274                         .WORD   T4BFR                   ;ADDRESS OF MESSAGE BUFFER
4954 027262  000000                         .WORD   0
4955 027264  000016                         .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
4956 027266  000000                         .WORD   0
4957 027270                         T4SP:
4958
4959 027270  000000  000000                .WORD   0,0                     ;SPACE
4960 027274                         T4BFR:  .BLKW   8.                      ;MESSAGE BUFFER
4961
4962                                ;+
4963                                ;
```

```
 4964                                   ;TEST DATA FOR SUBTEST TWO
 4965                                   ;
 4966                                   ;DATA HAS FORMAT:
 4967                                   ;
 4968                                   ;       1ST WORD        OFFSET TO TEST WORD IN PACKET
 4969                                   ;       2ND WORD        BITS TO SET FOR TEST
 4970                                   ;
 4971                                   ;-
 4972
 4973 027314                            T42DATA:
 4974 027314  000000  037140                       .WORD   0,BIT5!BIT6!BIT9!BIT10!BIT11!BIT12!BIT13
 4975 027320  000002  000001                       .WORD   2,BIT0
 4976 027324  000004  100100                       .WORD   4,BIT6!BIT15
 4977         027330                    T42DONE=.
 4978
 4979
 4980                                   ;*
 4981                                   ;LOCAL TEXT MESSAGES FOR TEST
 4982                                   ;-
 4983
 4984 027330     116     102     101   T42NBA: .ASCIZ  'NBA Not Set On Rejected WRITE CHARACTERISTICS'
 4985 027406     127     122     111   T4NBA:  .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
 4986 027461     127     122     111   T42REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Non-Zero Unused Fields'
 4987 027560     127     122     111   T44REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Block Address'
 4988 027656     103     157     156   T4SSR:  .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
 4989 027745     125     156     145   T4INT:  .ASCIZ  'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
 4990 030034     111     156     143   T4TSBA: .ASCIZ  'Incorrect TSBA Address After WRITE CHARACTERISTICS'
 4991 030117     127     162     151   TST4ID: .ASCIZ  'Write Characteristics'
 4992                                           .EVEN
 4993
```

```
 4995
 4996                                   ;+
 4997                                   ;
 4998                                   ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
 4999                                   ;
 5000                                   ;-
 5001
 5002 030146                            T4REST:
 5003 030146                                    SAVREG                  ;SAVE THE REGISTERS
 5004 030152  012701  027250                      MOV     #T4PACKET,R1    ;START OF THE PACKET
 5005 030156  012721  100004                      MOV     #100004,(R1)+   ;WRITE CHARACTERISTICS WITH ACK
 5006 030162  012721  027260                      MOV     #T4DATA,(R1)+   ;ADDRESS OF CHAR DATA BLOCK
 5007 030166  005021                              CLR     (R1)+           ;EXTENDED ADDRESS
 5008 030170  012721  000010                      MOV     #8.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
 5009 030174  012721  027274                      MOV     #T4BFR,(R1)+    ;ADDRESS OF MESSAGE BUFFER
 5010 030200  005021                              CLR     (R1)+
 5011 030202  012721  000020                      MOV     #16.,(R1)+      ;LENGTH OF MESSAGE BUFFER
 5012 030206  005021                              CLR     (R1)+
 5013 030210  005011                              CLR     (R1)
 5014 030212  000207                              RTS     PC              ;RETURN
 5015 030214                                    ENDTST
      030214                                                                    L10050:
      030214  104401                                                                   TRAP    C$ETST
 5016
```

```
                                .SBTTL  TEST  5: VOLUME CHECK


                        ;       THIS TEST VERIFIES THAT THE VOLUME CHECK (VCK) BIT, A FLAG HELD
                        ;       WITHIN THE CONTROLLER AND APPEARING IN XSTO, IS SET BY INITIALIZE AND
                        ;       CLEARED BY EXECUTING A WRITE CHARACTERISTICS COMMAND WITH THE
                        ;       CVC BIT SET.  IT IS ALSO VERIFIED THAT A WRITE CHARACTERISTICS
                        ;       COMMAND WITH THE CVC BIT CLEAR DOES NOT AFFECT THE STATE OF THE
                        ;       VOLUME CHECK BIT.  THE ACTUAL FUNCTION OF VOLUME CHECK, THAT OF
                        ;       PREVENTING OR ALLOWING A TAPE MOTION COMMAND DEPENDING UPON
                        ;       WHETHER VOLUME CHECK IS SET OR CLEAR, IS NOT CHECKED BY THIS
                        ;       TEST;  THIS FUNCTIONALITY IS CHECKED IN THE INDIVIDUAL TESTS OF
                        ;       TAPE MOTION COMMANDS.
                        ;
                        ;       THE TEST PROCEEDS AS FOLLOWS:
                        ;
                        ;            1.  THE CONTROLLER IS INITIALIZED BY WRITING INTO THE TSSR.
                        ;
                        ;            2.  A WRITE CHARACTERISTICS COMMAND IS ISSUED (WITH CVC=0)
                        ;                AND XSTO IN THE RETURNED MESSAGE BUFFER IS EXAMINED;
                        ;                THE VCK BIT SHOULD BE CLEAR (0).
                        ;
                        ;            3.  THE PREVIOUS STEP IS REPEATED TO VERIFY THAT VCK DOES
                        ;                NOT CHANGE (REMAINS AT 0).
                        ;
                        ;            4.  A WRITE CHARACTERISTICS COMMAND IS ISSUED WITH CVC=1
                        ;                AND THE VCK BIT IN XSTO IN THE MESSAGE BUFFER IS
                        ;                EXAMINED;  THE VCK BIT SHOULD BE CLEAR (0).
                        ;
                        ;            5.  A WRITE CHARACTERISTICS COMMAND IS ISSUED WITH CVC=0
                        ;                AND THE VCK BIT IN XSTO IN THE MESSAGE BUFFER IS
                        ;                EXAMINED;  THE VCK BIT SHOULD REMAIN CLEAR (0).
                        ;
                        ;
030216                          BGNTST
030216                                                          T5::
030216  005037  002170        CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
030222  012737  005672  002146  MOV     #EPRT1,EPRTSW           ;SET UP ERROR MESSAGE SWITCH
030230  005037  003100        CLR     KTFLG           ;HOLD OFF KT11
030234  012700  031407        MOV     #TST5ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
030240  004737  017232        JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
030244  012737  000002  002164  MOV     #2.,LOOPCNT             ;PERFORM 2 ITERATIONS
030252                  T5LOOP:
                        
030252  012704  030720        MOV     #T5PACKET,R4            ;PACKET FOR WRITE CHARACTERISTICS
030256  012702  030742        MOV     #T5BFR,R2               ;ADDRESS OF THE MESSAGE BUFFER
030262  012762  052525  000006  MOV     #052525,XSTO(R2)        ;SET XSTATO TO KNOWN VALUE
030270  004737  016470  5$:     JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
030274  103405                BCS     10$                     ;BR IF SOFT INIT = OK
030276  010001                MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
030300                        ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
030300  104455                                                        TRAP    C$ERDF
030302  000765                                                        .WORD   501
030304  003550                                                        .WORD   SFIERR
030306  011506                                                        .WORD   SFIMSG
030310  042714  040000  10$:     BIC     #BIT14,(R4)             ;CLEAR THE CVC BIT
030314  010465  177776          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS FOR WRITE CHAR
```

```
5077 030320  004737  017060                JSR     PC,CHKTSSR          ;WAIT FOR SSR TO SET
5078 030324  103405                        BCS     15$                 ;BR IF CARRY SET (GOOD RETURN)
5079 030326  010001                        MOV     R0,R1               ;SAVE CONTENTS OF TSSR
5083 030330                                ERRDF   ERRNO,T5SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
     030330  104455                                                        TRAP    C$ERDF
     030332  000766                                                        .WORD   502
     030334  031221                                                        .WORD   T5SSR
     030336  011520                                                        .WORD   PKTSSR
5084 030340                        15$:    CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
     030340  104406                                                        TRAP    C$CLP1
5085 030342                                ESCAPE  TST                 ;EXIT IF FATAL ERROR
     030342  104410                                                        TRAP    C$ESCAPE
     030344  001060                                                        .WORD   L10054-.
5086 030346  016203  000006                MOV     XSTO(R2),R3         ;STORE STATUS FOR A WHILE
5087 030352  020327  052525                CMP     R3,#052525          ;CHECK FOR XSTATO OVER WRITTEN (GOOD!)
5088 030356  001006                        BNE     20$                 ;BR, IF XSTATO HAS BEEN UPDATED
5089 030360  016501  000000                MOV     TSSR(R5),R1         ;PICK UP TSSR FOR ERROR PRINTOUT
5093 030364                                ERRHRD  ERRNO,T5NMSG,PKTSSR ;"NO MESSAGE PACKET RETURNED"
     030364  104456                                                        TRAP    C$ERHRD
     030366  000767                                                        .WORD   503
     030370  031310                                                        .WORD   T5NMSG
     030372  011520                                                        .WORD   PKTSSR
5094 030374  032762  000020  000006  20$:    BIT     #XSOVCK,XSTO(R2)    ;IS VOLUME CHECK CLEAR IN XSTO ?
5095 030402  001006                        BNE     23$                 ;OKAY IF VOLUME CHECK IS CLEARED
5099 030404  016501  000000                MOV     TSSR(R5),R1         ;CONTENTS OF TSSR FOR ERROR REPORT
5100 030410                                ERRHRD  ERRNO,T5VCK2,PKTMES ;VOLUME CHECK NOT SET
     030410  104456                                                        TRAP    C$ERHRD
     030412  000770                                                        .WORD   504
     030414  031055                                                        .WORD   T5VCK2
     030416  011574                                                        .WORD   PKTMES
5101 030420                        23$:    CKLOOP                      ;LOOP ON ERROR ?
     030420  104406                                                        TRAP    C$CLP1
5102 030422  010465  177776                MOV     R4,TSDB(R5)         ;SET THE PACKET ADDRESS FOR WRITE CHAR
5103 030426  004737  017060                JSR     PC,CHKTSSR          ;WAIT FOR SSR TO SET
5104 030432  103405                        BCS     25$                 ;BR IF CARRY SET (GOOD RETURN)
5105 030434  010001                        MOV     R0,R1               ;SAVE CONTENTS OF TSSR
5109 030436                                ERRDF   ERRNO,T5SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
     030436  104455                                                        TRAP    C$ERDF
     030440  000771                                                        .WORD   505
     030442  031221                                                        .WORD   T5SSR
     030444  011520                                                        .WORD   PKTSSR
5110 030446                        25$:    CKLOOP                      ;LOOP ON ERROR, IF FLAG SET
     030446  104406                                                        TRAP    C$CLP1
5111 030450                                ESCAPE  TST                 ;EXIT IF FATAL ERROR
     030450  104410                                                        TRAP    C$ESCAPE
     030452  000752                                                        .WORD   L10054-.
5112 030454  026203  000006                CMP     XSTO(R2),R3         ;THE XSTO SHOULD NOT HAVE CHANGED
5113 030460  001406                        BEQ     27$                 ;OKAY IF VOLUME CHECK IS SET
5117 030462  016501  000000                MOV     TSSR(R5),R1         ;CONTENTS OF TSSR FOR ERROR REPORT
5118 030466                                ERRHRD  ERRNO,T5NVCK,PKTMES ;VOLUME CHECK NOT SET
     030466  104456                                                        TRAP    C$ERHRD
     030470  000772                                                        .WORD   506
     030472  031131                                                        .WORD   T5NVCK
     030474  011574                                                        .WORD   PKTMES
5119 030476                        27$:    CKLOOP                      ;LOOP ON ERROR ?
     030476  104406                                                        TRAP    C$CLP1
5120 030500  032762  000020  000006  30$:    BIT     #XSOVCK,XSTO(R2)    ;IS VOLUME CHECK SET IN XSTO ?
```

```
5121 030506  001006                          BNE     33$                ;OKAY IF VOLUME CHECK IS SET
5125 030510  016501  000000                  MOV     T5SR(R5),R1        ;CONTENTS OF TSSR FOR ERROR REPORT
5126 030514                                  ERRHRD  ERRNO,T5VCK2,PKTMES ;VOLUME CHECK NOT SET
     030514  104456                                                          TRAP    C$ERHRD
     030516  000773                                                          .WORD   507
     030520  031055                                                          .WORD   T5VCK2
     030522  011574                                                          .WORD   PKTMES
5127 030524                          33$:    CKLOOP                     ;LOOP ON ERROR ?
     030524  104406                                                          TRAP    C$CLP1
5128 030526  052714  040000                  BIS     #BIT14,(R4)        ;SET THE CVC BIT
5129 030532  010465  177776                  MOV     R4,T5DB(R5)        ;SET THE PACKET ADDRESS FOR WRITE CHAR
5130 030536  004737  017060                  JSR     PC,CHKTSSR         ;WAIT FOR SSR TO SET
5131 030542  103405                          BCS     35$                ;BR IF CARRY SET (GOOD RETURN)
5132 030544  010001                          MOV     R0,R1              ;SAVE CONTENTS OF TSSR
5136 030546                                  ERRDF   ERRNO,T5SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
     030546  104455                                                          TRAP    C$ERDF
     030550  000774                                                          .WORD   508
     030552  031221                                                          .WORD   T5SSR
     030554  011520                                                          .WORD   PKTSSR
5137 030556                          35$:    CKLOOP                     ;LOOP ON ERROR, IF FLAG SET
     030556  104406                                                          TRAP    C$CLP1
5138 030560                                  ESCAPE  TST                ;EXIT IF FATAL ERROR
     030560  104410                                                          TRAP    C$ESCAPE
     030562  000642                                                          .WORD   L10054-.
5139 030564  032762  000020  000006          BIT     #XS0VCK,XST0(R2)   ;IS VOLUME CHECK CLEAR IN XST0 ?
5140 030572  001406                          BEQ     40$                ;OKAY IF VOLUME CHECK IS CLEARED
5144 030574  016501  000000                  MOV     TSSR(R5),R1        ;CONTENTS OF TSSR FOR ERROR REPORT
5145 030600                                  ERRHRD  ERRNO,T5VCK,PKTMES ;VOLUME CHECK NOT CLEARED
     030600  104456                                                          TRAP    C$ERHRD
     030602  000775                                                          .WORD   509
     030604  030762                                                          .WORD   T5VCK
     030606  011574                                                          .WORD   PKTMES
5146 030610                          40$:    CKLOOP                     ;LOOP ON ERROR ?
     030610  104406                                                          TRAP    C$CLP1
5147 030612  042714  040000                  BIC     #BIT14,(R4)        ;CLEAR THE CVC BIT
5148 030616  010465  177776                  MOV     R4,T5DB(R5)        ;SET THE PACKET ADDRESS FOR WRITE CHAR
5149 030622  004737  017060                  JSR     PC,CHKTSSR         ;WAIT FOR SSR TO SET
5150 030626  103405                          BCS     45$                ;BR IF CARRY SET (GOOD RETURN)
5151 030630  010001                          MOV     R0,R1              ;SAVE CONTENTS OF TSSR
5155 030632                                  ERRDF   ERRNO,T5SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
     030632  104455                                                          TRAP    C$ERDF
     030634  000776                                                          .WORD   510
     030636  031221                                                          .WORD   T5SSR
     030640  011520                                                          .WORD   PKTSSR
5156 030642                          45$:    CKLOOP                     ;LOOP ON ERROR, IF FLAG SET
     030642  104406                                                          TRAP    C$CLP1
5157 030644                                  ESCAPE  TST                ;EXIT IF FATAL ERROR
     030644  104410                                                          TRAP    C$ESCAPE
     030646  000556                                                          .WORD   L10054-.
5158 030650  032762  000020  000006          BIT     #XS0VCK,XST0(R2)   ;IS VOLUME CHECK CLEAR IN XST0 ?
5159 030656  001406                          BEQ     50$                ;OKAY IF VOLUME CHECK IS CLEARED
5163 030660  016501  000000                  MOV     TSSR(R5),R1        ;CONTENTS OF TSSR FOR ERROR REPORT
5164 030664                                  ERRHRD  ERRNO,T5NVCK,PKTMES ;VOLUME CHECK NOT CLEARED
     030664  104456                                                          TRAP    C$ERHRD
     030666  000777                                                          .WORD   511
     030670  031131                                                          .WORD   T5NVCK
     030672  011574                                                          .WORD   PKTMES
```

```
5165 030674                          50$:    CKLOOP                          ;LOOP ON ERROR ?
     030674  104406                                                                  TRAP    C$CLP1
5166 030676  004737  017200          60$:    JSR     PC,TSTLOOP              ;SHOULD WE DO ITERATIONS ?
5167 030702  103002                          BCC     62$                     ;BRANCH IF NOT
5168 030704  000137  030252                  JMP     T5LOOP                  ;LOOP UNTIL COUNT EXPIRED
5169 030710                          62$:    EXIT    TST                     ;ALL DONE THIS TEST
     030710  104432                                                                  TRAP    C$EXIT
     030712  000512                                                                  .WORD   L10054-.
5170
5171                                 ;+
5172                                 ;LOCAL STORAGE FOR THIS TEST
5173                                 ;-
5174
5176 030714                                  .BLKB   10-<.-TUV2A&7>
5178 030720                          T5PACKET:                               ;COMMAND PACKET FOR TEST
5179 030720  100004                          .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND
5180 030722  030730                          .WORD   T5DATA                  ;ADDRESS OF CHARACTERISTICS BLOCK
5181 030724  000000                          .WORD   0
5182 030726  000010                          .WORD   10                      ;STARTING VALUE OF COUNTER
5183
5184 030730                          T5DATA:                                 ;CHARACTERISTICS DATA BLOCK
5185 030730  030742                          .WORD   T5BFR                   ;ADDRESS OF MESSAGE BUFFER
5186 030732  000000                          .WORD   0
5187 030734  000020                          .WORD   16.                     ;LENGTH OF MESSAGE BUFFER
5188 030736  000000  000000                  .WORD   0,0
5189
5190 030742                          T5BFR:  .BLKW   8.                      ;MESSAGE BUFFER
5191
5192
5193                                 ;+
5194                                 ;LOCAL TEXT MESSAGES FOR TEST
5195                                 ;-
5196
5197 030762     126     103     113 T5VCK:  .ASCIZ  'VCK Bit NOT Cleared After WRITE CHARACTERISTICS With CVC=1'
5198 031055     126     103     113 T5VCK2: .ASCIZ  'VCK Bit NOT Set After INITIALIZE With CVC=0'
5199 031131     126     103     113 T5NVCK: .ASCIZ  'VCK Bit Modified After WRITE CHARACTERISTICS With CVC=0'
5200 031221     103     157     156 T5SSR:  .ASCIZ  'Contents of TSSR Incorrect After Write Characteristics'
5201 031310     116     157     040 T5NMSG: .ASCIZ  'No Message Packet Returned To Host After WRITE CHARACTERISTICS'
5202 031407     126     157     154 TST5ID: .ASCIZ  'Volume Check'
5203                                         .EVEN
5204 031424                                  ENDTST
     031424                                                         L10054:
     031424  104401                                                         TRAP    C$ETST
```

```
                                .SBTTL  TEST  6: COMPLETION INTERRUPT

                        ;       THIS TEST VERIFIES THAT AN INTERRUPT IS GENERATED AT THE
                        ;       COMPLETION OF THE WRITE CHARACTERISTICS COMMAND IF THE INTERRUPT
                        ;       ENABLE (IE) BIT IN THE COMMAND HEADER WORD IS SET.  THIS TEST
                        ;       CHECKS THE FUNCTIONING OF THE INTERRUPT LOGIC AND BASIC
                        ;       PROCESSING OF THE IE BIT.
                        ;
                        ;       THE SEQUENCES OF TEST 7 ARE REPEATED, EXCEPT THAT THE INTERRUPT
                        ;       SERVICE ROUTINE IS SET UP TO EXPECT INTERRUPTS AND EACH WRITE
                        ;       CHARACTERISTICS COMMAND IS ISSUED WITH THE IE BIT SET (1).  IT
                        ;       IS VERIFIED, WHERE APPROPRIATE, THAT THE IE STATUS BIT IN XSTO
                        ;       OF ANY MESSAGE PACKET IS SET AND THAT A COMPLETION INTERRUPT IS
                        ;       GENERATED.  FINALLY, A SEQUENCE OF TWO COMMANDS ARE ISSUED, THE
                        ;       FIRST WITH IE=1 AND THE SECOND WITH IE=0.  IT IS VERIFIED THAT
                        ;       NO INTERRUPT IS GENERATED AFTER THE SECOND COMMAND AND THAT THE
                        ;       IE BIT IN XSTO IS 0.
                        ;
031426                          BGNTST
031426                                                          T6::
031426  005037  002170          CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
031432  012737  005672  002146  MOV     #EPRT1,EPRTSW           ;SET UP ERROR MESSAGE SWITCH
031440  005037  003100          CLR     KTFLG           ;HOLD OFF KT11
031444  012700  033401          MOV     #TST6ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
031450  004737  017232          JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
031454  012737  000002  002164  MOV     #2.,LOOPCNT             ;PERFORM 2 ITERATIONS
031462                  T6LOOP:
031462                          BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
031462                                                                  T6.1:
031462  104402                                                                  TRAP    C$BSUB
                        ;
031464  004737  033426          JSR     PC,T6REST               ;SET PACKET TO INITIAL VALUES
031470                          SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
031470  012700  000000                                                  MOV     #PRI00,R0
031474  104441                                                          TRAP    C$SPRI
031476  012703  002732          MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
031502  012704  032330          MOV     #T6PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
031506  012764  000010  000006  MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
031514                  5$:
031514                          BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
031514  104404                                                          TRAP    C$BSEG
                        ;
031516  004737  016470          JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
031522  103405          BCS     10$                             ;BR IF SOFT INIT = OK
031524  010001          MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
031526                          ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
031526  104455                                                          TRAP    C$ERDF
031530  001131                                                          .WORD   601
031532  003550                                                          .WORD   SFIERR
031534  011506                                                          .WORD   SFIMSG
031536  005037  002170          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
031542  005037  002172          CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
031546  010465  177776          MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
031552  004737  017060          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
031556  103407          BCS     15$                             ;BR IF CARRY SET (GOOD RETURN)
031560  010001          MOV     R0,R1                           ;SAVE CONTENTS OF TSSR
```

```
 5263 031562                            ERRDF   ERRNO,T6SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      031562  104455                                                                 TRAP    C$ERDF
      031564  001132                                                                 .WORD   602
      031566  033047                                                                 .WORD   T6SSR
      031570  011520                                                                 .WORD   PKTSSR
 5264 031572  004737  017724            JSR     PC,FATCHK       ;INC AND CHECK FOR MORE THAN 25 ERRORS
 5265 031576                    15$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
      031576  104406                                                                 TRAP    C$CLP1
 5266 031600                            ESCAPE  SEG             ;BY-PASS SUBTEST IF FATAL ERROR
      031600  104410                                                                 TRAP    C$ESCAPE
      031602  000056                                                                 .WORD   10000$-.
 5267 031604  005737  002172            TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
 5268 031610  001004                    BNE     22$             ;BRANCH IF YES
 5272 031612                            ERRHRD  ERRNO,T6NINT,PKTSSR
      031612  104456                                                                 TRAP    C$ERHRD
      031614  001133                                                                 .WORD   603
      031616  033136                                                                 .WORD   T6NINT
      031620  011520                                                                 .WORD   PKTSSR
 5273 031622  016501  000000    22$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
 5274 031626  012702  000200            MOV     #SSR,R2         ;EXPECTED CONTENTS OF TSSR
 5275 031632  032701  000100            BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
 5276 031636  001402                    BEQ     25$             ;BRANCH IF NOT OFF-LINE
 5277 031640  052702  000100            BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
 5278 031644  020201    25$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
 5279 031646  001404                    BEQ     30$             ;OKAY IF MATCH
 5283 031650                            ERRHRD  ERRNO,T6NBA,PKTSSR      ;NBA NOT ZERO
      031650  104456                                                                 TRAP    C$ERHRD
      031652  001134                                                                 .WORD   604
      031654  032406                                                                 .WORD   T6NBA
      031656  011520                                                                 .WORD   PKTSSR
 5284 031660                    30$:
 5285 031660                            ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      031660                                                                    10000$:
      031660  104405                                                                 TRAP    C$ESEG
 5286
 5287 031662  012364  000006            MOV     (R3)+,PKBCNT(R4)        ;SET THE TEST WORD
 5288 031666  020327  003030            CMP     R3,#TBLEND      ;HAS ALL DATA BEEN TESTED ?
 5289 031672  103002                    BHIS    55$             ;BRANCH IF ALL DATA DONE
 5290 031674  000137  031514            JMP     5$              ;BRANCH TILL BACK TO ZERO
 5291
 5292 031700                    55$:    ENDSUB                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
      031700                                                                    L10056:
      031700  104403                                                                 TRAP    C$ESUB
 5293
 5294 031702  005737  002170            TST     FATFLG          ;ANY FATAL ERRORS ?
 5295 031706  001402                    BEQ     70$             ;BRANCH IF NOT
 5296 031710  004737  017776            JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
 5297 031714                    70$:
```

```
 5299                                   ;*
 5300                                   ;
 5301                                   ;TEST 6, SUBTEST 2
 5302                                   ;
 5303                                   ;CHECK THAT UNUSED BITS BEING SET CAUSES
 5304                                   ;WRITE CHARACTERISTICS COMMAND TO BE REJECTED
 5305                                   ;
 5306                                   ;-
 5307
 5308 031714                                   BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
      031714                                                                                    T6.2:
      031714  104402                                                                                   TRAP    C$BSUB
 5309
 5310 031716                                   SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      031716  012700  000000                                                                           MOV     #PRI00,R0
      031722  104441                                                                                   TRAP    C$SPRI
 5311 031724  012703  032372                   MOV     #T62DATA,R3             ;START OF TEST DATA FOR SUBTEST
 5312 031730  012704  032330           5$:     MOV     #T6PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
 5313 031734  004737  033426                   JSR     PC,T6REST               ;RESTORE PACKET TO STARTING VALUES
 5314
 5315 031740                                   BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      031740  104404                                                                             TRAP    C$BSEG
 5316
 5317 031742  004737  016470                   JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 5318 031746  103405                          BCS     10$                     ;BR IF SOFT INIT = OK
 5322 031750  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5323 031752                                   ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      031752  104455                                                                             TRAP    C$ERDF
      031754  001135                                                                             .WORD   605
      031756  003550                                                                             .WORD   SFIERR
      031760  011506                                                                             .WORD   SFIMSG
 5324 031762  005037  002172           10$:    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 5325 031766  010400                          MOV     R4,R0                   ;START OF THE COMMAND PACKET
 5326 031770  061300                          ADD     (R3),R0                 ;OFFSET TO THE DATA WORD TO TEST
 5327 031772  056310  000002                   BIS     2(R3),(R0)              ;SET THE DATA BITS TO BE TESTED
 5328 031776  010465  177776                   MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 5329 032002  004737  016744                   JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 5330 032006  103405                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
 5331 032010  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5335 032012                                   ERRDF   ERRNO,T6SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      032012  104455                                                                                TRAP    C$ERDF
      032014  001136                                                                                .WORD   606
      032016  033047                                                                                .WORD   T6SSR
      032020  011520                                                                                .WORD   PKTSSR
 5336 032022                           15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      032022  104406                                                                                TRAP    C$CLP1
 5337 032024                                   ESCAPE  SEG                     ;BY-PASS CHECKS IF FATAL ERROR
      032024  104410                                                                                TRAP    C$ESCAPE
      032026  000056                                                                                .WORD   10000$-.
 5338 032030  005737  002172                   TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 5339 032034  001004                          BNE     22$                     ;BRANCH IF YES
 5343 032036                                   ERRHRD  ERRNO,T6NINT,PKTSSR
      032036  104456                                                                               TRAP    C$ERHRD
      032040  001137                                                                               .WORD   607
      032042  033136                                                                               .WORD   T6NINT
      032044  011520                                                                               .WORD   PKTSSR
 5344 032046  016501  000000           22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
```

```
 5345 032052  012702  102206                MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
 5346 032056  032701  000100                BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
 5347 032062  001402                        BEQ     25$                     ;BRANCH IF NOT OFF-LINE
 5348 032064  052702  000100                BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
 5349 032070  020201                25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
 5350 032072  001404                        BEQ     30$                     ;OKAY IF MATCH
 5354 032074                                ERRHRD  ERRNO,T62REJ,PKTSSR     ;COMMAND NOT REJECTED
      032074  104456                                                            TRAP    C$ERHRD
      032076  001140                                                            .WORD   608
      032100  032461                                                            .WORD   T62REJ
      032102  011520                                                            .WORD   PKTSSR
 5355 032104                        30$:
 5356 032104                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      032104                                                                    10000$:
      032104  104405                                                            TRAP    C$ESEG
 5357
 5358 032106  062703  000004                ADD     #4,R3                   ;POINT TO NEXT DATA PAIR
 5359 032112  020327  032406                CMP     R3,#T62DONE             ;COMPARE TO END OF TEST DATA
 5360 032116  103002                        BHIS    59$                     ;BRANCH IF ALL DATA TESTED
 5361 032120  000137  031730                JMP     5$                      ;BRANCH TILL BACK TO ZERO
 5362
 5363 032124                        59$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
      032124                                                                    L10057:
      032124  104403                                                            TRAP    C$ESUB
```

```
                                        ;+
                                        ;
                                        ;TEST 6, SUBTEST 3
                                        ;
                                        ;SUBTEST TO VERIFY THAT A WRITE CHARACTERISTICS COMMAND IS
                                        ;REJECTED IF AN ILLEGAL DATA BLOCK ADDRESS IS ISSUED.
                                        ;
                                        ;-

032126                                          BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
032126                                                                                  T6.3:
032126  104402                                                                          TRAP    C$BSUB

032130                                          SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
032130  012700  000000                                                                  MOV     #PRI00,R0
032134  104441                                                                          TRAP    C$SPRI
032136  012703  032372                5$:     MOV     #T62DATA,R3             ;START OF TEST DATA FOR SUBTEST
032142  012704  032330                        MOV     #T6PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
032146  004737  033426                        JSR     PC,T6REST               ;RESTORE PACKET TO STARTING VALUES


032152  004737  016470                        JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
032156  103405                              BCS     10$                     ;BR IF SOFT INIT = OK
032160  010001                              MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
032162                                      ERRDF   ERRN0,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
032162  104455                                                                          TRAP    C$ERDF
032164  001141                                                                          .WORD   609
032166  003550                                                                          .WORD   SFIERR
032170  011506                                                                          .WORD   SFIMSG
032172  005037  002172                10$:     CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
032176  052737  000001  032340                BIS     #1,T6DATA               ;MAKE ADDRESS ODD
032204  010465  177776                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
032210  004737  016744                        JSR     PC,WAITF                ;WAIT FOR SSR TO SET
032214  103405                              BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
032216  010001                              MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
032220                                      ERRDF   ERRN0,T6SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
032220  104455                                                                          TRAP    C$ERDF
032222  001142                                                                          .WORD   610
032224  033047                                                                          .WORD   T6SSR
032226  011520                                                                          .WORD   PKTSSR
032230                                  15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
032230  104406                                                                          TRAP    C$CLP1
032232                                          ESCAPE  SUB                     ;BY-PASS SUBTEST IF FATAL ERROR
032232  104410                                                                          TRAP    C$ESCAPE
032234  000056                                                                          .WORD   L10060-.
032236  005737  002172                        TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
032242  001004                              BNE     22$                     ;BRANCH IF YES
032244                                      ERRHRD  ERRN0,T6NINT,PKTSSR
032244  104456                                                                          TRAP    C$ERHRD
032246  001143                                                                          .WORD   611
032250  033136                                                                          .WORD   T6NINT
032252  011520                                                                          .WORD   PKTSSR
032254  016501  000000                22$:     MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
032260  012702  102206                        MOV     #SC!SSR!TSREJ!NBA,R2    ;EXPECTED CONTENTS OF TSSR
032264  032701  000100                        BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
032270  001402                              BEQ     25$                     ;BRANCH IF NOT OFF-LINE
032272  052702  000100                        BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
```

TEST 6: COMPLETION INTERRUPT

```
5412 032276  020201          25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
5413 032300  001404                  BEQ     30$                     ;OKAY IF MATCH
5417 032302                          ERRHRD  ERRNO,T64REJ,PKTSSR     ;COMMAND NOT REJECTED
     032302  104456                                                          TRAP    C$ERHRD
     032304  001144                                                          .WORD   612
     032306  032653                                                          .WORD   T64REJ
     032310  011520                                                          .WORD   PKTSSR
5418 032312                          30$:
5419
5420 032312                          ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     032312                                                                L10060:
     032312  104403                                                          TRAP    C$ESUB
5421
```

```
5423
5424 032314                                    EXIT    TST                     ;ALL DONE THIS TEST
     032314  104432                                                                    TRAP    C$EXIT
     032316  001162                                                                    .WORD   L10055-.
5425
5426                                    ;+
5427                                    ;LOCAL STORAGE FOR THIS TEST
5428                                    ;-
5429
5431 032320                                     .BLKB   10-<.-TUV2A&7>
5433 032330                             T6PACKET:                               ;COMMAND PACKET FOR TEST
5434 032330  100204                             .WORD   100204                  ;WRITE CHAR COMMAND, WITH IE, ACK
5435 032332  032340                             .WORD   T6DATA                  ;ADDRESS OF CHARACTERISTICS BLOCK
5436 032334  000000                             .WORD   0
5437 032336  000010                             .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
5438
5439 032340                             T6DATA:                                 ;CHARACTERISTICS DATA BLOCK
5440 032340  032352                             .WORD   T6BFR                   ;ADDRESS OF MESSAGE BUFFER
5441 032342  000000                             .WORD   0
5442 032344  000016                             .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
5443 032346  000000  000000                     .WORD   0,0
5444
5445 032352                             T6BFR:  .BLKW   8.                      ;MESSAGE BUFFER
5446
5447                                    ;+
5448                                    ;
5449                                    ;TEST DATA FOR SUBTEST TWO
5450                                    ;
5451                                    ;DATA HAS FORMAT:
5452                                    ;
5453                                    ;       1ST WORD        OFFSET TO TEST WORD IN PACKET
5454                                    ;       2ND WORD        BITS TO SET FOR TEST
5455                                    ;
5456                                    ;-
5457
5458 032372                             T62DATA:
5459 032372  000000  036140                     .WORD   0,BIT5!BIT6!BIT6!BIT10!BIT11!BIT12!BIT13
5460 032376  000002  000001                     .WORD   2,BIT0
5461 032402  000004  100100                     .WORD   4,BIT6!BIT15
5462         032406                     T62DONE=.
5463
5464
5465                                    ;+
5466                                    ;LOCAL TEXT MESSAGES FOR TEST
5467                                    ;-
5468
5469 032406     127     122     111     T6NBA:  .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
5470 032461     127     122     111     T62REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Non-Zero Unused Fields'
5471 032560     127     122     111     T63REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Data Count'
5472 032653     127     122     111     T64REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Block Address'
5473 032751     127     122     111     T65REJ: .ASCIZ  'WRITE CHARACTERISTICS Not Rejected With Invalid Buffer Length'
5474 033047     103     157     156     T6SSR:  .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS
5475 033136     105     170     160     T6NINT: .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
5476 033227     125     156     145     T6INT:  .ASCIZ  'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
5477 033316     111     156     143     T6TSBA: .ASCIZ  'Incorrect TSBA Address After WRITE CHARACTERISTICS'
5478 033401     103     157     155     TST6ID: .ASCIZ  'Completion Interrupt'
5479                                             .EVEN
```

```
 5481
 5482                                   ;+
 5483                                   ;
 5484                                   ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
 5485                                   ;
 5486                                   ;-
 5487
 5488 033426                           T6REST:
 5489 033426                                   SAVREG                          ;SAVE THE REGISTERS
 5490 033432  012701  032330                     MOV     #T6PACKET,R1          ;START OF THE PACKET
 5491 033436  012721  100204                     MOV     #100204,(R1)+         ;WRITE CHARACTERISTICS WITH ACK, IE
 5492 033442  012721  032340                     MOV     #T6DATA,(R1)+         ;ADDRESS OF CHAR DATA BLOCK
 5493 033446  005021                             CLR     (R1)+                 ;EXTENDED ADDRESS
 5494 033450  012721  000010                     MOV     #8.,(R1)+             ;SIZE OF DATA BLOCK IN BYTES
 5495 033454  012721  032352                     MOV     #T6BFR,(R1)+          ;ADDRESS OF MESSAGE BUFFER
 5496 033460  005021                             CLR     (R1)+
 5497 033462  012721  000016                     MOV     #14.,(R1)+            ;LENGTH OF MESSAGE BUFFER
 5498 033466  005021                             CLR     (R1)+
 5499 033470  005011                             CLR     (R1)
 5500 033472  005037  032352                     CLR     T6BFR                 ;CLEAR 1ST LOC IN MESSAGE BUFFER
 5501 033476  000207                             RTS     PC                    ;RETURN
 5502 033500                                   ENDTST
      033500
      033500  104401                                                          L10055:
                                                                                     TRAP    C$ETST
```

```
                                        .SBTTL  TEST  7: BASIC PACKET PROTOCOL

                                ;       THIS TEST VERIFIES BASIC OPERATION OF THE MESSAGE BUFFER RELEASE
                                ;       COMMAND, THE FUNCTION OF THE ACK BIT IN THE COMMAND HEADER WORD,
                                ;       AND THE REGISTER MODIFICATION REFUSED (RMR) LOGIC.
                                ;
                                ;
                                ;+
                                ;TEST 7, SUBTEST 1
                                ;
                                ;CHECKS THAT THE MESSAGE BUFFER RELEASE COMMAND WORKS
                                ;PROPERLY AND THAT NO INTERRUPT IS GENERATED EVEN
                                ;IF THE "IE" BIT IS SET IN THE COMMAND PACKET
                                ;
                                ;-
033502                                  BGNTST
033502                                                                  T7::
033502  005037  002170                 CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
033506  012737  005672  002146         MOV     #EPRT1,EPRTSW           ;SET UP ERROR MESSAGE SWITCH
033514  005037  003100                 CLR     KTFLG           ;HOLD OFF KT11
033520  012700  036537                 MOV     #TST7ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
033524  004737  017232                 JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
033530  012737  000002  002164         MOV     #2.,LOOPCNT             ;PERFORM 2 ITERATIONS
033536                          T7LOOP:
                                ;
033536                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
033536                                                                  T7.1:
033536  104402                                                                  TRAP    C$BSUB
                                ;
033540  004737  036566                 JSR     PC,T7RST                ;SET PACKET TO INITIAL VALUES
033544                                  SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
033544  012700  000000                                                          MOV     #PRI00,R0
033550  104441                                                                  TRAP    C$SPRI
033552  012704  035640                 MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
033556  012764  000010  000006         MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
033564                          5$:
033564                                  BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
033564  104404                                                                  TRAP    C$BSEG
                                ;
033566  004737  016470                 JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
033572  103405                         BCS     10$                     ;BR IF SOFT INIT = OK
033574  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
033576                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
033576  104455                                                                  TRAP    C$ERDF
033600  001275                                                                  .WORD   701
033602  003550                                                                  .WORD   SFIERR
033604  011506                                                                  .WORD   SFIMSG
033606  005037  002170          10$:   CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
033612  005037  002172                 CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
033616  010465  177776                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
033622  004737  017060                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
033626  103407                         BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
033630  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
033632                                  ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
033632  104455                                                                  TRAP    C$ERDF
033634  001276                                                                  .WORD   702
033636  036270                                                                  .WORD   T7SSR
```

```
     033640  011520                                                          .WORD   PKTSSR
5558 033642  004737  017724                JSR     PC,FATCHK          ;INC AND CHECK FOR MORE THAN 25 ERRORS
5559 033646                        15$:    CKLOOP                     ;LOOP ON ERROR, IF FLAG SET
     033646  104406                                                          TRAP    C$CLP1
5560 033650                                ESCAPE  SEG                ;BY-PASS SUBTEST IF FATAL ERROR
     033650  104410                                                          TRAP    C$ESCAPE
     033652  000056                                                          .WORD   10000$-.
5561 033654  005737  002172                TST     INTRECV            ;DID AN INTERRUPT OCCUR ?
5562 033660  001004                        BNE     22$                ;BRANCH IF YES
5566 033662                                ERRHRD  ERRNO,T7NINT,PKTSSR
     033662  104456                                                          TRAP    C$ERHRD
     033664  001277                                                          .WORD   703
     033666  036357                                                          .WORD   T7NINT
     033670  011520                                                          .WORD   PKTSSR
5567 033672  016501  000000        22$:    MOV     TSSR(R5),R1        ;GET THE CONTENTS OF TSSR
5568 033676  012702  000200                MOV     #SSR,R2            ;EXPECTED CONTENTS OF TSSR
5569 033702  032701  000100                BIT     #OFL,R1            ;IS OFF-LINE BIT SET ?
5570 033706  001402                        BEQ     25$                ;BRANCH IF NOT OFF-LINE
5571 033710  052702  000100                BIS     #OFL,R2            ;SET OFF-LINE IN EXPECTED DATA
5572 033714  020201        25$:    CMP     R2,R1              ;DOES EXPECTED MATCH RECEIVED ?
5573 033716  001404                        BEQ     30$                ;OKAY IF MATCH
5577 033720                                ERRHRD  ERRNO,T7SSR,PKTSSR ;NBA NOT ZERO
     033720  104456                                                          TRAP    C$ERHRD
     033722  001300                                                          .WORD   704
     033724  036270                                                          .WORD   T7SSR
     033726  011520                                                          .WORD   PKTSSR
5578 033730                        30$:
5579 033730                                ENDSEG                     ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     033730                                                                10000$:
     033730  104405                                                          TRAP    C$ESEG
5580 033732                                BGNSEG                     ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
     033732  104404                                                          TRAP    C$BSEG
5581
5582 033734  005037  002172                CLR     INTRECV            ;CLEAR INTERRUPT RECEIVED FLAG
5583 033740  012737  025252  035662        MOV     #025252,T7BFR      ;WIPE OUT MESSAGE BUFFER AREA
5584 033746  012714  100212                MOV     #100212,(R4)       ;SET COMMAND PACKET TO MESS BUF REL
5585 033752  010465  177776                MOV     R4,TSDB(R5)        ;SET THE PACKET ADDRESS
5586 033756  004737  017060                JSR     PC,CHKTSSR         ;WAIT FOR SSR TO SET
5587 033762  103407                        BCS     45$                ;BR IF CARRY SET (GOOD RETURN)
5588 033764  010001                        MOV     R0,R1              ;SAVE CONTENTS OF TSSR
5592 033766                                ERRDF   ERRNO,T7SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
     033766  104455                                                          TRAP    C$ERDF
     033770  001301                                                          .WORD   705
     033772  036270                                                          .WORD   T7SSR
     033774  011520                                                          .WORD   PKTSSR
5593 033776  004737  017724                JSR     PC,FATCHK          ;INC AND CHECK FOR MORE THAN 25 ERRORS
5594 034002                        45$:    CKLOOP                     ;LOOP ON ERROR, IF FLAG SET
     034002  104406                                                          TRAP    C$CLP1
5595 034004  005737  002172                TST     INTRECV            ;DID AN INTERRUPT OCCUR ?
5596 034010  001404                        BEQ     52$                ;BRANCH IF NO
5600 034012                                ERRHRD  ERRNO,T7INT,PKTSSR
     034012  104456                                                          TRAP    C$ERHRD
     034014  001302                                                          .WORD   706
     034016  036450                                                          .WORD   T7INT
     034020  011520                                                          .WORD   PKTSSR
5601 034022  016501  000000        52$:    MOV     TSSR(R5),R1        ;GET THE CONTENTS OF TSSR
5602 034026  012702  000200                MOV     #SSR,R2            ;EXPECTED CONTENTS OF TSSR
```

```
 5603 034032  032701  000100            BIT     #OFL,R1          ;IS OFF-LINE BIT SET ?
 5604 034036  001402                    BEQ     55$              ;BRANCH IF NOT OFF-LINE
 5605 034040  052702  000100            BIS     #OFL,R2          ;SET OFF-LINE IN EXPECTED DATA
 5606 034044  020201              55$:  CMP     R2,R1            ;DOES EXPECTED MATCH RECEIVED ?
 5607 034046  001404                    BEQ     60$              ;OKAY IF MATCH
 5611 034050                            ERRHRD  ERRNO,T7SSR,PKTSSR       ;"TSSR INCORRECT AFTER WRT CHARA."
      034050  104456                                                      TRAP    C$ERHRD
      034052  001303                                                      .WORD   707
      034054  036270                                                      .WORD   T7SSR
      034056  011520                                                      .WORD   PKTSSR
 5612 034060                      60$:
 5613 034060  013701  035662            MOV     T7BFR,R1         ;PICK UP THE 1ST WORD OF MESSAGE BUFFER
 5614 034064  012702  025252            MOV     #025252,R2       ;SET UP EXPECTED DATA
 5615 034070  020102                    CMP     R1,R2            ;WAS ANY MESSAGE REC'D
 5616 034072  001404                    BEQ     70$              ;BR, IF OK (EQUAL)
 5620 034074                            ERRHRD  ERRNO,T7MBF,EXPREC       ;MESSAGE BUFFER WAS MODIFIED
      034074  104456                                                      TRAP    C$ERHRD
      034076  001304                                                      .WORD   708
      034100  035744                                                      .WORD   T7MBF
      034102  016170                                                      .WORD   EXPREC
 5621 034104                      70$:
 5622 034104  005737  002170            TST     FATFLG           ;ANY FATAL ERRORS
 5623 034110  001403                    BEQ     80$              ;BR, IF NO FATAL ERRORS
 5624 034112  004737  017776            JSR     PC,CKDROP        ;TRY TO DROP THE UNIT
 5625 034116                            ENDSEG                   ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      034116                                                            10001$:
      034116  104405                                                      TRAP    C$ESEG
 5626 034120                      80$:
 5627 034120                            ENDSUB                   ;////////////// END SUBTEST //////////////
      034120                                                            L10062:
      034120  104403                                                      TRAP    C$ESUB
 5628
```

```
 5630                                   ;+
 5631                                   ;TEST 7, SUBTEST 2
 5632                                   ;
 5633                                   ;CHECKS THAT THE MESSAGE BUFFER RELEASE COMMAND WORKS
 5634                                   ;PROPERLY AND THAT THERE IS AN INTERRUPT IF THE "IE"
 5635                                   ;BIT IS SET IN THE COMMAND PACKET AND THE "ERI" BIT
 5636                                   ;IS SET IN THE CHARACTERISTICS DATA PACKET
 5637                                   ;
 5638                                   ;-
 5639 034122                           BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
      034122                                                                           T7.2:
      034122  104402                                                                   TRAP    C$BSUB
 5640
 5641 034124  004737  036566           JSR     PC,T7RST                ;SET PACKET TO INITIAL VALUES
 5642 034130                           SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      034130  012700  000000                                                           MOV     #PRI00,R0
      034134  104441                                                                   TRAP    C$SPRI
 5643 034136  012704  035640           MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
 5644 034142  012764  000010  000006   MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
 5645 034150                           5$:
 5646 034150                           BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      034150  104404                                                                   TRAP    C$BSEG
 5647
 5648 034152  004737  016470           JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 5649 034156  103405                   BCS     10$                     ;BR IF SOFT INIT = OK
 5653 034160  010001                   MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5654 034162                           ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      034162  104455                                                                   TRAP    C$ERDF
      034164  001305                                                                   .WORD   709
      034166  003550                                                                   .WORD   SFIERR
      034170  011506                                                                   .WORD   SFIMSG
 5655 034172  005037  002170           10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 5656 034176  005037  002172                   CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 5657 034202  012737  000020  035656           MOV     #000020,T7DATA+6        ;SET ERI IN CHARACTERISTICS DATA
 5658 034210  010465  177776                   MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 5659 034214  004737  017060                   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 5660 034220  103407                           BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
 5661 034222  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5665 034224                                   ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      034224  104455                                                                           TRAP    C$ERDF
      034226  001306                                                                           .WORD   710
      034230  036270                                                                           .WORD   T7SSR
      034232  011520                                                                           .WORD   PKTSSR
 5666 034234  004737  017724           JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 5667 034240                   15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      034240  104406                                                                   TRAP    C$CLP1
 5668 034242                           ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
      034242  104410                                                                   TRAP    C$ESCAPE
      034244  000056                                                                   .WORD   10000$-.
 5669 034246  005737  002172           TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 5670 034252  001004                   BNE     22$                     ;BRANCH IF YES
 5674 034254                           ERRHRD  ERRNO,T7NINT,PKTSSR
      034254  104456                                                                   TRAP    C$ERHRD
      034256  001307                                                                   .WORD   711
      034260  036357                                                                   .WORD   T7NINT
      034262  011520                                                                   .WORD   PKTSSR
 5675 034264  016501  000000   22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
```

```
 5676 034270  012702  000200            MOV     #SSR,R2          ;EXPECTED CONTENTS OF TSSR
 5677 034274  032701  000100            BIT     #OFL,R1          ;IS OFF-LINE BIT SET ?
 5678 034300  001402                    BEQ     25$              ;BRANCH IF NOT OFF-LINE
 5679 034302  052702  000100            BIS     #OFL,R2          ;SET OFF-LINE IN EXPECTED DATA
 5680 034306  020201              25$:  CMP     R2,R1            ;DOES EXPECTED MATCH RECEIVED ?
 5681 034310  001404                    BEQ     30$              ;OKAY IF MATCH
 5685 034312                            ERRHRD  ERRNO,T7SSR,PKTSSR  ;NBA NOT ZERO
      034312  104456                                                    TRAP    C$ERHRD
      034314  001310                                                    .WORD   712
      034316  036270                                                    .WORD   T7SSR
      034320  011520                                                    .WORD   PKTSSR
 5686 034322                      30$:
 5687 034322                            ENDSEG                   ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      034322                                                          10000$:
      034322  104405                                                    TRAP    C$ESEG
 5688 034324                            BGNSEG                   ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
      034324  104404                                                    TRAP    C$BSEG
 5689
 5690 034326  005037  002172            CLR     INTRECV          ;CLEAR INTERRUPT RECEIVED FLAG
 5691 034332  012737  025252  035662    MOV     #025252,T7BFR    ;WIPE OUT MESSAGE BUFFER AREA
 5692 034340  012714  100212            MOV     #100212,(R4)     ;SET COMMAND PACKET TO MESS BUF REL
 5693 034344  010465  177776            MOV     R4,TSDB(R5)      ;SET THE PACKET ADDRESS
 5694 034350  004737  017060            JSR     PC,CHKTSSR       ;WAIT FOR SSR TO SET
 5695 034354  103407                    BCS     45$              ;BR IF CARRY SET (GOOD RETURN)
 5696 034356  010001                    MOV     R0,R1            ;SAVE CONTENTS OF TSSR
 5700 034360                            ERRDF   ERRNO,T7SSR,PKTSSR  ;DEVICE FATAL SSR FAILED TO SET
      034360  104455                                                    TRAP    C$ERDF
      034362  001311                                                    .WORD   713
      034364  036270                                                    .WORD   T7SSR
      034366  011520                                                    .WORD   PKTSSR
 5701 034370  004737  017724            JSR     PC,FATCHK        ;INC AND CHECK FOR MORE THAN 25 ERRORS
 5702 034374              45$:  CKLOOP                           ;LOOP ON ERROR, IF FLAG SET
      034374  104406                                                    TRAP    C$CLP1
 5703 034376  005737  002172            TST     INTRECV          ;DID AN INTERRUPT OCCUR ?
 5704 034402  001004                    BNE     52$              ;BRANCH IF YES
 5708 034404                            ERRHRD  ERRNO,T7INT,PKTSSR
      034404  104456                                                    TRAP    C$ERHRD
      034406  001312                                                    .WORD   714
      034410  036450                                                    .WORD   T7INT
      034412  011520                                                    .WORD   PKTSSR
 5709 034414  016501  000000      52$:  MOV     TSSR(R5),R1      ;GET THE CONTENTS OF TSSR
 5710 034420  012702  000200            MOV     #SSR,R2          ;EXPECTED CONTENTS OF TSSR
 5711 034424  032701  000100            BIT     #OFL,R1          ;IS OFF-LINE BIT SET ?
 5712 034430  001402                    BEQ     55$              ;BRANCH IF NOT OFF-LINE
 5713 034432  052702  000100            BIS     #OFL,R2          ;SET OFF-LINE IN EXPECTED DATA
 5714 034436  020201              55$:  CMP     R2,R1            ;DOES EXPECTED MATCH RECEIVED ?
 5715 034440  001404                    BEQ     60$              ;OKAY IF MATCH
 5719 034442                            ERRHRD  ERRNO,T7SSR,PKTSSR  ;NBA NOT SET
      034442  104456                                                    TRAP    C$ERHRD
      034444  001313                                                    .WORD   715
      034446  036270                                                    .WORD   T7SSR
      034450  011520                                                    .WORD   PKTSSR
 5720 034452                      60$:
 5721 034452  013701  035662            MOV     T7BFR,R1         ;PICK UP THE 1ST WORD OF MESSAGE BUFFER
 5722 034456  012702  025252            MOV     #025252,R2       ;SET UP EXPECTED DATA
 5723 034462  020102                    CMP     R1,R2            ;WAS ANY MESSAGE REC'D
 5724 034464  001404                    BEQ     70$              ;BR, IF OK (EQUAL)
```

```
 5728 034466                                ERRHRD  ERRNO,T7MBF,EXPREC      ;MESSAGE BUFFER WAS MODIFIED
      034466  104456                                                          TRAP    C$ERHRD
      034470  001314                                                          .WORD   716
      034472  035744                                                          .WORD   T7MBF
      034474  016170                                                          .WORD   EXPREC
 5729
 5730 034476                          70$:
 5731 034476  005737  002170                 TST     FATFLG                  ;ANY FATAL ERRORS
 5732 034502  001402                         BEQ     80$                     ;BR, IF NO FATAL ERRORS
 5733 034504  004737  017776                 JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 5734 034510                          80$:
 5735 034510                                ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      034510                                                                  10001$:
      034510  104405                                                          TRAP    C$ESEG
 5736 034512                                ENDSUB                          ;////////////// END SUBTEST //////////////
      034512                                                                  L10063:
      034512  104403                                                          TRAP    C$ESUB
```

```
                                        ;+
                                        ;TEST 7, SUBTEST 3
                                        ;
                                        ;CHECKS THAT THE CPU GIVES UP OWNERSHIP OF THE MESSAGE BUFFER
                                        ;AFTER THE MESSAGE BUFFER RELEASE, AND THAT FOLLOWING COMMANDS
                                        ;WORK CORRECTLY
                                        ;
                                        ;-
5740 034514                             BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
     034514                                                                                 T7.3:
     034514  104402                                                                             TRAP    C$BSUB
5747
5748 034516  004737  036566             JSR     PC,T7RST                ;SET PACKET TO INITIAL VALUES
5749 034522                             SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
     034522  012700  000000                                                                     MOV     #PRI00,R0
     034526  104441                                                                             TRAP    C$SPRI
5750 034530  012704  035640             MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
5751 034534  012764  000010  000006     MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
5752 034542                     5$:
5753 034542                             BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
     034542  104404                                                                             TRAP    C$BSEG
5754
5755 034544  004737  016470             JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
5756 034550  103405                     BCS     10$                     ;BR IF SOFT INIT = OK
5760 034552  010001                     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5761 034554                             ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
     034554  104455                                                                             TRAP    C$ERDF
     034556  001315                                                                             .WORD   717
     034560  003550                                                                             .WORD   SFIERR
     034562  011506                                                                             .WORD   SFIMSG
5762 034564  005037  002170     10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
5763 034570  005037  002172             CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
5764 034574  010465  177776             MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
5765 034600  004737  017060             JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
5766 034604  103407                     BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
5767 034606  010001                     MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
5771 034610                             ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     034610  104455                                                                             TRAP    C$ERDF
     034612  001316                                                                             .WORD   718
     034614  036270                                                                             .WORD   T7SSR
     034616  011520                                                                             .WORD   PKTSSR
5772 034620  004737  017724             JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
5773 034624                     15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     034624  104406                                                                             TRAP    C$CLP1
5774 034626                             ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     034626  104410                                                                             TRAP    C$ESCAPE
     034630  000056                                                                             .WORD   10000$-.
5775 034632  005737  002172             TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
5776 034636  001004                     BNE     22$                     ;BRANCH IF YES
5780 034640                             ERRHRD  ERRNO,T7NINT,PKTSSR
     034640  104456                                                                             TRAP    C$ERHRD
     034642  001317                                                                             .WORD   719
     034644  036357                                                                             .WORD   T7NINT
     034646  011520                                                                             .WORD   PKTSSR
5781 034650  016501  000000     22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
5782 034654  012702  000200             MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
5783 034660  032701  000100             BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
```

```
 5784 034664  001402                    BEQ     25$                ;BRANCH IF NOT OFF-LINE
 5785 034666  052702  000100            BIS     #OFL,R2            ;SET OFF-LINE IN EXPECTED DATA
 5786 034672  020201              25$:  CMP     R2,R1              ;DOES EXPECTED MATCH RECEIVED ?
 5787 034674  001404                    BEQ     30$                ;OKAY IF MATCH
 5791 034676                            ERRHRD  ERRNO,T7SSR,PKTSSR ;NBA NOT ZERO
      034676  104456                                                       TRAP    C$ERHRD
      034700  001320                                                       .WORD   720
      034702  036270                                                       .WORD   T7SSR
      034704  011520                                                       .WORD   PKTSSR
 5792 034706                      30$:
 5793 034706                            ENDSEG                     ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      034706                                                              10000$:
      034706  104405                                                       TRAP    C$ESEG
 5794 034710                            BGNSEG                     ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
      034710  104404                                                       TRAP    C$BSEG
 5795 034712  004737  036566            JSR     PC,T7RST           ;RESET PACKETS AND COMMANDS
 5796 034716  005037  002172            CLR     INTRECV            ;CLEAR INTERRUPT RECEIVED FLAG
 5797 034722  012737  025252  035662    MOV     #025252,T7BFR      ;WIPE OUT MESSAGE BUFFER AREA
 5798 034730  012714  100212            MOV     #100212,(R4)       ;SET COMMAND PACKET TO MESS BUF REL
 5799 034734  010465  177776            MOV     R4,TSDB(R5)        ;SET THE PACKET ADDRESS
 5800 034740  004737  017060            JSR     PC,CHKTSSR         ;WAIT FOR SSR TO SET
 5801 034744  103407                    BCS     45$                ;BR IF CARRY SET (GOOD RETURN)
 5802 034746  010001                    MOV     R0,R1              ;SAVE CONTENTS OF TSSR
 5806 034750                            ERRDF   ERRNO,T7SSRM,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      034750  104455                                                       TRAP    C$ERDF
      034752  001321                                                       .WORD   721
      034754  036200                                                       .WORD   T7SSRM
      034756  011520                                                       .WORD   PKTSSR
 5807 034760  004737  017724            JSR     PC,FATCHK          ;INC AND CHECK FOR MORE THAN 25 ERRORS
 5808 034764                      45$:  CKLOOP                     ;LOOP ON ERROR, IF FLAG SET
      034764  104406                                                       TRAP    C$CLP1
 5809 034766  005737  002172            TST     INTRECV            ;DID AN INTERRUPT OCCUR ?
 5810 034772  001404                    BEQ     52$                ;BRANCH IF NO
 5814 034774                            ERRHRD  ERRNO,T7INT,PKTSSR
      034774  104456                                                       TRAP    C$ERHRD
      034776  001322                                                       .WORD   722
      035000  036450                                                       .WORD   T7INT
      035002  011520                                                       .WORD   PKTSSR
 5815 035004  016501  000000      52$:  MOV     TSSR(R5),R1        ;GET THE CONTENTS OF TSSR
 5816 035010  012702  000200            MOV     #SSR,R2            ;EXPECTED CONTENTS OF TSSR
 5817 035014  032701  000100            BIT     #OFL,R1            ;IS OFF-LINE BIT SET ?
 5818 035020  001402                    BEQ     55$                ;BRANCH IF NOT OFF-LINE
 5819 035022  052702  000100            BIS     #OFL,R2            ;SET OFF-LINE IN EXPECTED DATA
 5820 035026  020201              55$:  CMP     R2,R1              ;DOES EXPECTED MATCH RECEIVED ?
 5821 035030  001404                    BEQ     60$                ;OKAY IF MATCH
 5825 035032                            ERRHRD  ERRNO,T7SSR,PKTSSR ;NBA NOT SET
      035032  104456                                                       TRAP    C$ERHRD
      035034  001323                                                       .WORD   723
      035036  036270                                                       .WORD   T7SSR
      035040  011520                                                       .WORD   PKTSSR
 5826 035042                      60$:
 5827 035042  013701  035662            MOV     T7BFR,R1           ;PICK UP THE 1ST WORD OF MESSAGE BUFFER
 5828 035046  012702  025252            MOV     #025252,R2         ;SET UP EXPECTED DATA
 5829 035052  020102                    CMP     R1,R2              ;WAS ANY MESSAGE REC'D
 5830 035054  001404                    BEQ     70$                ;BR, IF OK (EQUAL)
 5834 035056                            ERRHRD  ERRNO,T7MBF,EXPREC ;MESSAGE BUFFER WAS MODIFIED
      035056  104456                                                       TRAP    C$ERHRD
```

```
      035060  001324                                                  .WORD   724
      035062  005744                                                  .WORD   T7MBF
      035064  016170                                                  .WORD   EXPREC
 5835
 5836 035066                          70$:    CKLOOP                  ;LOOP ON ERROR IF FLAG SET
      035066  104406                                                  TRAP    C$CLP1
 5837
 5838 035070  005037  002172                  CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
 5839 035074  004737  036566                  JSR     PC,T7RST        ;RESET THE PACKETS AND COMMANDS
 5840 035100  042714  100000                  BIC     #100000,(R4)    ;CLEAR THE ACK BIT
 5841 035104  010465  177776                  MOV     R4,TSDB(R5)     ;SET THE PACKET ADDRESS
 5842 035110  004737  017060                  JSR     PC,CHKTSSR      ;WAIT FOR SSR TO SET
 5843 035114  103407                          BCS     75$             ;BR IF CARRY SET (GOOD RETURN)
 5844 035116  010001                          MOV     R0,R1           ;SAVE CONTENTS OF TSSR
 5848 035120                                  ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      035120  104455                                                  TRAP    C$ERDF
      035122  001325                                                  .WORD   725
      035124  036270                                                  .WORD   T7SSR
      035126  011520                                                  .WORD   PKTSSR
 5849 035130  004737  017724                  JSR     PC,FATCHK       ;INC AND CHECK FOR MORE THAN 25 ERRORS
 5850 035134                          75$:    CKLOOP                  ;LOOP ON ERROR, IF FLAG SET
      035134  104406                                                  TRAP    C$CLP1
 5851 035136                                  ESCAPE  SEG             ;BY-PASS SUBTEST IF FATAL ERROR
      035136  104410                                                  TRAP    C$ESCAPE
      035140  000062                                                  .WORD   10001$-.
 5852 035142  005737  002172                  TST     INTRECV         ;DID AN INTERRUPT OCCUR ?
 5853 035146  001006                          BNE     82$             ;BRANCH IF YES
 5857 035150  016500  000000                  MOV     TSSR(R5),R0     ;GET TSSR FOR ERROR REPORT
 5858 035154                                  ERRHRD  ERRNO,T7NINT,PKTSSR
      035154  104456                                                  TRAP    C$ERHRD
      035156  001326                                                  .WORD   726
      035160  036357                                                  .WORD   T7NINT
      035162  011520                                                  .WORD   PKTSSR
 5859 035164  016501  000000          82$:    MOV     TSSR(R5),R1     ;GET THE CONTENTS OF TSSR
 5860 035170  012702  000200                  MOV     #SSR,R2         ;EXPECTED CONTENTS OF TSSR
 5861 035174  032701  000100                  BIT     #OFL,R1         ;IS OFF-LINE BIT SET ?
 5862 035200  001402                          BEQ     85$             ;BRANCH IF NOT OFF-LINE
 5863 035202  052702  000100                  BIS     #OFL,R2         ;SET OFF-LINE IN EXPECTED DATA
 5864 035206  020201                  85$:    CMP     R2,R1           ;DOES EXPECTED MATCH RECEIVED ?
 5865 035210  001404                          BEQ     90$             ;OKAY IF MATCH
 5869 035212                                  ERRHRD  ERRNO,T7SSR,PKTSSR      ;NBA NOT ZERO
      035212  104456                                                  TRAP    C$ERHRD
      035214  001327                                                  .WORD   727
      035216  036270                                                  .WORD   T7SSR
      035220  011520                                                  .WORD   PKTSSR
 5870 035222                          90$:
 5871 035222                                  ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      035222                                                          10001$:
      035222  104405                                                  TRAP    C$ESEG
 5872 035224  005737  002170                  TST     FATFLG          ;ANY FATAL ERRORS
 5873 035230  001403                          BEQ     95$             ;BR, IF NO FATAL ERRORS
 5874 035232  004737  017776                  JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
 5875
 5876 035236                                  BGNSEG                  ;<<<<<<<<<<<<< BGN SEGMENT <<<<<<<<<<<<<
      035236  104404                                                  TRAP    C$BSEG
 5877 035240  005037  002172          95$:    CLR     INTRECV         ;CLEAR INTERRUPT RECEIVED FLAG
 5878 035244  004737  036566                  JSR     PC,T7RST        ;RESET THE PACKETS AND COMMANDS
```

```
 5879 035250  010465  177776                 MOV     R4,TSDB(R5)          ;SET THE PACKET ADDRESS
 5880 035254  004737  017060                 JSR     PC,CHKTSSR           ;WAIT FOR SSR TO SET
 5881 035260  103407                         BCS     100$                 ;BR IF CARRY SET (GOOD RETURN)
 5882 035262  010001                         MOV     R0,R1                ;SAVE CONTENTS OF TSSR
 5886 035264                                 ERRDF   ERRNO,T7SSR,PKTSSR   ;DEVICE FATAL SSR FAILED TO SET
      035264  104455                                                              TRAP    C$ERDF
      035266  001330                                                             .WORD   728
      035270  036270                                                             .WORD   T7SSR
      035272  011520                                                             .WORD   PKTSSR
 5887 035274  004737  017724                 JSR     PC,FATCHK            ;INC AND CHECK FOR MORE THAN 25 ERRORS
 5888 035300                         100$:   CKLOOP                       ;LOOP ON ERROR, IF FLAG SET
      035300  104406                                                             TRAP    C$CLP1
 5889 035302                                 ESCAPE  SEG                  ;BY-PASS SUBTEST IF FATAL ERROR
      035302  104410                                                             TRAP    C$ESCAPE
      035304  000062                                                             .WORD   10002$-.
 5890 035306  005737  002172                 TST     INTRECV              ;DID AN INTERRUPT OCCUR ?
 5891 035312  001006                         BNE     112$                 ;BRANCH IF YES
 5895 035314  016500  000000                 MOV     TSSR(R5),R0          ;GET TSSR FOR ERROR REPORT
 5896 035320                                 ERRHRD  ERRNO,T7NINT,PKTSSR
      035320  104456                                                             TRAP    C$ERHRD
      035322  001331                                                             .WORD   729
      035324  036357                                                             .WORD   T7NINT
      035326  011520                                                             .WORD   PKTSSR
 5897 035330  016501  000000         112$:   MOV     TSSR(R5),R1          ;GET THE CONTENTS OF TSSR
 5898 035334  012702  000200                 MOV     #SSR,R2              ;EXPECTED CONTENTS OF TSSR
 5899 035340  032701  000100                 BIT     #OFL,R1              ;IS OFF-LINE BIT SET ?
 5900 035344  001402                         BEQ     125$                 ;BRANCH IF NOT OFF-LINE
 5901 035346  052702  000100                 BIS     #OFL,R2              ;SET OFF-LINE IN EXPECTED DATA
 5902 035352  020201                 125$:   CMP     R2,R1                ;DOES EXPECTED MATCH RECEIVED ?
 5903 035354  001405                         BEQ     130$                 ;OKAY IF MATCH
 5907 035356                                 ERRHRD  ERRNO,T7SSR,PKTSSR   ;NBA NOT ZERO
      035356  104456                                                             TRAP    C$ERHRD
      035360  001332                                                             .WORD   730
      035362  036270                                                             .WORD   T7SSR
      035364  011520                                                             .WORD   PKTSSR
 5908 035366                                 ENDSEG                       ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
      035366                                                                  10002$:
      035366  104405                                                             TRAP    C$ESEG
 5909 035370  005737  002170         130$:   TST     FATFLG               ;CHECK FATAL FLAG FOR SET
 5910 035374  001402                         BEQ     140$                 ;BR, IF NOT SET (OK)
 5911 035376  004737  017776                 JSR     PC,CKDROP            ;DROP DEVICE IF INDICATED
 5912 035402                         140$:
 5913 035402                                 ENDSUB                       ;////////////// END SUBTEST //////////////
      035402                                                                  L10064:
      035402  104403                                                             TRAP    C$ESUB
 5914 035404                         30$:
```

```
 5916                                   ;+
 5917                                   ;TEST 7, SUBTEST 4
 5918                                   ;
 5919                                   ;CHECKS THAT THE REGISTER MODIFICATION REFUSED (RMR) BIT IN
 5920                                   ;THE TSSR WILL BE SET IF A WRITE CHARACTERISTICS COMMAND
 5921                                   ;BEING EXECUTED AND ANOTHER "WC" COMMAND IS ATTEMPTED
 5922                                   ;
 5923                                   ;-
 5924
 5925 035404                                   BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
      035404                                                                                    T7.4:
      035404  104402                                                                                   TRAP    C$BSUB
 5926
 5927 035406  004737  036640                   JSR     PC,T7RT2                ;SET SECOND PACKET UP
 5928 035412  004737  036566                   JSR     PC,T7RST                ;SET PACKET TO INITIAL VALUES
 5929 035416                                   SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      035416  012700  000000                                                                            MOV     #PRI00,R0
      035422  104441                                                                                   TRAP    C$SPRI
 5930 035424  012704  035640                   MOV     #T7PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
 5931 035430  012703  035702                   MOV     #T7PKT,R3               ;GET THE ADDRESS OF 2ND CMD PACKET
 5932 035434  012764  000010  000006           MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
 5933 035442  012763  000010  000006           MOV     #8.,PKBCNT(R3)          ;START WITH MINIMUM ALLOWABLE VALUE
 5934 035450                            5$:
 5935 035450                                   BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      035450  104404                                                                                   TRAP    C$BSEG
 5936 035452  004737  016470                   JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 5937 035456  103405                           BCS     10$                     ;BR IF SOFT INIT = OK
 5941 035460  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 5942 035462                                   ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      035462  104455                                                                                   TRAP    C$ERDF
      035464  001333                                                                                   .WORD   731
      035466  003550                                                                                   .WORD   SFIERR
      035470  011506                                                                                   .WORD   SFIMSG
 5943 035472  005037  002170            10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 5944 035476  005037  002172                    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 5945 035502  010465  177776                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 5946 035506  010365  177776                    MOV     R3,TSDB(R5)             ;SECOND COMMAND PACKET
 5947 035512  004737  016744                    JSR     PC,WAITF                ;WAIT FOR SSR TO SET
 5948 035516  016501  000000                    MOV     TSSR(R5),R1             ;GET CONTENTS OF TSSR REGISTER
 5949 035522  032701  000200                    BIT     #SSR,R1                 ;CHECK FOR SSR (TSSR) SET
 5950 035526  001006                            BNE     15$                     ;BR, IF SSR SET (GOOD)
 5954 035530                                    ERRDF   ERRNO,T7SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
      035530  104455                                                                                   TRAP    C$ERDF
      035532  001334                                                                                   .WORD   732
      035534  036270                                                                                   .WORD   T7SSR
      035536  011520                                                                                   .WORD   PKTSSR
 5955 035540  004737  017724                    JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
 5956 035544                            15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      035544  104406                                                                                   TRAP    C$CLP1
 5957 035546                                    ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
      035546  104410                                                                                   TRAP    C$ESCAPE
      035550  000056                                                                                   .WORD   10000$-.
 5958 035552  005737  002172                    TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 5959 035556  001004                            BNE     22$                     ;BRANCH IF YES
 5960
 5961
 5965 035560                                    ERRHRD  ERRNO,T7NINT,PKTSSR
```

```
        035560  104456                                                          TRAP    C$ERHRD
        035562  001335                                                          .WORD   733
        035564  036357                                                          .WORD   T7NINT
        035566  011520                                                          .WORD   PKTSSR
  5966  035570  016501  000000          22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
  5967  035574  012702  110200                  MOV     #SSR!RMR!SC,R2          ;EXPECTED CONTENTS OF TSSR
  5968  035600  032701  000100                  BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
  5969  035604  001402                          BEQ     25$                     ;BRANCH IF NOT OFF-LINE
  5970  035606  052702  000100                  BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
  5971  035612  020201          25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
  5972  035614  001404                          BEQ     30$                     ;OKAY IF MATCH
  5976  035616                                  ERRHRD  ERRNO,T7SSR,PKTSSR      ;NBA NOT ZERO
        035616  104456                                                          TRAP    C$ERHRD
        035620  001336                                                          .WORD   734
        035622  036270                                                          .WORD   T7SSR
        035624  011520                                                          .WORD   PKTSSR
  5977  035626          30$:
  5978  035626                                  ENDSEG                  ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
        035626                                                                  10000$:
        035626  104405                                                          TRAP    C$ESEG
  5979  035630                                  ENDSUB                  ;////////////// END SUBTEST //////////////
        035630                                                                  L10065:
        035630  104403                                                          TRAP    C$ESUB
  5980  035632                                  EXIT    TST             ;ALL DONE WITH THIS TEST
        035632  104432                                                          TRAP    C$EXIT
        035634  001056                                                          .WORD   L10061-.
```

```
                                        ;*
                                        ;LOCAL STORAGE FOR THIS TEST
                                        ;-

                                                .BLKB   10-<.-TUV2A&7>
                                        T7PACKET:                       ;COMMAND PACKET FOR TEST
                                                .WORD   100204          ;WRITE CHAR COMMAND, WITH IE, ACK
                                                .WORD   T7DATA          ;ADDRESS OF CHARACTERISTICS BLOCK
                                                .WORD   0
                                                .WORD   8.              ;STARTING VALUE OF BLOCK SIZE

                                        T7DATA:                         ;CHARACTERISTICS DATA BLOCK
                                                .WORD   T7BFR           ;ADDRESS OF MESSAGE BUFFER
                                                .WORD   0
                                                .WORD   14.             ;LENGTH OF MESSAGE BUFFER
                                                .WORD   0,0

                                        T7BFR:  .BLKW   8.              ;MESSAGE BUFFER

                                        ;*
                                        ;
                                        ;TEST DATA FOR SUBTEST FOUR
                                        ;
                                        T7PKT:                          ;COMMAND PACKET FOR TEST
                                                .WORD   100204          ;WRITE CHAR COMMAND, WITH IE, ACK
                                                .WORD   T7DTA           ;ADDRESS OF CHARACTERISTICS BLOCK
                                                .WORD   0
                                                .WORD   8.              ;STARTING VALUE OF BLOCK SIZE

                                        T7DTA:                          ;CHARACTERISTICS DATA BLOCK
                                                .WORD   T7BUFR          ;ADDRESS OF MESSAGE BUFFER
                                                .WORD   0
                                                .WORD   14.             ;LENGTH OF MESSAGE BUFFER
                                                .WORD   0,0

                                        T7BUFR:.BLKW    8.              ;MESSAGE BUFFER

                                        ;*
                                        ;LOCAL TEXT MESSAGES FOR TEST
                                        ;-


                115     145     163  T7MBF:  .ASCIZ  'Message Buffer Modified after MESSAGE BUFFER RELEASE Command'
                116     102     101  T7NBA:  .ASCIZ  'NBA Not Clear After WRITE CHARACTERISTICS Command'
                116     102     101  T7NNBA: .ASCIZ  'NBA Set After MESSAGE BUFFER RELEASE Command'

                103     157     156  T7SSRM: .ASCIZ  'Contents Of TSSR Incorrect After Message Buffer Release'
                103     157     156  T7SSR:  .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
                105     170     160  T7NINT: .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
                125     156     145  T7INT:  .ASCIZ  'Unexpected Interrupt Received On WRITE CHARACTERISTICS'
                102     141     163  TST7ID: .ASCIZ  'Basic Packet Protocol'
                                                .EVEN
```

```
6038
6039                                    ;*
6040                                    ;
6041                                    ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
6042                                    ;
6043                                    ;-
6044
6045 036566                            T7RST:
6046 036566                                    SAVREG                          ;SAVE THE REGISTERS
6047 036572  012701  035640                     MOV     #T7PACKET,R1            ;START OF THE PACKET
6048 036576  012721  100204                     MOV     #100204,(R1)+           ;WRITE CHARACTERISTICS WITH ACK, IE
6049 036602  012721  035650                     MOV     #T7DATA,(R1)+           ;ADDRESS OF CHAR DATA BLOCK
6050 036606  005021                             CLR     (R1)+                   ;EXTENDED ADDRESS
6051 036610  012721  000010                     MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
6052 036614  012721  035662                     MOV     #T7BFR,(R1)+            ;ADDRESS OF MESSAGE BUFFER
6053 036620  005021                             CLR     (R1)+
6054 036622  012721  000016                     MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
6055 036626  005021                             CLR     (R1)+
6056 036630  005011                             CLR     (R1)
6057 036632  005037  035662                     CLR     T7BFR                   ;CLEAR 1ST LOC IN MESSAGE BUFFER
6058 036636  000207                             RTS     PC                      ;RETURN
6059                                    ;*
6060                                    ;
6061                                    ;ROUTINE TO RESTORE COMMAND PACKET #2 TO START-UP (DEFAULT) VALUES
6062                                    ;
6063                                    ;-
6064
6065 036640                            T7RT2:
6066 036640                                    SAVREG                          ;SAVE THE REGISTERS
6067 036644  012701  035702                     MOV     #T7PKT,R1               ;START OF THE PACKET
6068 036650  012721  100204                     MOV     #100204,(R1)+           ;WRITE CHARACTERISTICS WITH ACK, IE
6069 036654  012721  035712                     MOV     #T7DTA,(R1)+            ;ADDRESS OF CHAR DATA BLOCK
6070 036660  005021                             CLR     (R1)+                   ;EXTENDED ADDRESS
6071 036662  012721  000010                     MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
6072 036666  012721  035724                     MOV     #T7BUFR,(R1)+           ;ADDRESS OF MESSAGE BUFFER
6073 036672  005021                             CLR     (R1)+
6074 036674  012721  000016                     MOV     #14.,(R1)+              ;LENGTH OF MESSAGE BUFFER
6075 036700  005021                             CLR     (R1)+
6076 036702  005011                             CLR     (R1)
6077 036704  005037  035724                     CLR     T7BUFR                  ;CLEAR 1ST LOC IN MESSAGE BUFFER
6078 036710  000207                             RTS     PC                      ;RETURN
6079 036712                            ENDTST
     036712                                                             L10061:
     036712  104401                                                             TRAP    C$ETST
```

```
                                        .SBTTL  TEST  8: NON-TAPE MOTION COMMANDS

                                ;+
                                ;
                                ;THIS TEST VERIFIES PROPER OPERATION OF THE INITIALIZE
                                ;COMMAND. TWO SUBTESTS ARE USED. THE FIRST VERIFIES THAT
                                ;THE COMMAND RUNS TO COMPLETION AND STORES A VALID
                                ;MESSAGE PACKET. THE SECOND VERIFIES THAT NON-ZERO
                                ;VALUES IN THE COMMAND MODE FIELD CAUSES COMMAND REJECT.
                                ;
                                ;-

036714                                  BGNTST
036714                                                                  T8::
036714  005037  002170                    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
036720  012737  005672  002146            MOV     @EPRT1,EPRTSW           ;SET UP ERROR MESSAGE SWITCH
036726  005037  003100                    CLR     KTFLG           ;HOLD OFF KT11
036732  012700  040272                    MOV     @TST8ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
036736  004737  017232                    JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
036742  012737  000002  002164            MOV     @2.,LOOPCNT             ;PERFORM 2 ITERATIONS
036750                          T8LOOP:
036750                                  BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
036750                                                                          T8.1:
036750  104402                                                                  TRAP    C$BSUB

036752                                  SETPRI  @PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
036752  012700  000000                                                          MOV     @PRI00,R0
036756  104441                                                                  TRAP    C$SPRI
036760  004737  016470                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
036764  103405                          BCS     3$                      ;BR IF SOFT INIT = OK
036766  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
036770                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
036770  104455                                                                  TRAP    C$ERDF
036772  001441                                                                  .WORD   801
036774  003550                                                                  .WORD   SFIERR
036776  011506                                                                  .WORD   SFIMSG
037000                          3$:
037000  012704  037520                  MOV     @T8PK2,R4               ;WRITE CHARACTERISTICS PACKET
037004  004737  010152                  JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
037010  103404                          BCS     4$                      ;BR, IF COMMAND ISSUED OK
037012                                  ERR'1RD ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
037012  104456                                                                  TRAP    C$ERHRD
037014  001442                                                                  .WORD   802
037016  004754                                                                  .WORD   WRTMSG
037020  011506                                                                  .WORD   SFIMSG
037022                          4$:
037022  004737  040324                  JSR     PC,T8REST               ;SET UP PACKET FOR COMMAND
037026  012704  037450                  MOV     @T8PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
037032                          5$:
037032                                  BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
037032  104404                                                                  TRAP    C$BSEG

037034  005037  002170          10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
037040  005037  002172                  CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
037044  010465  177776                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
037050  004737  017060                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
037054  103407                          BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
```

```
6135 037056  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6139 037060                                 ERRDF   ERRNO,T8SSR,PKTSSR      ;DEVICE FATAL SSR FAILED TO SET
     037060  104455                                                                 TRAP    C$ERDF
     037062  001443                                                                 .WORD   803
     037064  040014                                                                 .WORD   T8SSR
     037066  011520                                                                 .WORD   PKTSSR
6140 037070  004737  017724                 JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
6141 037074                         15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     037074  104406                                                                 TRAP    C$CLP1
6142 037076                                 ESCAPE  SEG                     ;BY-PASS SUBTEST IF FATAL ERROR
     037076  104410                                                                 TRAP    C$ESCAPE
     037100  000074                                                                 .WORD   10000$-.
6143 037102  005737  002172                 TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
6144 037106  001004                         BNE     22$                     ;BRANCH IF YES
6148 037110                                 ERRHRD  ERRNO,T8NINT,PKTSSR
     037110  104456                                                                 TRAP    C$ERHRD
     037112  001444                                                                 .WORD   804
     037114  040144                                                                 .WORD   T8NINT
     037116  011520                                                                 .WORD   PKTSSR
6149 037120  016501  000000         22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
6150 037124  012702  000200                 MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
6151 037130  032701  000100                 BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
6152 037134  001402                         BEQ     25$                     ;BRANCH IF NOT OFF-LINE
6153 037136  052702  000100                 BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
6154 037142  020201         25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
6155 037144  001404                         BEQ     30$                     ;OKAY IF MATCH
6159 037146                                 ERRHRD  ERRNO,T8NBA,PKTSSR      ;NBA NOT ZERO
     037146  104456                                                                 TRAP    C$ERHRD
     037150  001445                                                                 .WORD   805
     037152  037562                                                                 .WORD   T8NBA
     037154  011520                                                                 .WORD   PKTSSR
6160 037156                         30$:
6161 037156  004737  010354         35$:    JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
6162 037162  103405                         BCS     59$                     ;RAM OK GO ON
6166 037164                                 ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
     037164  104456                                                                 TRAP    C$ERHRD
     037166  001446                                                                 .WORD   806
     037170  004643                                                                 .WORD   PKTRAM
     037172  016204                                                                 .WORD   RAMERR
6167 037174                                 ENDSEG                          ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     037174                                                                 10000$:
     037174  104405                                                                 TRAP    C$ESEG
6168
6169
6170 037176                         59$:    ENDSUB                          ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     037176                                                                 L10067:
     037176  104403                                                                 TRAP    C$ESUB
6171
6172 037200  005737  002170                 TST     FATFLG                  ;ANY FATAL ERRORS ?
6173 037204  001402                         BEQ     60$                     ;BRANCH IF NOT
6174 037206  004737  017776                 JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
6175 037212                         60$:
```

```
 6177                                   ;+
 6178                                   ;
 6179                                   ;TEST 8, SUBTEST 2
 6180                                   ;
 6181                                   ;CHECK THAT NON-ZERO MODE BITS BEING SET CAUSES
 6182                                   ;INITIALIZE COMMAND TO BE REJECTED
 6183                                   ;
 6184                                   ;-
 6185
 6186 037212                                   BGNSUB                          ;////////////// BEGIN SUBTEST //////////////
      037212                                                                                     T8.2:
      037212  104402                                                                                TRAP    C$BSUB
 6187
 6188 037214                                   SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      037214  012700  000000                                                                      MOV     #PRI00,R0
      037220  104441                                                                              TRAP    C$SPRI
 6189 037222                                   BGNSEG                          ;>>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>>
      037222  104404                                                                              TRAP    C$BSEG
 6190
 6191
 6192 037224  004737  016470                        JSR     PC,SOFINIT        ;DO SOFT INIT OF CONTROLLER
 6193 037230  103405                                BCS     3$                ;BR IF SOFT INIT = OK
 6197 037232  010001                                MOV     R0,R1             ;SAVE CONTENTS OF TSSR
 6198 037234                                        ERRDF   ERRNO,SFIERR,SFIMSG ;DEVICE FATAL ERROR DURING INIT
      037234  104455                                                                              TRAP    C$ERDF
      037236  001447                                                                              .WORD   807
      037240  003550                                                                              .WORD   SFIERR
      037242  011506                                                                              .WORD   SFIMSG
 6199 037244                                   3$:
 6200 037244  012704  037520                        MOV     #T8PK2,R4         ;WRITE CHARACTERISTICS PACKET
 6201 037250  004737  010152                        JSR     PC,WRTCHR         ;ISSUE WRITE CHARACTERISTICS
 6202 037254  103404                                BCS     4$                ;BR, IF COMMAND ISSUED OK
 6206 037256                                        ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTISC FAILED
      037256  104456                                                                              TRAP    C$ERHRD
      037260  001450                                                                              .WORD   808
      037262  004754                                                                              .WORD   WRTMSG
      037264  011506                                                                              .WORD   SFIMSG
 6207 037266                                   4$:
 6208 037266  004737  040324                             JSR     PC,T8REST    ;SET UP PACKET FOR COMMAND
 6209 037272  012704  037450                             MOV     #T8PACKET,R4 ;GET THE ADDRESS OF COMMAND PACKET
 6210 037276                                   5$:
 6211 037276  005037  002172                   10$:     CLR     INTRECV      ;CLEAR INTERRUPT RECEIVED FLAG
 6212 037302  052714  007400                             BIS     #P.MODE,(R4) ;NON-ZERO COMMAND MODE BITS
 6213 037306  010465  177776                             MOV     R4,TSDB(R5)  ;SET THE PACKET ADDRESS
 6214 037312  004737  017060                             JSR     PC,CHKTSSR   ;WAIT FOR SSR TO SET
 6215 037316  103405                                     BCS     15$          ;BR IF CARRY SET (GOOD RETURN)
 6216 037320  010001                                     MOV     R0,R1        ;SAVE CONTENTS OF TSSR
 6220 037322                                             ERRDF   ERRNO,T8SSR,PKTSSR ;DEVICE FATAL SSR FAILED TO SET
      037322  104455                                                                                    TRAP    C$ERDF
      037324  001451                                                                                    .WORD   809
      037326  040014                                                                                    .WORD   T8SSR
      037330  011520                                                                                    .WORD   PKTSSR
 6221 037332                                   15$:     CKLOOP                   ;LOOP ON ERROR, IF FLAG SET
      037332  104406                                                                                    TRAP    C$CLP1
 6222 037334                                            ESCAPE  SEG              ;BY-PASS CHECKS IF FATAL ERROR
      037334  104410                                                                                    TRAP    C$ESCAPE
      037336  000074                                                                                    .WORD   10000$-.
```

```
6223 037340  005737  002172          TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
6224 037344  001004                  BNE     22$                     ;BRANCH IF YES
6228 037346                          ERRHRD  ERRNO,T8NINT,PKTSSR
     037346  104456                                                  TRAP    C$ERHRD
     037350  001452                                                  .WORD   810
     037352  040144                                                  .WORD   T8NINT
     037354  011520                                                  .WORD   PKTSSR
6229 037356  016501  000000  22$:    MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
6230 037362  012702  100206          MOV     #SC!SSR!TSREJ,R2        ;EXPECTED CONTENTS OF TSSR
6231 037366  032701  000100          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
6232 037372  001402                  BEQ     25$                     ;BRANCH IF NOT OFF-LINE
6233 037374  052702  000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
6234 037400  020201          25$:    CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
6235 037402  001404                  BEQ     30$                     ;OKAY IF MATCH
6239 037404                          ERRHRD  ERRNO,T82REJ,PKTSSR     ;COMMAND NOT REJECTED
     037404  104456                                                  TRAP    C$ERHRD
     037406  001453                                                  .WORD   811
     037410  037622                                                  .WORD   T82REJ
     037412  011520                                                  .WORD   PKTSSR
6240 037414                  30$:
6241 037414  004737  010354  35$:    JSR     PC,CKRAM                ;CHECK RAM TO MEMORY
6242 037420  103405                  BCS     59$                     ;RAM OK GO ON
6246 037422                          ERRHRD  ERRNO,PKTRAM,RAMERR     ;THEY DON'T MATCH
     037422  104456                                                  TRAP    C$ERHRD
     037424  001454                                                  .WORD   812
     037426  004643                                                  .WORD   PKTRAM
     037430  016204                                                  .WORD   RAMERR
6247 037432                          ENDSEG                          ;<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<
     037432                                                  10000$:
     037432  104405                                                  TRAP    C$ESEG
6248
6249 037434                  59$:    ENDSUB                          ;\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\
     037434                                                  L10070:
     037434  104403                                                  TRAP    C$ESUB
```

```
6251 037436                              EXIT    TST                     ;ALL DONE THIS TEST
     037436  104432                                                                TRAP    C$EXIT
     037440  000770                                                                .WORD   L10066-.
6252
6253                                     ;+
6254                                     ;LOCAL STORAGE FOR THIS TEST
6255                                     ;-
6256
6258 037442                                      .BLKB   10-<.-TUV2A&7>
6260 037450                              T8PACKET:                       ;COMMAND PACKET FOR TEST
6261 037450  100204                              .WORD   100204          ;WRITE CHAR COMMAND, WITH IE, ACK
6262 037452  037460                              .WORD   T8DATA          ;ADDRESS OF CHARACTERISTICS BLOCK
6263 037454  000000                              .WORD   0
6264 037456  000010                              .WORD   8.              ;STARTING VALUE OF BLOCK SIZE
6265
6266 037460                              T8DATA:                         ;CHARACTERISTICS DATA BLOCK
6267 037460  037472                              .WORD   T8BFR           ;ADDRESS OF MESSAGE BUFFER
6268 037462  000000                              .WORD   0
6269 037464  000016                              .WORD   14.             ;LENGTH OF MESSAGE BUFFER
6270 037466  000000  000000                      .WORD   0,0
6271
6272 037472                              T8BFR:  .BLKW   8.              ;MESSAGE BUFFER
6273
6274
6276 037512                                      .BLKB   10-<.-TUV2A&7>
6278 037520                              T8PK2:                          ;COMMAND PACKET FOR TEST
6279 037520  100204                              .WORD   100204          ;WRITE CHAR COMMAND, WITH IE, ACK
6280 037522  037530                              .WORD   T8DTA           ;ADDRESS OF CHARACTERISTICS BLOCK
6281 037524  000000                              .WORD   0
6282 037526  000010                              .WORD   8.              ;STARTING VALUE OF BLOCK SIZE
6283
6284 037530                              T8DTA:                          ;CHARACTERISTICS DATA BLOCK
6285 037530  037542                              .WORD   T8BF2           ;ADDRESS OF MESSAGE BUFFER
6286 037532  000000                              .WORD   0
6287 037534  000016                              .WORD   14.             ;LENGTH OF MESSAGE BUFFER
6288 037536  000000  000000                      .WORD   0,0
6289
6290 037542                              T8BF2:  .BLKW   8.              ;MESSAGE BUFFER
6291
6292
6293
6294                                     ;+
6295                                     ;LOCAL TEXT MESSAGES FOR TEST
6296                                     ;-
6297
6298 037562      111     116     111     T8NBA:  .ASCIZ  'INITIALIZE Command Not Accepted'
6299 037622      111     116     111     T82REJ: .ASCIZ  'INITIALIZE Not Rejected With Non-Zero Mode Field'
6300 037703      107     105     124     T83REJ: .ASCIZ  'GET STATUS Not Accepted'
6301 037733      107     105     124     T84REJ: .ASCIZ  'GET STATUS Not Rejected With Non-Zero Mode Field'
6302 040014      103     157     156     T8SSR:  .ASCIZ  'Contents of TSSR Incorrect After INITIALIZE'
6303 040070      103     157     156     T8SR2:  .ASCIZ  'Contents of TSSR Incorrect After CET STATUS'
6304 040144      105     170     160     T8NINT: .ASCIZ  'Expected Interrupt Not Received On INITIALIZE'
6305 040222      111     156     143     T8TSBA: .ASCIZ  'Incorrect TSBA Address After INITIALIZE'
6306 040272      116     157     156     TST8ID: .ASCIZ  'Non-Tape Motion Commands'
6307                                             .EVEN
6308
```

```
                                ;+
                                ;
                                ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
                                ;INITIALIZE COMMAND
                                ;
                                ;-

040324                          T8REST:
040324                                  SAVREG                          ;SAVE THE REGISTERS
040330  012701  037450                  MOV     #T8PACKET,R1            ;START OF THE PACKET
040334  012721  100213                  MOV     #100213,(R1)+           ;INITIALIZE WITH ACK, IE
040340  005021                          CLR     (R1)+                   ;ADDRESS OF CHAR DATA BLOCK
040342  005021                          CLR     (R1)+                   ;EXTENDED ADDRESS
040344  005021                          CLR     (R1)+                   ;SIZE OF DATA BLOCK IN BYTES
040346  005021                          CLR     (R1)+                   ;ADDRESS OF MESSAGE BUFFER
040350  005021                          CLR     (R1)+
040352  005021                          CLR     (R1)+                   ;LENGTH OF MESSAGE BUFFER
040354  005021                          CLR     (R1)+
040356  005011                          CLR     (R1)
040360  005037  037472                  CLR     T8BFR                   ;CLEAR 1ST LOC IN MESSAGE BUFFER
040364  000207                          RTS     PC                      ;RETURN
                                ;+
                                ;
                                ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
                                ;GET STATUS COMMAND
                                ;
                                ;-

040366                          T8RT2:
040366                                  SAVREG                          ;SAVE THE REGISTERS
040372  012701  037450                  MOV     #T8PACKET,R1            ;START OF THE PACKET
040376  012721  100217                  MOV     #100217,(R1)+           ;GET STATUS WITH ACK, IE
040402  005021                          CLR     (R1)+                   ;ADDRESS OF CHAR DATA BLOCK
040404  005021                          CLR     (R1)+                   ;EXTENDED ADDRESS
040406  005021                          CLR     (R1)+                   ;SIZE OF DATA BLOCK IN BYTES
040410  005021                          CLR     (R1)+                   ;ADDRESS OF MESSAGE BUFFER
040412  005021                          CLR     (R1)+
040414  005021                          CLR     (R1)+                   ;LENGTH OF MESSAGE BUFFER
040416  005021                          CLR     (R1)+
040420  005011                          CLR     (R1)
040422  005037  037472                  CLR     T8BFR                   ;CLEAR 1ST LOC IN MESSAGE BUFFER
040426  000207                          RTS     PC                      ;RETURN
040430                          ENDTST
040430
040430  104401                                                  L10066:
                                                                        TRAP    C$ETST
```

```
                                        .SBTTL  TEST  9: DMA MEMORY ADDRESSING

                                ;++
                                ; TEST 1
                                ;
                                ; TEST DESCRIPTION
                                ;
                                ;       This test verifies that the controller can properly address and
                                ;       access all available CPU memory (other than that occupied by the
                                ;       diagnostic and diagnostic supervisor code) for both reading (DATI)
                                ;       and writing (DATO). Verified are the LSI-11 Bus drivers for all
                                ;       available address lines. Up to this point only 16 bits have been
                                ;       used for DMA transfers.
                                ;
                                ; TEST STEPS
                                ;
                                ;  REPEAT FROM 1 TO LOOPCNT
                                ;    BEGIN
                                ;       Do Subtest 1 - Verify GET STATUS selected locations
                                ;       Do Subtest 2 - Verify message packets selected locations
                                ;       Do Subtest 3 - Verify Characteristic data selected locations
                                ;       Do Subtest 4 - Verify NXM to selected invalid addresses
                                ;    END
                                ;
                                ;--

040432                                  BGNTST
040432                                                                          T9::
040432  005037  002170                  CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
040436  012737  005672  002146          MOV     #EPRT1,EPRTSW           ;SET UP ERROR MESSAGE SWITCH
040444  005037  003100                  CLR     KTFLG           ;HOLD OFF KT11
040450  012700  042100                  MOV     #TST9ID,R0              ;ASCII MESSAGE TO IDENTIFY TEST
040454  004737  017232                  JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
040460  012737  000002  002164          MOV     #2.,LOOPCNT             ;PERFORM 2 ITERATIONS
040466                          T9LOOP:                         ;LOOP ON TEST LABEL
```

TEST 9: SUBTEST 1: GET STATUS SELECTED LOCATIONS

```
                                        .SBTTL  TEST 9: SUBTEST 1: GET STATUS SELECTED LOCATIONS
                                ;++
                                ; TEST 9: SUBTEST 1:
                                ;
                                ; SUBTEST DESCRIPTION:
                                ;
                                ;       This subtest verifies the controller can fetch a get status
                                ;       command from all available memory locations.
                                ;       Two word blocks are tested one at a time by first setting
                                ;       all available memory to a background pattern of 125252.
                                ;       A Get Status command is then executed to various addresses in
                                ;       each available memory 4k word block. The various addresses
                                ;       are determined by floating a 1 then a 0 through the address bits.
                                ;
                                ; TEST STEPS:
                                ;
                                ;    BEGIN
                                ;       Fill Memory with background pattern of 125252
                                ;       Write to TSSR to soft initialize
                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer
                                ;
                                ;       REPEAT FOR SELECTED VALID ADDRESSES IN DIAGNOSTIC FREE SPACE AND ABOVE 32K
                                ;         BEGIN
                                ;               Get a valid modulo-4 test address
                                ;               Do a GET STATUS command from the test address
                                ;         END
                                ;    END
                                ;--

040466                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
040466                                                                          T9.1:
040466  104402                                                                  TRAP    C$BSUB

                                ;Fill Memory with background pattern of 125252
040470  012700  125252                  MOV     #125252,R0              ;BACKGROUND DATA
040474  004737  020216                  JSR     PC,FILLMEM              ;FILL MEMORY WITH BACKGROUND DATA

                                ;Write to TSSR to soft initialize
040500  004737  016470                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
040504  103405                          BCS     15$                     ;BR IF SOFT INIT = OK
040506                                  NEXT.ERRNO
040506  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
040510                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
040510  104455                                                                  TRAP    C$ERDF
040512  001605                                                                  .WORD   901
040514  003550                                                                  .WORD   SFIERR
040516  011506                                                                  .WORD   SFIMSG

                                ;Do a WRITE CHARACTERISTICS to setup a message buffer
040520                          15$:
040520  012704  041670                  MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
040524  004737  043436                  JSR     PC,T9SWRT               ;RESTORE PACKET TO STARTING VALUES
040530  005037  003102                  CLR     KTENABLE                ;TURN OFF KT-11
040534  010465  177776                  MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
040540  004737  017060                  JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
040544                                  FORCERROR       17$
040560  103405                          BCS     20$                     ;BR IF SSR SET IN CHKTSSR
```

```
6447 040562  010001                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6448 040564                                NEXT.ERRNO
6449 040564                        17$:    ERRDF   ERRNO,T9WRTSSR,PKTSSR   ;DEVICE FATAL SSR FAILED TO SET
     040564  104455                                                                TRAP    C$ERDF
     040566  001606                                                                .WORD   902
     040570  042202                                                                .WORD   T9WRTSSR
     040572  011520                                                                .WORD   PKTSSR
6450
6451                                       ;Verify a Get Status can be fetched from each address
6452                                       ;Get a valid modulo-4 test address
6453                                       ;Do a GET STATUS command from the test address
6454 040574  005037  002170           20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
6455 040600  005037  041740                   CLR     T9KT                    ;TEST ABOVE 28K SWITCH
6456 040604  012702  041744                   MOV     #T9BLK,R2               ;POINT TO TEST PATTERN TABLE
6457 040610                        T91LOOP:
6458 040610  005037  003102                   CLR     KTENABLE                ;TURN OFF ABOVE 28K .EST FLAG
6459 040614  012201                           MOV     (R2)+,R1                ;GET TEST PATTERN ADDRESS
6460 040616  005000                           CLR     R0                      ;ASSUME NO TEST ABOVE 28K
6461 040620  005737  041740                   TST     T9KT                    ;TEST ABOVE 28K THIS TIME?
6462 040624  001407                           BEQ     25$                     ;BR IF NO
6463 040626  016200  177776                   MOV     -2(R2),R0               ;GET TEST PATTERN AGAIN
6464 040632  042700  177774                   BIC     #^C<A1716>,R0           ;SAVE 18 BIT ADDRESS ONLY
6465 040636  012737  000001  003102           MOV     #1,KTENABLE             ;TURN ON ABOVE 28K TEST FLAG
6466 040644  004737  042746           25$:    JSR     PC,T9CONVERT            ;CONVERT TEST PATTERN TO TEST ADDRESS
6467 040650  103034                           BCC     65$                     ;BR IF INVALID PACKET ADDRESS
6468 040652  013704  041734                   MOV     T9LOADD,R4              ;COPY CURRENT PACKET LOW ADDRESS
6469 040656  013703  041732                   MOV     T9HIADD,R3              ;COPY CURRENT PACKET HIGH ADDRESS
6470 040662  004737  043504                   JSR     PC,T9SETGET             ;SETUP CURRENT PACKET TO GET STATUS
6471 040666  042703  177774                   BIC     #^C<A1716>,R3           ;SAVE ADDRESS BITS 17+16
6472 040672  050304                           BIS     R3,R4                   ;SETUP 18 BIT PACKET ADDRESS
6473 040674  004737  020070                   JSR     PC,KTOFF                ;TURN OFF KT-11
6474 040700  010465  177776                   MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
6475 040704  004737  017060                   JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
6476 040710                                   FORCERROR       32$
6477 040724  103405                           BCS     40$                     ;BR IF SSR SET IN CHKTSSR
6478 040726  010001                           MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
6479 040730                                   NEXT.ERRNO
6480 040730                        32$:    ERRDF   ERRNO,T9GETSSR,PKTGETS  ;DEVICE FATAL SSR FAILED TO SET
     040730  104455                                                                TRAP    C$ERDF
     040732  001607                                                                .WORD   903
     040734  042126                                                                .WORD   T9GETSSR
     040736  011550                                                                .WORD   PKTGETS
6481 040740                        40$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     040740  104406                                                                TRAP    C$CLP1
6482 040742                        65$:
6483 040742                                   FORCEXIT        80$
6484 040752  020227  042076                   CMP     R2,#T9TBE               ;DONE ALL TSTBLK TEST PATTERNS?
6485 040756  103002                           BHIS    70$                     ;BR IF YES
6486 040760  000137  040610                   JMP     T91LOOP         ;DO ANOTHER MODULO- 4 ADDRESS
6487 040764  005737  041740           70$:    TST     T9KT                    ;DONE ABOVE 28K TESTING TOO?
6488 040770  003012                           BGT     80$                     ;BR IF YES
6489 040772  005737  003100                   TST     KTFLG                   ;ANY MEMORY ABOVE 28K ON SYSTEM?
6490 040776  001407                           BEQ     80$                     ;BR IF NO
6491 041000  012737  000001  041740           MOV     #1,T9KT         ;SET SWITCH
6492 041006  012702  041744                   MOV     #T9BLK,R2               ;RESET TEST PATTERN TABLE
6493 041012  000137  040610                   JMP     T91LOOP         ;DO ABOVE 28K TESTING
6494 041016  004737  020070           80$:    JSR     PC,KTOFF                ;TURN OFF KT11
```

```
6495 041022                                ENDSUB                    ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     041022                                                                L10072:
     041022  104403                                                                TRAP    C$ESUB
6496 041024  005737  002170                TST     FATFLG            ;ANY FATAL ERRORS ?
6497 041030  001402                        BEQ     100$              ;BRANCH IF NOT
6498 041032  004737  017776                JSR     PC,CKDROP         ;TRY TO DROP THE UNIT
6499 041036                          100$:
6500
```

```
                                        .SBTTL  TEST 9: SUBTEST 2: MESSAGE PACKETS TO SELECTED LOCATIONS
                                ;++
                                ; TEST 9: SUBTEST 2:
                                ;
                                ; SUBTEST DESCRIPTION:
                                ;
                                ;       This subtest verifies the controller can deposit message packets
                                ;       to all available memory locations.
                                ;       First all available memory is set to a background pattern
                                ;       of 125252.
                                ;       Write Characteristics commands are then executed with message
                                ;       buffer addresses set to various addresses in each available
                                ;       memory location.
                                ;       The various addresses are determined by floating a 1 then a 0
                                ;       through the address bits.
                                ;
                                ; TEST STEPS:
                                ;
                                ;    BEGIN
                                ;       Fill Memory with background pattern of 125252
                                ;       Write to TSSR to soft initialize
                                ;       Do a WRITE CHARACTERISTICS to setup a message buffer to compare
                                ;
                                ;       REPEAT FOR SELECTED ADDRESSES IN DIAGNOSTIC FREE SPACE AND ABOVE 32K
                                ;         BEGIN
                                ;               Get a valid modulo-4 test address
                                ;               Set the packet message buffer to the TEST ADDRESS
                                ;               Do a WRITE CHARACTERISTICS
                                ;               Restore the test message buffer to background pattern
                                ;         END
                                ;    END
                                ;--

041036                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
041036                                                                          T9.2:
041036  104402                                                                  TRAP    C$BSUB

                                ;Fill Memory with background pattern of 125252
041040  012700  125252                  MOV     #125252,R0              ;BACKGROUND DATA
041044  004737  020216                  JSR     PC,FILLMEM              ;FILL MEMORY WITH BACKGROUND DATA

                                ;Write to TSSR to soft initial'ze
041050  004737  016470                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
041054  103405                          BCS     15$                     ;BR IF SOFT INIT = OK
041056                                  NEXT.ERRNO
041056  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
041060                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
041060  104455                                                                  TRAP    C$ERDF
041062  001610                                                                  .WORD   904
041064  003550                                                                  .WORD   SFIERR
041066  011506                                                                  .WORD   SFIMSG

                                ;Do a WRITE CHARACTERISTICS to setup a message buffer to compare
041070                          15$:
041070  012704  041670                  MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
041074  004737  043436                  JSR     PC,T9SWRT               ;SET PACKET TO WRITE CHARACTERISTICS
041100  004737  020070                  JSR     PC,KTOFF                ;TURN OFF KT 11
```

```
 6553 041104  010465  177776                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 6554 041110  004737  017060                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 6555 041114                                 FORCERROR       17$
 6556 041130  103405                         BCS     20$                     ;BR IF SSR SET IN CHKTSSR
 6557 041132  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6558 041134                                 NEXT.ERRNO
 6559 041134                         17$:    ERRDF   ERRNO,T9WRTSSR,PKTSSR   ;DEVICE FATAL SSR FAILED TO SET
      041134  104455                                                                 TRAP    C$ERDF
      041136  001611                                                                .WORD   905
      041140  042202                                                                .WORD   T9WRTSSR
      041142  011520                                                                .WORD   PKTSSR
 6560
 6561                                ;Get a valid modulo-4 test address
 6562                                ;Set the packet message buffer to the test address
 6563                                ;Do a WRITE CHARACTERISTICS
 6564 041144  005037  002170         20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 6565 041150  012703  041744                 MOV     #T98LK,R3               ;POINT TO TEST PATTERN TABLE
 6566 041154                         T92LOOP:
 6567 041154  012301                         MOV     (R3)+,R1                ;GET TEST PATTERN ADDRESS
 6568 041156  010100                         MOV     R1,R0                   ;GET ADDRESS ALL "18 BITS"
 6569 041160  042700  177774                 BIC     #177774,R0              ;LEAVE ONLY A17 AND A16
 6570 041164  042701  000001                 BIC     #1,R1                   ;ALWAYS ON A WORD BOUNDRY
 6571 041170  004737  043140                 JSR     PC,T9CT2                ;CONVERT TEST PATTERN TO TEST ADDRESS
 6572 041174  103402                         BCS     25$                     ;BR IF VALID MESSAGE BUFFER ADDRESS
 6573 041176  000137  041274                 JMP     150$                    ;GET ANOTHER TEST PATTERN TO TRY
 6574 041202  012704  041670         25$:    MOV     #T9PACKET,R4            ;SET THE COMMAND PACKET ADDRESS
 6575 041206  004737  043436                 JSR     PC,T9SWRT               ;SETUP T9PACKET TO WRITE CHAR.
 6576 041212  013737  041734  041700         MOV     T9LOADD,T9DATA  ;SETUP LOW ORDER MESSAGE BUFFER ADD.
 6577 041220  013737  041732  041702         MOV     T9HIADD,T9DATA+2        ;SETUP HIGH ORDER MESSAGE BUFFER ADD.
 6578 041226  004737  020070                 JSR     PC,KTOFF                ;TURN OFF KT-11
 6579 041232  010465  177776                 MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 6580 041236  004737  017060                 JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 6581 041242                                 FORCERROR       32$
 6582 041256  103405                         BCS     50$                     ;BR IF SSR SET IN CHKTSSR
 6583 041260  010001                         MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6584 041262                                 NEXT.ERRNO
 6585 041262                         32$:    ERRDF   ERRNO,T9WRTSSR,PKTSSR   ;DEVICE FATAL SSR FAILED TO SET
      041262  104455                                                                 TRAP    C$ERDF
      041264  001612                                                                .WORD   906
      041266  042202                                                                .WORD   T9WRTSSR
      041270  011520                                                                .WORD   PKTSSR
 6586 041272                         50$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      041272  104406                                                                TRAP    C$CLP1
 6587 041274                         150$:
 6588 041274                                 FORCEXIT        160$
 6589 041304  020327  042076                 CMP     R3,#T9TBE               ;DONE ALL TST9BLK TEST PATTERNS?
 6590 041310  103002                         BHIS    160$                    ;BR IF YES
 6591 041312  000137  041154                 JMP     T92LOOP         ;DO ANOTHER MODULO- 4 ADDRESS
 6592 041316  004737  020070         160$:   JSR     PC,KTOFF                ;TURN OFF KT11
 6593 041322                                 ENDSUB                          ;\\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\\
      041322                                                                    L10073:
      041322  104403                                                                TRAP    C$ESUB
 6594 041324  005737  002170                 TST     FATFLG                  ;ANY FATAL ERRORS ?
 6595 041330  001402                         BEQ     180$                    ;BRANCH IF NOT
 6596 041332  004737  017776                 JSR     PC,CKDROP               ;TRY TO DROP THE UNIT
 6597 041336                         180$:
```

```
                                        .SBTTL  TEST 9: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS
                                ;**
                                ; TEST 9: SUBTEST 3:
                                ;
                                ; SUBTEST DESCRIPTION:
                                ;
                                ;       This subtest verifies the controller can fetch a
                                ;       Write Characteristics data block from all available
                                ;       memory locations.
                                ;       First all available memory is set to a background
                                ;       pattern of 125252.
                                ;       Then Write Characteristics commands are executed with
                                ;       characteristic data blocks at various memory addresses.
                                ;       The various memory addresses are determined by floating
                                ;       a 1 then a 0 through the address bits.
                                ;
                                ; TEST STEPS:
                                ;
                                ;    BEGIN
                                ;       Fill Memory with background pattern of 125252
                                ;       Write to TSSR to soft initialize
                                ;
                                ;       REPEAT FOR SELECTED VALID ADDRESSES IN DIAGNOSTIC FREE SPACE AND ABOVE 32K
                                ;         BEGIN
                                ;               Get a valid test address
                                ;               Set the test packet characteristics data pointer to the
                                ;                 test address.
                                ;               Store expected characteristic data in test address block
                                ;               Do a WRITE CHARACTERISTIC command
                                ;         END
                                ;    END
                                ;--

041336                                  BGNSUB                          ;////////////// BEGIN SUBTEST //////////
041336                                                                                  T9.3:
041336  104402                                                                          TRAP    C$BSUB

                                ;Fill Memory with background pattern of 125252
041340  012700  125252                  MOV     #125252,R0              ;BACKGROUND DATA
041344  004737  020216                  JSR     PC,FILLMEM              ;FILL MEMORY WITH BACKGROUND DATA

                                ;Write to TSSR to soft initialize
041350  004737  016470                  JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
041354  103405                          BCS     20$                     ;BR IF SOFT INIT = OK
041356                                  NEXT.ERRNO
041356  010001                          MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
041360                                  ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
041360  104455                                                                          TRAP    C$ERDF
041362  001613                                                                          .WORD   907
041364  003550                                                                          .WORD   SFIERR
041366  011506                                                                          .WORD   SFIMSG

                                ;Get a valid test address
041370  005037  002170          20$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
041374  005037  041740                  CLR     T9KT                    ;TEST ABOVE 28K SWITCH
041400  012703  041744                  MOV     #T9BLK,R3               ;POINT TO TEST PATTERN TABLE
041404                          T93LOOP:
```

```
 6650 041404  005037  003102                    CLR     KTENABLE                ;TURN OFF ABOVE 28K TEST FLAG
 6651 041410  012301                            MOV     (R3)+,R1                ;GET TEST PATTERN ADDRESS
 6652 041412  010100                            MOV     R1,R0                   ;GET ADDRESS ALL "18 BITS"
 6653 041414  042700  177774                    BIC     #177774,R0              ;LEAVE ONLY A17 AND A16
 6654 041420  042701  000003                    BIC     #3,R1                   ;GET RID OF A17 AND A16
 6655 041424  005737  041740                    TST     T9KT                    ;TEST ABOVE 28K THIS TIME?
 6656 041430  001407                            BEQ     25$                     ;BR IF NO
 6657 041432  016300  177776                    MOV     -2(R3),R0               ;GET TEST PATTERN AGAIN
 6658 041436  042700  177774                    BIC     #^C<A1716>,R0           ;SAVE 18 BIT ADDRESS ONLY
 6659 041442  012737  000001  003102            MOV     #1,KTENABLE             ;TURN ON ABOVE 28K TEST FLAG
 6660 041450  004737  042746          25$:      JSR     PC,T9CONVERT            ;CONVERT TEST PATTERN TO TEST ADDRESS
 6661 041454  103402                            BCS     30$                     ;BR IF VALID TEST ADDRESS
 6662 041456  000137  041560                    JMP     60$                     ;GET NEXT TEST PATTERN
 6663                                ;Set the test packet characteristics data pointer to the test address
 6664 041462  012704  041670          30$:      MOV     #T9PACKET,R4            ;GET THE ADDRESS OF COMMAND PACKET
 6665 041466  004737  043436                    JSR     PC,T9SWRT               ;RESTORE PACKET TO STARTING VALUES
 6666 041472  013764  041734  000002            MOV     T9LOADD,PKLOW(R4)       ;STORE CHAR. DATA PTR LOW ADDRESS
 6667 041500  013764  041732  000004            MOV     T9HIADD,PKHI(R4)        ;STORE CHAR. DATA PTR HIGH ADDRESS
 6668 041506  004737  043546                    JSR     PC,T9CHAR               ;STORE EXPECTED DATA IN DATA BLOCK
 6669                                ;Do a WRITE CHARACTERISTIC command
 6670 041512  004737  020070                    JSR     PC,KTOFF                ;TURN OFF KT-11
 6671 041516  010465  177776                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
 6672 041522  004737  017060                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 6673 041526                                    FORCERROR       32$
 6674 041542  103405                            BCS     40$                     ;BR IF SSR SET IN CHKTSSR
 6675 041544  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 6676 041546                                    NEXT.ERRNO
 6677 041546                          32$:      ERRDF   ERRNO,T9WRTSSR,PKTSSR   ;DEVICE FATAL SSR FAILED TO SET
      041546  104455                                                                    TRAP    C$ERDF
      041550  001614                                                                   .WORD   908
      041552  042202                                                                   .WORD   T9WRTSSR
      041554  011520                                                                   .WORD   PKTSSR
 6678 041556                          40$:      CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
      041556  104406                                                                    TRAP    C$CLP1
 6679 041560                          60$:
 6680 041560  020327  042076                    CMP     R3,#T9TBE                       ;DONE ALL TSTBLK TEST PATTERNS?
 6681 041564  103002                            BHIS    65$                             ;BR IF YES
 6682 041566  000137  041404                    JMP     T93LOOP         ;DO ANOTHER MODULO- 4 ADDRESS
 6683 041572  005737  041740          65$:      TST     T9KT                            ;DONE ABOVE 28K TESTING TOO?
 6684 041576  003012                            BGT     70$                             ;BR IF YES
 6685 041600  005737  003100                    TST     KTFLG                           ;ANY MEMORY ABOVE 28K ON SYSTEM?
 6686 041604  001407                            BEQ     70$                             ;BR IF NO
 6687 041606  012737  000001  041740            MOV     #1,T9KT         ;SET SWITCH
 6688 041614  012703  041744                    MOV     #T9BLK,R3                       ;RESET TEST PATTERN TABLE
 6689 041620  000137  041404                    JMP     T93LOOP         ;DO ABOVE 28K TESTING
 6690 041624  004737  020070          70$:      JSR     PC,KTOFF                        ;TURN OFF KT11
 6691 041630                                    ENDSUB                                  ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\
      041630                                                                        L10074:
      041630  104403                                                                    TRAP    C$ESUB
 6692 041632  005737  002170                    TST     FATFLG                          ;ANY FATAL ERRORS ?
 6693 041636  001402                            BEQ     75$                             ;BRANCH IF NOT
 6694 041640  004737  017776                    JSR     PC,CKDROP                       ;TRY TO DROP THE UNIT
 6695 041644                          75$:
 6696 041644  004737  017200          100$:     JSR     PC,TSTLOOP                      ;SHOULD WE DO ITERATIONS?
 6697 041650  103002                            BCC     105$                            ;BR IF NO
 6698 041652  000137  040466                    JMP     T9LOOP                          ;LOOP UNTIL ITERATION COUNT DONE
 6699 041656                          105$:
```

```
6700 041656  004737  020070          JSR     PC,KTOFF                ;TURN OFF MEMORY MANAGEMENT
6701 041662                          EXIT    TST                     ;ALL DONE THIS TEST
     041662  104432                                                          TRAP    C$EXIT
     041664  001724                                                          .WORD   L10071-.
6702
6703
6704                                 ;+
6705                                 ;LOCAL STORAGE FOR THIS TEST
6706                                 ;-
6707
6709 041666                          .BLKB   10-<.-TUV2A&7>
6711 041670                          T9PACKET:
6712 041670  100004                  .WORD   100004                  ;COMMAND PACKET FOR TEST
6713 041672  041700                  .WORD   T9DATA                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
6714 041674  000000                  .WORD   0                       ;ADDRESS OF CHARACTERISTICS BLOCK
6715 041676  000010                  .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
6716
6717 041700                          T9DATA:                 ;CHARACTERISTICS DATA BLOCK
6718 041700  041712                  .WORD   T9BFR                   ;LOW ADDRESS OF MESSAGE BUFFER
6719 041702  000000                  .WORD   0                       ;HIGH ORDER OF MESSAGE BUFFER
6720 041704  000016                  .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
6721 041706  000000  000000          .WORD   0,0
6722
6723 041712                          T9BFR:  .BLKW   8.              ;MESSAGE BUFFER
6724
6725 041732  000000                  T9HIADD:        .WORD   0       ;HIGH ADDRESS
6726 041734  000000                  T9LOADD:        .WORD   0       ;LOW ADDRESS
6727 041736  000000                  T9PAR6: .WORD   0       ;ADDR  S IN PAR FORMAT
6728 041740  000000                  T9KT:           .WORD   0       ;TEST ABOVE 28K SWITCH
6729 041742  000000                  T94TST: .WORD   0       ;ADDRE S TEST BIT
6730                                 ;+
6731                                 ;
6732                                 ;TABLE OF ADDRESSES
6733                                 ;
6734                                 ;-
6735 041744  000001                  T9BLK:  .WORD   000001
6736 041746  000002                          .WORD   000002
6737 041750  000003                          .WORD   000003
6738 041752  000005                          .WORD   000005
6739 041754  000006                          .WORD   000006
6740 041756  000007                          .WORD   000007
6741 041760  000011                          .WORD   000011
6742 041762  000012                          .WORD   000012
6743 041764  000013                          .WORD   000013
6744 041766  000021                          .WORD   000021
6745 041770  000022                          .WORD   000022
6746 041772  000023                          .WORD   000023
6747 041774  000041                          .WORD   000041
6748 041776  000042                          .WORD   000042
6749 042000  000043                          .WORD   000043
6750 042002  000101                          .WORD   000101
6751 042004  000102                          .WORD   000102
6752 042006  000103                          .WORD   000103
6753 042010  000201                          .WORD   000201
6754 042012  000202                          .WORD   000202
6755 042014  000203                          .WORD   000203
6756 042016  000401                          .WORD   000401
```

```
 6757 042020  000402                          .WORD   000402
 6758 042022  000403                          .WORD   000403
 6759 042024  001001                          .WORD   001001
 6760 042026  001002                          .WORD   001002
 6761 042030  001003                          .WORD   001003
 6762 042032  002001                          .WORD   002001
 6763 042034  002002                          .WORD   002002
 6764 042036  002003                          .WORD   002003
 6765 042040  004001                          .WORD   004001
 6766 042042  004002                          .WORD   004002
 6767 042044  004003                           WORD   004003
 6768 042046  010001                          .WORD   010001
 6769 042050  010002                          .WORD   010002
 6770 042052  010003                          .WORD   010003
 6771 042054  020001                          .WORD   020001
 6772 042056  020002                          .WORD   020002
 6773 042060  020003                          .WORD   020003
 6774 042062  040001                          .WORD   040001
 6775 042064  040002                          .WORD   040002
 6776 042066  040003                          .WORD   040003
 6777 042070  100001                          .WORD   100001
 6778 042072  100002                          .WORD   100002
 6779 042074  100003                          .WORD   100003
 6780 042076  177777                T9TBE:    .WORD           177777
 6781                                ;+
 6782                                ;LOCAL TEXT MESSAGES FOR TEST
 6783                                ;-
 6784
 6785 042100      104     115     101  TST9ID: .ASCIZ  'DMA Memory Addressing'
 6786 042126      103     157     156  T9GETSSR:       .ASCIZ  'Contents of TSSR Incorrect After GET STATUS'
 6787 042202      103     157     156  T9WRTSSR:       .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
 6788 042271      115     145     163  T9MSGBUF:       .ASCIZ  'Message Buffer Contents Incorrect After WRITE CHARACTERISTICS'
 6789 042367      102     141     143  T9BKGND:        .ASCIZ  'Background Pattern Disturbed By WRITE CHARACTERISTICS'
 6790 042455      105     170     160  T9NINT:  .ASCIZ  'Expected Interrupt Not Received On WRITE CHARACTERISTICS'
 6791 042546      127     162     151  T9DPR:          .ASCIZ  'Write Characteristic data in ram does not match expected'
 6792 042637      124     123     123  T9NXM:          .ASCIZ  'TSSR NXM bit failed to set when non-existent memory address specifi
ed'
 6793                                .EVEN
 6794
```

```
                                ;+
                                ;ROUTINE TO CONVERT A TEST PATTERN TO A VALID ADDRESS IN DIAGNOSTIC FREE SPACE
                                ;       DIAGNOSTIC FREE SPACE IS BETWEEN THE END OF THE DIAGNOSTIC AND THE
                                ;       BEGINNING OF THE SUPERVISOR. THIS IS ALWAYS BELOW 24K.
                                ;       IF MEMORY ABOVE 28K SPECIFIED (VIA R1) THEN PAR 6 IS SET
                                ;       TO THE RELOCATION BASE.
                                ;
                                ; INPUTS:
                                ;
                                ;       R0      HIGH ORDER ADDRESS BITS
                                ;       R1      LOW ORDER ADDRESS BITS
                                ;
                                ; OUPTUTS:
                                ;       T9PAR6 = ADDRESS BIASED TO PAR6 IF >28K UNDER TEST
                                ;       T9HIADD = HIGH ORDER ADDRESS IN NON PAR6 FORMAT
                                ;       T9LOADD = LOW ORDER ADDRESS IN NON PAR6 FORMAT
                                ;       C BIT = 1 IF GOOD ADDRESS RETURNED
                                ;       C BIT = 0 IF TEST PATTERN DID NOT YIELD A VALID ADDRESS
                                ;-
042746                          T9CONVERT:
042746                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
042752  005037  041734                  CLR     T9LOADD ;CLEAR LOW ADDRESS
042756  005037  041732                  CLR     T9HIADD ;CLEAR HIGH ADDRESS
042762  005037  041736                  CLR     T9PAR6          ;CLEAR PAR6 BIASED ADDRESS
042766  042701  170000                  BIC     #^C<7777>,R1    ;FORCE TO LOWER 12 BITS OF ADDRESS
042772  010005                          MOV     R0,R5           ;SAVE HIGH ORDER ADDRESS BITS
042774  004737  020070                  JSR     PC,KTOFF        ;SHUTOFF MEMORY MANAGEMENT
043000  013702  003072                  MOV     FREE,R2         ;GET FIRST FREE ADDRESS
043004  062702  000020                  ADD     #16.,R2         ;IN CASE TEST PATTERN=0
043010  060102                          ADD     R1,R2           ;ADD IN TEST PATTERN
043012  042702  000003                  BIC     #3,R2           ;MAKE IT MODULO-4
043016  013703  003076          25$:    MOV     FREEHI,R3       ;GET LAST FREE ADDRESS
043022  162703  000020                  SUB     #16.,R3         ;SAVE AT LEAST 8 WORDS (IN CASE MESSAGE BUFFER)
043026  010237  041734                  MOV     R2,T9LOADD      ;SAVE POSSIBLE LOW ADDRESS
043032  010237  041736                  MOV     R2,T9PAR6       ;SAVE IT IN PAR6 BIASED TOO
043036  020203                          CMP     R2,R3           ;IS THIS ADDRESS ABOVE FREE SPACE?
043040  101007                          BHI     35$             ;BR IF YES
043042  020237  003072                  CMP     R2,FREE         ;IS IT IN FREE SPACE?
043046  103007                          BHIS    50$             ;BR IF YES- ITS GOOD
043050  005737  003102                  TST     KTENABLE        ;TESTING ABOVE 28K?
043054  001004                          BNE     50$             ;BR IF YES
043056  000424                          BR      90$             ;BR IF NOT IN FREE SPACE
043060  162702  000020          35$:    SUB     #16.,R2         ;FORCE FIT THE TEST PATTERN
043064  000754                          BR      25$             ;TRY THIS TEST PATTERN ADDRESS
043066                          50$:
043066  005737  003102                  TST     KTENABLE        ;TESTING ABOVE 28K?
043072  001420                          BEQ     100$            ;BR IF NO
043074  005737  003100                  TST     KTFLG           ;ANY MEMORY ABOVE 28K?
043100  001413                          BEQ     90$             ;BR IF NO
043102  004737  020052                  JSR     PC,KTON         ;TURN ON MEMORY MANAGEMENT
043106  010500                          MOV     R5,R0           ;GET HIGH ORDER ADDRESS
043110  010037  041732                  MOV     R0,T9HIADD      ;SAVE POSSIBLE HIGH ADDRESS
043114  010201                          MOV     R2,R1           ;GET COMPUTED LOW ORDER ADDRESS
043116  004737  020112                  JSR     PC,SETMAP       ;RETURN PAR6 BIASED ADDRESS IN R0
043122  010037  041736                  MOV     R0,T9PAR6       ;COPY PAR6 BIASED ADDRESS
043126  103403                          BCS     105$            ;BR IF VALID ADDRESS
```

TEST 9: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS

```
 6853 043130  000241                90$:    CLC                     ;CLR C BIT FOR FAILURE
 6854 043132  000401                        BR      105$            ;
 6855 043134  000261                100$:   SEC                     ;SET SUCCESS
 6856 043136  000207                105$:   RTS     PC              ;RETURN
 6857
 6858
```

```
6860
6861                                    ;+
6862                                    ;ONLY FOR MESSAGE BUFFER ADDRESSES
6863                                    ;ROUTINE TO CONVERT A TEST PATTERN TO A VALID ADDRESS IN DIAGNOSTIC FREE SPACE
6864                                    ;       DIAGNOSTIC FREE SPACE IS BETWEEN THE END OF THE DIAGNOSTIC AND THE
6865                                    ;       BEGINNING OF THE SUPERVISOR. THIS IS ALWAYS BELOW 24K.
6866                                    ;       IF MEMORY ABOVE 28K SPECIFIED (VIA R1) THEN PAR 6 IS SET
6867                                    ;       TO THE RELOCATION BASE.
6868                                    ;
6869                                    ; INPUTS:
6870                                    ;
6871                                    ;       R0      HIGH ORDER ADDRESS BITS
6872                                    ;       R1      LOW ORDER ADDRESS BITS
6873                                    ;
6874                                    ; OUPTUTS:
6875                                    ;       T9PAR6 = ADDRESS BIASED TO PAR6 IF >28K UNDER TEST
6876                                    ;       T9HIADD = HIGH ORDER ADDRESS IN NON PAR6 FORMAT
6877                                    ;       T9LOADD = LOW ORDER ADDRESS IN NON PAR6 FORMAT
6878                                    ;       C BIT = 1 IF GOOD ADDRESS RETURNED
6879                                    ;       C BIT = 0 IF TEST PATTERN DID NOT YIELD A VALID ADDRESS
6880                                    ;-
6881 043140                            T9CT2:
6882 043140                                    SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
6883 043144  005037  041734                    CLR     T9LOADD ;CLEAR LOW ADDRESS
6884 043150  005037  041732                    CLR     T9HIADD ;CLEAR HIGH ADDRESS
6885 043154  005037  041736                    CLR     T9PAR6          ;CLEAR PAR6 BIASED ADDRESS
6886 043160  042701  170000                    BIC     #^C<7777>,R1    ;FORCE TO LOWER 12 BITS OF ADDRESS
6887 043164  010005                            MOV     R0,R5           ;SAVE HIGH ORDER ADDRESS BITS
6888 043166  004737  020070                    JSR     PC,KTOFF        ;SHUTOFF MEMORY MANAGEMENT
6889 043172  013702  003072                    MOV     FREE,R2         ;GET FIRST FREE ADDRESS
6890 043176  062702  000020                    ADD     #16.,R2         ;IN CASE TEST PATTERN=0
6891 043202  060102                            ADD     R1,R2           ;ADD IN TEST PATTERN
6892 043204  013703  003076            25$:    MOV     FREEHI,R3       ;GET LAST FREE ADDRESS
6893 043210  162703  000020                    SUB     #16.,R3         ;SAVE AT LEAST 8 WORDS (IN CASE MESSAGE BUFFER)
6894 043214  010237  041734                    MOV     R2,T9LOADD      ;SAVE POSSIBLE LOW ADDRESS
6895 043220  010237  041736                    MOV     R2,T9PAR6       ;SAVE IT IN PAR6 BIASED TOO
6896 043224  020203                            CMP     R2,R3           ;IS THIS ADDRESS ABOVE FREE SPACE?
6897 043226  101007                            BHI     35$             ;BR IF YES
6898 043230  020237  003072                    CMP     R2,FREE         ;IS IT IN FREE SPACE?
6899 043234  103007                            BHIS    50$             ;BR IF YES- ITS GOOD
6900 043236  005737  003102                    TST     KTENABLE        ;TESTING ABOVE 28K?
6901 043242  001004                            BNE     50$             ;BR IF YES
6902 043244  000424                            BR      90$             ;BR IF NOT IN FREE SPACE
6903 043246  162702  000020            35$:    SUB     #16.,R2         ;FORCE FIT THE TEST PATTERN
6904 043252  000754                            BR      25$             ;TRY THIS TEST PATTERN ADDRESS
6905 043254                            50$:
6906 043254  005737  003102                    TST     KTENABLE        ;TESTING ABOVE 28K?
6907 043260  001420                            BEQ     100$            ;BR IF NO
6908 043262  005737  003100                    TST     KTFLG           ;ANY MEMORY ABOVE 28K?
6909 043266  001413                            BEQ     90$             ;BR IF NO
6910 043270  004737  020052                    JSR     PC,KTON         ;TURN ON MEMORY MANAGEMENT
6911 043274  010500                            MOV     R5,R0           ;GET HIGH ORDER ADDRESS
6912 043276  010037  041732                    MOV     R0,T9HIADD      ;SAVE POSSIBLE HIGH ADDRESS
6913 043302  010201                            MOV     R2,R1           ;GET COMPUTED LOW ORDER ADDRESS
6914 043304  004737  020112                    JSR     PC,SETMAP       ;RETURN PAR6 BIASED ADDRESS IN R0
6915 043310  010037  041736                    MOV     R0,T9PAR6       ;COPY PAR6 BIASED ADDRESS
6916 043314  103403                            BCS     105$            ;BR IF VALID ADDRESS
```

TEST 9: SUBTEST 3: CHARACTERISTIC DATA SELECTED LOCATIONS

```
 6917 043316  000241                90$:    CLC                     ;CLR C BIT FOR FAILURE
 6918 043320  000401                        BR      105$            ;
 6919 043322  000261                100$:   SEC                     ;SET SUCCESS
 6920 043324  000207                105$:   RTS     PC              ;RETURN
 6921
```

```
                                        ;+
                                        ;
                                        ;ROUTINE TO READ THE FIRST 2 BYTES FROM RAM
                                        ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
                                        ;
                                        ;INPUT:
                                        ;
                                        ;       R4      ADDRESS OF THE COMMAND PACKET
                                        ;       R5      FIRST DEVICE UNIBUS ADDRESS
                                        ;
                                        ;OUTPUT:
                                        ;
                                        ;       CARRY   SET - RAM MATCHES PACKET
                                        ;               CLR - RAM DOES NOT MATCH PACKET
                                        ;
                                        ;IMPLICIT OUTPUT:
                                        ;
                                        ;       THE TABLE RAMDATA IS FILLED WITH THE
                                        ;       DATA HELD IN RAM.
                                        ;       RAMSIZ  SET TO 2 FOR PRAMPKT ROUTINE
                                        ;
                                        ;SIDE EFFECTS:
                                        ;
                                        ;       THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
                                        ;
                                        ;-
043326                                  T9CKRAM::
043326                                          SAVREG                          ;SAVE THE GENERAL REGISTERS
043332  012701  002206                          MOV     #RAMDATA,R1             ;ADDRESS TO SAVE THE RAM DATA
043336  012702  000020                          MOV     #RMPKTBEG,R2            ;BYTE ADDRESS OF FIRST RAM DATA
043342  005003                                  CLR     R3                      ;CLEAR THE ERROR FLAG
043344  004737  017060                          JSR     PC,CHKTSSR              ;WAIT FOR SSR
043350  112765  000000  177776                  MOVB    #0,TSDB(R5)             ;SET MAINTENANCE MODE
043356  004737  017060          10$:    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
043362  010265  177776                          MOV     R2,TSDB(R5)             ;SELECT NEXT RAM ADDRESS
043366  004737  017060                          JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
043372  116511  177776                          MOVB    TSBA(R5),(R1)           ;READ THE RAM DATA
043376  122124                                  CMPB    (R1)+,(R4)+             ;COMPARE TO EXPECTED
043400  001401                                  BEQ     20$                     ;BRANCH IF OK
043402  005203                                  INC     R3                      ;SET ERROR FLAG
043404  005202                  20$:    INC     R2                      ;ADDRESS OF NEXT RAM LOCATION
043406  020227  000022                          CMP     R2,#RMPKTBEG+2          ;DONE 2 BYTES?
043412  002761                                  BLT     10$                     ;BR IF NO
043414  005703                                  TST     R3                      ;WAS AN ERROR FOUND ?
043416  001402                                  BEQ     30$                     ;BRANCH IF NOT
043420  000241                                  CLC                             ;CLEAR CARRY TO SHOW ERROR
043422  000401                                  BR      50$                     ;AND EXIT
043424  000261                  30$:    SEC                             ;SHOW GOOD COMPARE
043426  012737  000002  002246  50$:    MOV     #2,RAMSIZ               ;SETUP RAMSIZ
043434  000207                                  RTS     PC                      ;RETURN
                                        ;+
                                        ;
                                        ;ROUTINE TO SETUP PACKET TO WRITE CHARACTERISTICS
                                        ;-

043436                                  T9SWRT:
043436                                          SAVREG                  ;SAVE THE REGISTERS
```

```
 6980 043442  012701  041670                  MOV     #T9PACKET,R1    ;START OF THE PACKET
 6981 043446  012721  100004                  MOV     #100004,(R1)+   ;WRITE CHARACTERISTICS WITH ACK
 6982 043452  012721  041700                  MOV     #T9DATA,(R1)+   ;ADDRESS OF CHAR DATA BLOCK
 6983 043456  005021                          CLR     (R1)+           ;EXTENDED ADDRESS
 6984 043460  012721  000010                  MOV     #8.,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
 6985 043464  012721  041712                  MOV     #T9BFR,(R1)+    ;ADDRESS OF MESSAGE BUFFER
 6986 043470  005021                          CLR     (R1)+
 6987 043472  012721  000016                  MOV     #14.,(R1)+      ;LENGTH OF MESSAGE BUFFER
 6988 013476  005021                          CLR     (R1)+
 6989 043500  005011                          CLR     (R1)
 6990 043502  000207                          RTS     PC              ;RETURN
 6991                                 ;+
 6992                                 ;
 6993                                 ;ROUTINE TO SETUP A GET STATUS COMMAND PACKET AT CURRENT PACKET ADDRESS
 6994                                 ;
 6995                                 ;       R3      HIGH ORDER PACKET ADDRESS
 6996                                 ;       R4      LOW ORDER PACKET ADDRESS
 6997                                 ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG CLEAR
 6998                                 ;
 6999                                 ;-
 7000
 7001 043504                          T9SETGET:
 7002 043504                                  SAVREG                  ;SAVE THE REGISTERS
 7003 043510  010401                          MOV     R4,R1           ;GET LOW ORDER ADDRESS
 7004 043512  005737  003102                  TST     KTENABLE        ;TESTING ABOVE 28K?
 7005 043516  001404                          BEQ     10$             ;BR IF NO
 7006 043520  010300                          MOV     R3,R0           ;GET HIGH ORDER ADDRESS
 7007 043522  004737  020112                  JSR     PC,SETMAP       ;RETURN ADDRESS BIASED TO PAR6 IN R0
 7008 043526  010001                          MOV     R0,R1           ;GET ADDRESS
 7009 043530  012700  000017          10$:    MOV     #P.GETSTATUS,R0 ;GET STATUS COMMAND CODE NO IE
 7010 043534  052700  100000                  BIS     #P.ACK,R0       ;SET ACK
 7011 043540  010021                          MOV     R0,(R1)+        ;STORE GET STATUS IN PACKET
 7012 043542  005021                          CLR     (R1)+           ;CLEAR UNUSED WORD
 7013 043544  000207                          RTS     PC              ;RETURN
 7014                                 ;+
 7015                                 ;
 7016                                 ;ROUTINE TO SETUP A CHARACTERISTIC DATA BLOCK AT A TEST ADDRESS
 7017                                 ;
 7018                                 ;-
 7019 043546                          T9CHAR:
 7020 043546                                  SAVREG                  ;SAVE R1-R5 UNTIL NEXT RETURN
 7021 043552  012700  041700                  MOV     #T9DATA,R0      ;GET T9PACKET DATA POINTER
 7022 043556  013701  041734                  MOV     T9LOADD,R1      ;ASSUME NOT ABOVE 28K
 7023 043562  005737  003102                  TST     KTENABLE        ;TESTING ABOVE 28K?
 7024 043566  001402                          BEQ     10$             ;BR IF NO
 7025 043570  013701  041736                  MOV     T9PAR6,R1       ;SET TEST ADDRESS ABOVE 28K
 7026 043574  012021                  10$:    MOV     (R0)+,(R1)+     ;STORE DATA WORD 1
 7027 043576  012021                          MOV     (R0)+,(R1)+     ;STORE DATA WORD 2
 7028 043600  012021                          MOV     (R0)+,(R1)+     ;STORE DATA WORD 3
 7029 043602  012021                          MOV     (R0)+,(R1)+     ;STORE DATA WORD 4
 7030 043604  012021                          MOV     (R0)+,(R1)+     ;STORE DATA WORD 5
 7031 043606  000207                          RTS     PC              ;RETURN
 7032 043610                                  ENDTST
      043610
      043610  104401                                                  L10071:
                                                                              TRAP    C$ETST
```

```
                                        .SBTTL  TEST  10: INITIALIZE AFTER WRITE CHARACTERISTICS
                                ;*
                                ; TEST DESCRIPTION:
                                ;
                                ;       This test verifies that a Hardware Initialize command
                                ;       invoked after a Write Characteristics command sets up
                                ;       the Command, Message and Characteristic image blocks
                                ;       in the controller ram correctly.
                                ;
                                ; TEST STEPS:
                                ;
                                ;       REPEAT FOR LOOPCNT
                                ;         BEGIN
                                ;               Do WRITE CHARACTERISTICS command.
                                ;               If the NBA bit in the TSSR register is NOT=0 then Print Error.
                                ;               Write to TSSR register to soft initialize the controller
                                ;               If controller RAM 310-377 NOT=0 then Print Error
                                ;         END
                                ;--


                                        BGNTST
                                                                                        T10::
                                        CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
                                        MOV     #EPRT1,EPRTSW           ;SET UP ERROR MESSAGE SWITCH
                                        CLR     KTFLG           ;HOLD OFF KT11
                                        MOV     #TST10ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
                                        JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
                                        MOV     #2.,LOOPCNT             ;PERFORM 2 ITERATIONS
                                T10LOOP:
                                        JSR     PC,T10REST              ;SET PACKET TO START-UP VALUES

                                        MOV     #TSTBLK+10.,R3          ;START OF TEST DATA
                                        MOV     #T10PACKET,R4           ;GET THE ADDRESS OF COMMAND PACKET
                                        MOV     #8.,PKBCNT(R4)          ;START WITH MINIMUM ALLOWABLE VALUE
                                5$:
                                        JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
                                        BCS     10$                     ;BR IF SOFT INIT OKAY
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
                                                                                        TRAP    C$ERDF
                                                                                        .WORD   1000
                                                                                        .WORD   SFIERR
                                                                                        .WORD   SFIMSG

                                ;Do WRITE CHARACTERISTICS command.
                                10$:    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
                                        MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS TO EXECUTE
                                        JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
                                        FORCERROR       12$                     ;@@FORCE ERROR IF FORCER=1
                                        BCS     15$                     ;BR IF CARRY SET (GOOD RETURN)
                                        MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
                                        NEXT.ERRNO
                                12$:    ERRDF   ERRNO,T10SSR,PKTSSR     ;DEVICE FATAL SSR FAILED TO SET
                                                                                        TRAP    C$ERDF
                                                                                        .WORD   1001
                                                                                        .WORD   T10SSR
```

```
     043752  011520                                                                  .WORD   PKTSSR
7088 043754  004737  017724          JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
7089 043760                  15$:    CKLOOP                          ;LOOP ON ERROR, IF FLAG SET
     043760  104406                                                                  TRAP    C$CLP1
7090 043762  016501  000000          MOV     TSSR(R5),R1             ;GET THE CONTENTS OF TSSR
7091 043766  012702  000200          MOV     #SSR,R2                 ;EXPECTED CONTENTS OF TSSR
7092 043772  032701  000100          BIT     #OFL,R1                 ;IS OFF-LINE BIT SET ?
7093 043776  001402                  BEQ     25$                     ;BRANCH IF NOT OFF-LINE
7094 044000  052702  000100          BIS     #OFL,R2                 ;SET OFF-LINE IN EXPECTED DATA
7095
7096                                ;If the NBA bit in the TSSR register is NOT=0 then Print Error.
7097 044004                  25$:
7098 044004                          FORCERROR       27$                     ;@@D
7099 044020  020201                  CMP     R2,R1                   ;DOES EXPECTED MATCH RECEIVED ?
7100 044022  001404                  BEQ     30$                     ;OKAY IF MATCH
7101 044024                          NEXT.ERRNO
7102 044024                  27$:    ERRHRD  ERRNO,T10NBA,PKTSSR     ;NBA NOT ZERO
     044024  104456                                                                  TRAP    C$ERHRD
     044026  001752                                                                  .WORD   1002
     044030  044374                                                                  .WORD   T10NBA
     044032  011520                                                                  .WORD   PKTSSR
7103 044034                  30$:    CKLOOP                          ;LOOP ON ERROR ?
     044034  104406                                                                  TRAP    C$CLP1
7104
7105                               ;Write to TSSR register to soft initialize the controller
7106 044036                  40$:
7107 044036  004737  016470          JSR     PC,SOFINIT              ;WRITE TO TSSR TO SOFT INITIALIZE
7108 044042                          FORCERROR       42$                     ;@@D
7109 044056  103405                  BCS     50$                     ;BR IF SOFT INIT OKAY
7110 044060  010001                  MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
7111 044062                          NEXT.ERRNO
7112 044062                  42$:    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL DURING INIT
     044062  104455                                                                  TRAP    C$ERDF
     044064  001753                                                                  .WORD   1003
     044066  003550                                                                  .WORD   SFIERR
     044070  011506                                                                  .WORD   SFIMSG
7113
7114                               ;If controller RAM 310-377 NOT=0 then Print Error
7115 044072  012704  000310  50$:   MOV     #310,R4                 ;START WITH LOC 310
7116 044076  005002                  CLR     R2                      ;MEMORY EXPECTED SHOULD BE 000000
7117 044100  004737  017060          JSR     PC,CHKTSSR              ;WAIT FOR SSR READY
7118 044104  110465  177777  60$:    MOVB    R4,TSDBH(R5)            ;SELECT RAM ADDRESS
7119 044110  116501  177776          MOVB    TSBAL(R5),R1            ;READ LOC CONTENTS
7120 044114                          FORCERROR       62$,NOTSSR              ;@@D
7121 044124  120102                  CMPB    R1,R2                   ;CHECK MEMORY FOR 000000
7122 044126  001406                  BEQ     70$                     ;BRANCH IF DATA OKAY
7123 044130                          NEXT.ERRNO
7124 044130                  62$:    ERRDF   ERRNO,T10MEM,RAMEXP     ;MEMORY NOT ZERO AFTER INIT.
     044130  104455                                                                  TRAP    C$ERDF
     044132  001754                                                                  .WORD   1004
     044134  044335                                                                  .WORD   T10MEM
     044136  016224                                                                  .WORD   RAMEXP
7125 044140  004737  017724          JSR     PC,FATCHK               ;INC AND CHECK FOR MORE THAN 25 ERRORS
7126 044144                  70$:    CKLOOP
     044144  104406                                                                  TRAP    C$CLP1
7127 044146                          ESCAPE  TST                     ;EXIT ON FATAL ERROR
     044146  104410                                                                  TRAP    C$ESCAPE
```

```
     044150  000434                                                          .WORD   L10075-.
 7128
 7129 044152  005204                 82$:    INC     R4              ;LOOK AT NEXT RAM LOC.
 7130 044154  020427  000377                 CMP     R4,#377         ;AT TOP OF RAM ADDRESS SPACE
 7131 044160  001351                         BNE     60$             ;BRANCH TILL ALL MEMORY TESTED
 7132
 7133
 7134 044162  005737  002170                 TST     FATFLG          ;ANY FATAL ERRORS ?
 7135 044166  001402                         BEQ     160$            ;BRANCH IF NOT
 7136 044170  004737  017776                 JSR     PC,CKDROP       ;TRY TO DROP THE UNIT
 7137 044174  004737  017200         160$:   JSR     PC,TSTLOOP      ;DONE ALL ITERATIONS?
 7138 044200  103002                         BCC     165$            ;BR IF YES
 7139 044202  000137  043646                 JMP     T10LOOP         ;LOOP UNTIL ITERATION COUNT DONE
 7140 044206                         165$:
 7141 044206                         EXIT    TST
      044206  104432                                                    TRAP    C$EXIT
      044210  000374                                                    .WORD   L10075-.
 7142
 7143
```

```
                                        ;+
                                        ;LOCAL STORAGE FOR THIS TEST
                                        ;-

                                                .BLKB   10-<.-TUV2A&7>
                                        T10PACKET:                              ;COMMAND PACKET FOR TEST
                                                .WORD   100004                  ;WRITE CHARACTERISTICS COMMAND, WITH ACK
                                                .WORD   T10DATA                 ;ADDRESS OF CHARACTERISTICS BLOCK
                                                .WORD   0
                                                .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE

                                        T10DATA:                                        ;CHARACTERISTICS DATA BLOCK
                                                .WORD   T10BFR                  ;ADDRESS OF MESSAGE BUFFER
                                                .WORD   0
                                                .WORD   14.                     ;LENGTH OF MESSAGE BUFFER
                                                .WORD   0,0

                                        T10BFR: .BLKW   8.                      ;MESSAGE BUFFER
                                        ;LOCAL TEXT MESSAGES FOR TEST
                                        ;-

                                        TST10ID:        .ASCIZ  'Initialization After WRITE CHARACTERISTICS'
                                        T10MEM: .ASCIZ  'Incorrect RAM Data After Init'
                                                .EVEN
                                        T10NBA: .ASCIZ  'WRITE CHARACTERISTICS Command Not Accepted'
                                        T10SSR: .ASCIZ  'Contents of TSSR Incorrect After WRITE CHARACTERISTICS'
```

```
 7175
 7176                                   ;+
 7177                                   ;
 7178                                   ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
 7179                                   ;
 7180                                   ;-
 7181
 7182                                   .EVEN
 7183
 7184 044536                            T10REST:
 7185 044536                                    SAVREG                          ;SAVE THE REGISTERS
 7186 044542  012701  044220                      MOV     #T10PACKET,R1          ;START OF THE PACKET
 7187 044546  012721  100004                      MOV     #100004,(R1)+          ;WRITE CHARACTERISTICS WITH ACK
 7188 044552  012721  044230                      MOV     #T10DATA,(R1)+         ;ADDRESS OF CHAR DATA BLOCK
 7189 044556  005021                              CLR     (R1)+                  ;EXTENDED ADDRESS
 7190 044560  012721  000010                      MOV     #8.,(R1)+              ;SIZE OF DATA BLOCK IN BYTES
 7191 044564  012721  044242                      MOV     #T10BFR,(R1)+          ;ADDRESS OF MESSAGE BUFFER
 7192 044570  005021                              CLR     (R1)+
 7193 044572  012721  000016                      MOV     #14.,(R1)+             ;LENGTH OF MESSAGE BUFFER
 7194 044576  005021                              CLR     (R1)+
 7195 044600  005011                              CLR     (R1)
 7196 044602  000207                              RTS     PC                     ;RETURN
 7197 044604                                    ENDTST
      044604                                                                    L10075:
      044604  104401                                                                TRAP    C$ETST
```

```
 7199
 7200                                           .SBTTL  TEST 11: BASIC WRITE SUBSYSTEM MEMORY COMMAND
 7201                                   ;*
 7202                                   ;
 7203                                   ;THIS TEST VERIFIES THAT THE WRITE SUBSYSTEM MEMORY COMMAND WITH A
 7204                                   ;BSELO SELECT CODE OF 0 (NO-OP) EXECUTES CORRECTLY. IT ALSO
 7205                                   ;VERIFIES THAT A WRITE SUBSYSTEM MEMORY COMMAND WITH A NON-ZERO
 7206                                   ;MODE FIELD IS REJECTED. THE TEST FURTHER VERIFIES MICROPROGRAM
 7207                                   ;COMMAND DECODING AND HANDLING SEQUENCES.
 7208                                   ;
 7209                                   ;-
 7210 044606                                    BGNTST
      044606                                                                  T11::
 7211 044606  005037  002170                    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 7212 044612  012737  005672  002146            MOV     #EPRT1,EPRTSW           ;SET UP ERROR MESSAGE SWITCH
 7213 044620  005037  003100                    CLR     KTFLG           ;HOLD OFF KT11
 7218 044624  012700  045670                    MOV     #TST11ID,R0             ;ASCII MESSAGE TO IDENTIFY TEST
 7219 044630  004737  017232                    JSR     PC,TSTSETUP             ;DO INITIAL TEST SETUP
 7220 044634  012737  000002  002164            MOV     #2.,LOOPCNT             ;PERFORM 2 ITERATIONS
 7221 044642                            T11LOOP:
 7222 044642                                    BGNSUB                          ;////////////// BEGIN SUBTEST ////////////
      044642                                                                       T11.1:
      044642  104402                                                                   TRAP    C$BSUB
 7223 044644  004737  045736                    JSR     PC,T11REST              ;SET PACKET TO INITIAL VALUES
 7224 044650  004737  045774                    JSR     PC,T11RST               ;SET PACKET TO INITIAL VALUES
 7225 044654                                    SETPRI  #PRI00                  ;LOWER PRIORITY TO ALLOW INTERRUPTS
      044654  012700  000000                                                           MOV     #PRI00,R0
      044660  104441                                                                   TRAP    C$SPRI
 7226 044662                            5$:
 7227 044662                                    BGNSEG                          ;>>>>>>>>>>>> BEGIN SEGMENT >>>>>>>>>>>>
      044662  104404                                                                   TRAP    C$BSEG
 7228 044664  004737  016470                    JSR     PC,SOFINIT              ;DO SOFT INIT OF CONTROLLER
 7229 044670  103405                            BCS     10$                     ;BR IF SOFT INIT = OK
 7233 044672  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7234 044674                                    ERRDF   ERRNO,SFIERR,SFIMSG     ;DEVICE FATAL ERROR DURING INIT
      044674  104455                                                                   TRAP    C$ERDF
      044676  002115                                                                   .WORD   1101
      044700  003550                                                                   .WORD   SFIERR
      044702  011506                                                                   .WORD   SFIMSG
 7235 044704                            10$:
 7236 044704  012704  045500                    MOV     #T11PK2,R4              ;SUBROUTINE NEEDS PACKET ADDRESS
 7237 044710  004737  010152                    JSR     PC,WRTCHR               ;ISSUE WRITE CHARACTERISTICS
 7238 044714  103405                            BCS     11$                     ;BR, IF COMMAND ISSUED OK
 7242 044716  010001                            MOV     R0,R1                   ;SAVE CONTENTS OF TSSR
 7243 044720                                    ERRHRD  ERRNO,WRTMSG,SFIMSG     ;WRITE CHARACTERISTISC FAILED
      044720  104456                                                                   TRAP    C$ERHRD
      044722  002116                                                                   .WORD   1102
      044724  004754                                                                   .WORD   WRTMSG
      044726  011506                                                                   .WORD   SFIMSG
 7244 044730                            11$:
 7245 044730  005037  002170                    CLR     FATFLG                  ;CLEAR FATAL ERROR FLAG
 7246 044734  005037  002172                    CLR     INTRECV                 ;CLEAR INTERRUPT RECEIVED FLAG
 7247 044740  012704  045060                    MOV     #T11PACKET,R4           ;SET UP NEW WRT. SUBSYS MEM. COMMAND
 7248 044744  010465  177776                    MOV     R4,TSDB(R5)             ;SET THE PACKET ADDRESS
 7249 044750  004737  017060                    JSR     PC,CHKTSSR              ;WAIT FOR SSR TO SET
 7250 044754  005737  002172                    TST     INTRECV                 ;DID AN INTERRUPT OCCUR ?
 7251 044760  001004                            BNE     22$                     ;BRANCH IF YES
```

```
7255 044762                              ERRHRD  ERRNO,T11NINT,PKTSSR
     044762  104456                                                    TRAP    C$ERHRD
     044764  002117                                                    .WORD   1103
     044766  045576                                                    .WORD   T11NINT
     044770  011520                                                    .WORD   PKTSSR
7256 044772  016501  000000      22$:    MOV     TSSR(R5),R1         ;GET THE CONTENTS OF TSSR
7257 044776  012702  000200              MOV     #SSR,R2             ;EXPECTED CONTENTS OF TSSR
7258 045002  032701  000100              BIT     #OFL,R1             ;IS OFF-LINE BIT SET ?
7259 045006  001402                      BEQ     25$                 ;BRANCH IF NOT OFF-LINE
7260 045010  052702  000100              BIS     #OFL,R2             ;SET OFF-LINE IN EXPECTED DATA
7261 045014  020201      25$:    CMP     R2,R1               ;DOES EXPECTED MATCH RECEIVED ?
7262 045016  001404                      BEQ     30$                 ;OKAY IF MATCH
7266 045020                              ERRHRD  ERRNO,T11NBA,PKTSSR ;NBA NOT ZERO
     045020  104456                                                    TRAP    C$ERHRD
     045022  002120                                                    .WORD   1104
     045024  045522                                                    .WORD   T11NBA
     045026  011520                                                    .WORD   PKTSSR
7267 045030              30$:
7268 045030                              ENDSEG                      ;<<<<<<<<<<<<< END SEGMENT <<<<<<<<<<<<<
     045030                                                                  10000$:
     045030  104405                                                    TRAP    C$ESEG
7269
7270
7271 045032              59$:    ENDSUB                      ;\\\\\\\\\\\\\ END SUBTEST \\\\\\\\\\\\\
     045032                                                                  L10077:
     045032  104403                                                    TRAP    C$ESUB
7272
7273 045034  005737  002170              TST     FATFLG              ;ANY FATAL ERRORS ?
7274 045040  001402                      BEQ     60$                 ;BRANCH IF NOT
7275 045042  004737  017776              JSR     PC,CKDROP           ;TRY TO DROP THE UNIT
7276 045046              60$:
7277 045046                      EXIT    TST                         ;ALL DONE THIS TEST
     045046  104432                                                    TRAP    C$EXIT
     045050  000776                                                    .WORD   L10076-.
7278
7279                             ;+
7280                             ;LOCAL STORAGE FOR THIS TEST
7281                             ;-
7282
7284 045052                              .BLKB   10-<.-TUV2A&7>
7286 045060              T11PACKET:                                  ;COMMAND PACKET FOR TEST
7287 045060  100206              .WORD   100206                      ;WRITE SUBSYSTEM MEM. COMMAND, WITH IE, ACK
7288 045062  045070              .WORD   T11DATA                     ;ADDRESS OF CHARACTERISTICS BLOCK
7289 045064  000000              .WORD   0
7290 045066  000006              .WORD   6.                          ;STARTING VALUE OF BLOCK SIZE
7291 045070              T11DATA:                                    ;CHARACTERISTICS DATA BLOCK
7292 045070     000      T11BS0: .BYTE   0                           ;BSEL0 BYTE
7293 045071     000      T11BS1: .BYTE   0                           ;BSEL1 BYTE
7294 045072  000000      T11BS2: .WORD   0                           ;BSEL1 WORD
7295 045074  000000              .WORD   0                           ;DATA
7296 045076              T11BFR: .BLKW   128.                        ;MESSAGE BUFFER
7297
7298
7300 045476                              .BLKB   10-<.-TUV2A&7>
7302 045500              T11PK2:                                     ;COMMAND PACKET FOR TEST
7303 045500  100204              .WORD   100204                      ;WRITE CHARA. MEM. CMND., WITH IE, ACK
7304 045502  045510              .WORD   T11DTA                      ;ADDRESS OF DE ECT DATA BLOCK
```

```
7305 045504  000000                        .WORD   0
7306 045506  000010                        .WORD   8.                      ;STARTING VALUE OF BLOCK SIZE
7307
7308
7309 045510                        T11DTA:                                 ;SELECT DATA BLOCK
7310 045510  045076                        .WORD   T11BFR                  ;ADDRESS OF MESSAGE BUFFER
7311 045512  000000                        .WORD   0
7312 045514  000400                        .WORD   256.                    ;LENGTH OF MESSAGE BUFFER
7313 045516  000000  000000               .WORD   0,0
7314
7315
7316                               ;+
7317                               ;LOCAL TEXT MESSAGES FOR TEST
7318                               ;-
7319
7320 045522     127     122     111  T11NBA: .ASCIZ  'WRITE SUBSYSTEM MEMORY Command Not Accepted'
7321 045576     105     170     160  T11NINT:        .ASCIZ  'Expected Interrupt Not Received On WRITE SUBSYSTEM MEMORY'
7322 045670     102     141     163  TST11ID:        .ASCIZ  'Basic WRITE SUBSYSTEM MEMORY Command'
7323                               .EVEN
7324
```

```
 7326
 7327                                   ;+
 7328                                   ;
 7329                                   ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
 7330                                   ;WRITE SUBSYSTEM MEMORY COMMAND
 7331                                   ;
 7332                                   ;-
 7333
 7334 045736                           T11REST:
 7335 045736                                   SAVREG                          ;SAVE THE REGISTERS
 7336 045742  012701  045060                    MOV     #T11PACKET,R1           ;START OF THE PACKET
 7337 045746  012721  100206                    MOV     #100206,(R1)+           ;WRITE SUBSYSTEM MEM. WITH ACK, IE
 7338 045752  012721  045070                    MOV     #T11DATA,(R1)+          ;ADDRESS OF DATA BLOCK
 7339 045756  005021                            CLR     (R1)+                   ;EXTENDED ADDRESS
 7340 045760  012721  000006                    MOV     #6.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
 7341 045764  005021                            CLR     (R1)+                   ;CLEAR BSEL0 AND BSEL1
 7342 045766  005021                            CLR     (R1)+                   ;CLEAR SEL2
 7343 045770  005011                            CLR     (R1)                    ;CLEAR DATA AREA
 7344 045772  000207                            RTS     PC                      ;RETURN
 7345
 7346
 7347 045774                           T11RST:
 7348 045774                                   SAVREG                          ;SAVE THE REGISTERS
 7349 046000  012701  045500                    MOV     #T11PK2,R1              ;START OF THE PACKET
 7350 046004  012721  100204                    MOV     #100204,(R1)+           ;WRITE CHARA. WITH ACK, IE
 7351 046010  012721  045510                    MOV     #T11DTA,(R1)+           ;ADDRESS OF CHARAISTICS DATA BLOCK
 7352 046014  005021                            CLR     (R1)+                   ;EXTENDED ADDRESS
 7353 046016  012721  000010                    MOV     #8.,(R1)+               ;SIZE OF DATA BLOCK IN BYTES
 7354 046022  012721  045076                    MOV     #T11BFR,(R1)+           ;MESSAGE BUFFER ADDRESS
 7355 046026  005021                            CLR     (R1)+
 7356 046030  012721  000400                    MOV     #256.,(R1)+             ;LENGTH OF MESSAGE BUFFER
 7357 046034  005021                            CLR     (R1)+
 7358 046036  005011                            CLR     (R1)
 7359 046040  005037  045076                    CLR     T11BFR                  ;CLEAR 1ST LOC IN MESSAGE BUFFER
 7360 046044  000207                            RTS     PC                      ;RETURN
 7361 046046                                   ENDTST
      046046                                                                   L10076:
      046046
      046046  104401                                                                TRAP    C$ETST
```

```
                                        .SBTTL  HARDWARE PARAMETER CODING SECTION

                                ;++
                                ; THE HARDWARE 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.
                                ;--
046050                                  BGNHRD
046050  000010                                                          .WORD L10100-L$HARD/2
046052                                                          L$HARD::
                                
046052                                  GPRMA   HPM1,0,0,160000,177776,YES      ;GET TSBA/TSDB REGISTER ADDRESS.
046052  000031                                                          .WORD   T$CODE
046054  046072                                                          .WORD   HPM1
046056  160000                                                          .WORD   T$LOLIM
046060  177776                                                          .WORD   T$HILIM
046062                                  GPRMA   HPM2,2,0,0,776,YES              ;GET VECTOR ADDRESS.
046062  001031                                                          .WORD   T$CODE
046064  046121                                                          .WORD   HPM2
046066  000000                                                          .WORD   T$LOLIM
046070  000776                                                          .WORD   T$HILIM
                                ;       GPRMD   HPM3,4,0,340,0,7,YES            ;GET INTERRUPT PRIORITY.
046072                                  ENDHRD
                                                                        .EVEN
046072                                                          L10100:
046072  104     105     126     HPM1:   .ASCIZ  'DEVICE ADDRESS (TSSR) '
046121  111     116     124     HPM2:   .ASCIZ  'INTERRUPT VECTOR    '
046145  111     116     124     HPM3:   .ASCIZ  'INTERRUPT PRIORITY      '
                                        .EVEN
```

```
                                        .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.
                                ;--
046176                                  BGNSFT
046176  000006                                                                  .WORD L10101-L$SOFT/2
046200                                                                  L$SOFT::
046200                                  GPRML   SPM1,0,-1,YES           ;GET RAM DUMP TEST FLAG
046200  000130                                                                  .WORD   T$CODE
046202  046214                                                                  .WORD   SPM1
046204  177777                                                                  .WORD   -1
046206                                  GPRML   SPM4,2,-',YES           ; GET ITERATION CONTROL.
046206  001130                                                                  .WORD   T$CODE
046210  046260                                                                  .WORD   SPM4
046212  177777                                                                  .WORD   -1
                                ;       GPRMD   SPM6,4,D,7777,0,7777,YES        ; GET LOCAL ERROR LIMIT
                                ;       GPRMD   SPM7,6,D,7777,0,7777,YES        ; GET GLOBAL ERROR LIMIT
046214                                  ENDSFT
046214                                                                  .EVEN
                                                                        L10101:

046214  105  116  101  SPM1:   .ASCIZ  'ENABLE CONTROLLER RAM DUMP ON ERROR'
046260  111  116  110  SPM4:   .ASCIZ  'INHIBIT ITERATIONS     '
046310  120  105  122  SPM6:   .ASCIZ  'PER TEST ERROR LIMIT   '
046340  120  105  122  SPM7:   .ASCIZ  'PER UNIT ERROR LIMIT   '
                                .EVEN
                                .SBTTL  PATCH AREA
                                ;+
                                ;DISPATCH TABLE
                                ;
                                ; *** MOVE TO FRONT OF PROGRAM FOR RELEASE ***
                                ;-

046370                                  DISPATCH        TESTNO
046370  000013                                                          .WORD   11
046372                                                          L$DISPATCH::
046372  023470                                                          .WORD   T1
046374  023726                                                          .WORD   T2
046376  024700                                                          .WORD   T3
046400  026164                                                          .WORD   T4
046402  030216                                                          .WORD   T5
046404  0314 6                                                          .WORD   T6
046406  0335\2                                                          .WORD   T7
046410  036714                                                          .WORD   T8
046412  040432                                                          .WORD   T9
046414  043612                                                          .WORD   T10
046416  044606                                                          .WORD   T11

                                ;
                                ; FINALLY A GENEROUS PATCH AREA.
```

```
                                ;
                                ; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT7" HACK
                                ; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
                                ;
                                ;
046420                          PATCH::
                                ;       .IF     NZ,.&377
                                ;       .=.!377+1
                                ;       .ENDC
046420                                  LASTAD          ;SET LAST USED ADDRESS.
                                                                                .EVEN
046420  046436                                                                  .WORD T$FREE
046422  000005                                                                  .WORD T$SIZE
046424                          L$LAST::
                                        .SBTTL  HARD CODED P-TABLE
                                ;++
                                ;       DIAGNOSTIC IS PRE PARAMETERIZED PER THIS TABLE
                                ;--
046424                                  BGNSETUP        1
046424                                  BGNPTAB
046424  000000                                                                  .WORD   0
046426  000003                                                                  .WORD   L10104-./2-1
046430                                                                  L10102:
046430  172522                          .WORD           172522
046432  000224                          .WORD           224
046434  000240                          .WORD           PRI05
046436                                  ENDPTAB
046436                                                                  L10104:
046436                                  ENDSETUP

000001                                  .END
```

SYMBOL TABLE

```
ADDSSR  011612 G    C$AU  = 000052    DEBUGM  011304      FATERR= 000060      HIADDR= 001400
ADR   = 000020 G    C$AUTO= 000061    DEVCNT  002166 G    FATFLG  002170 G    HIMEM = 007776
AMBTSS  006156      C$BRK = 000022    DEVDRO  023420      FERCM   011374      HOE   = 100000 G
ASSEMB= 000010      C$BSEG= 000004    DEVNRD  023337      FIFEXP  011642 G    HPM1    046072
A1716 = 000003      C$BSUB= 000002    DEVNXR  023255      FIF1MS  011714      HPM2    046121
BADDAT  003110 G    C$CEFG= 000045    DEVONL  023173      FIF2MS  011763      HPM3    046145
BADSSR  016374 G    C$CLCK= 000062    DEVSUM  023136      FILLME  020216      IBE   = 010000 G
BAR   = 174402      C$CLEA= 000012    DFPTBL  002124 G    FNOINT  004113      IDU   = 000040 G
BENBSW  002174 G    C$CLOS= 000035    DIAGMC= 000000      FORCER  002144 G    IER   = 020000 G
BIE   = 040000      C$CLP1= 000006    DICEA = 000001      FREE    003072 G    IFAULT  004154
BIT0  = 000001 G    C$CVEC= 000036    DLCYL = 000177      FREEHI  003076      INCERK  017566
BIT00 = 000001 G    C$DCLN= 000044    DLDNER= 100200      FRESIZ  003074 G    INTCPC  016644
BIT01 = 000002 G    C$DODU= 000051    DLERR = 177730      FUSI    004015      INTFLA  016641
BIT02 = 000004 G    C$DRPT= 000024    DLGETS= 000004      F$AU  = 000015      INTMAS  016640
BIT03 = 000010 G    C$DU  = 000053    DLRDHD= 000010      F$AUTO= 000020      INTR    016712 G
BIT04 = 000020 G    C$EDIT= 000003    DLRDNH= 000016      F$BGN = 000040      INTREC  002172 G
BIT05 = 000040 G    C$ERDF= 000055    DLSR  = 000013      F$CLEA= 000007      INTVEC  016642
BIT06 = 000100 G    C$ERHR= 000056    DLUN  = 000006      F$DU  = 000016      INTX    004176
BIT07 = 000200 G    C$ERRO= 000060    DSBINT  016700      F$END = 000041      IOKCKI= 000200
BIT08 = 000400 G    C$ERSF= 000054    DUAD12  004541      F$HARD= 000004      IOKSTP= 000001
BIT09 = 001000 G    C$ERSO= 000057    DUFLG   003060 G    F$HW  = 000013      IPRI    002160 G
BIT1  = 000002 G    C$ESCA= 000010    DUMMY   003030      F$INIT= 000006      ISR   = 000100 G
BIT10 = 002000 G    C$ESEG= 000005    EF.CON= 000036 G    F$JMP = 000050      IVEC    002156 G
BIT11 = 004000 G    C$ESUB= 000003    EF.NEW= 000035 G    F$MOD = 000000      IXE   = 004000 G
BIT12 = 010000 G    C$ETST= 000001    EF.PWR= 000034 G    F$MSG = 000011      I$AU  = 000041
BIT13 = 020000 G    C$EXIT= 000032    EF.RES= 000037 G    F$PROT= 000021      I$AUTO= 000041
BIT14 = 040000 G    C$GETB= 00C026    EF.STA= 000040 G    F$PWR = 000017      I$CLN = 000041
BIT15 = 100000 G    C$GETW= 000027    EMAXDU  017521      F$RPT = 000012      I$DU  = 000041
BIT2  = 000004 G    C$GMAN= 000043    EN    = 000000      F$SEG = 000003      I$HRD = 000041
BIT3  = 000010 G    C$GPHR= 000042    ENAINT  016646      F$SOFT= 000005      I$INIT= 000041
BIT4  = 000020 G    C$GPLO= 000030    ENVIRN  021356      F$SRV = 000010      I$MOD = 000040
BIT5  = 000040 G    C$GPRI= 000040    EPRTSW  002146 G    F$SUB = 000002      I$MSG = 000041
BIT6  = 000100 G    C$INIT= 000011    EPRT1   005672      F$SW  = 000014      I$PROT= 000040
BIT7  = 000200 G    C$INLP= 000020    EPRT2   005672      F$TEST= 000001      I$PTAB= 000041
BIT8  = 000400 G    C$MANI= 000050    EPRT3   005672      GDDAT   003112 G    I$PWR = 000041
BIT9  = 001000 G    C$MEM = 000031    ERCM    011405      GERRMA  002142 G    I$RPT = 000041
BOE   = 000400 G    C$MSG = 000023    ERRHI   002202 G    GETPAT  020722 G    I$SEG = 000041
BRINIT  004355      C$OPEN= 000034    ERRK    017500      GETSEL  021004 G    I$SETU= 000041
BSEL0 = 000000      C$PNTB= 000014    ERRL0   002204 G    G$CNTO= 000200      I$SFT = 000041
BSEL1 = 000001      C$PNTF= 000017    ERRNO = 002120      G$DELM= 000372      I$SRV = 000041
CHKAMB  016540      C$PNTS= 000016    ERRVEC= 000004 G    G$DISP= 000003      I$SUB = 000041
CHKMAN  021226 G    C$PNTX= 000015    ERTABE  003330      G$EXCP= 000400      I$TST = 000041
CHKTSS  017060      C$QIO = 000377    ERTABL  003130      G$HILI= 000002      J$JMP = 000167
CKDROP  017776      C$RDBU= 000007    ESUM    017502      G$LOLI= 000001      KIPAR0= 172340
CKEMAX  017624      C$REFG= 000047    EVL   = 000004 G    G$NO  = 000000      KIPAR1= 172342
CKMSG   011032 G    C$RESE= 000033    EXBCNT= 000010      G$OFFS= 000400      KIPAR2= 172344
CKMSG2  011152 G    C$REVI= 000003    EXPBRE  016176 G    G$OFSI= 000376      KIPAR3= 172346
CKRAM   010354 G    C$RFLA= 000021    EXPD    002176 G    G$PRMA= 000001      KIPAR4= 172350
CKRAM2  010730 G    C$RPT = 000025    EXPGOT  004431      G$PRMD= 000002      KIPAR5= 172352
CMPMEM  020402      C$SEFG= 000046    EXPGT2  004465      G$PRML= 000000      KIPAR6= 172354
CONFIG  020044      C$SPRI= 000041    EXPMSG  002266 G    G$RADA= 000140      KIPAR7= 172356
COUNT   002254 G    C$SVEC= 000037    EXPREC  016170 G    G$RADB= 000000      KIPDR0= 172300
CSR   = 174400      C$TPRI= 000013    EXTA    005232      G$RADD= 000040      KIPDR1= 172302
CSRADD  002154 G    DAR   = 174404    EXTEND  005230      G$RADL= 000120      KIPDR2= 172304
CTAB    003116 G    DATA    002256 G  E$END = 002100      G$RADO= 000020      KIPDR3= 172306
CTABE   003130 G    DATAFL  014710    E$LOAD= 000035      G$XFER= 000004      KIPDR4= 172310
CTABM   003116 G    DATASC  020760    FATCHK  017724      G$YES = 000010      KIPDR5= 172312
```

SYMBOL TABLE

```
KIPDR6= 172314
KIPDR7= 172316
KTENAB  003102 G
KTFLG   003100 G
KTINIT  021444
KTOFF   020070
KTON    020052
LERRMA  002140 G
LERRNO= 000000
LISTAL= 000001
LOE   = 040000 G
LOOPCN  002164 G
LOOPCO  012600
LOOPFL  003114 G
LOT   = 000010 G
L$ACP   002110 G
L$APT   002036 G
L$AU    022366 G
L$AUT   002070 G
L$AUTO  022572 G
L$CCP   002106 G
L$CLEA  022646 G
L$CO    002032 G
L$DEPO  002011 G
L$DESC  003342 G
L$DESP  002076 G
L$DEVP  002060 G
L$DISP  046372 G
L$DLY   002116 G
L$DTP   002040 G
L$DTYP  002034 G
L$DU    022464 G
L$DUT   002072 G
L$DVTY  003334 G
L$EF    002052 G
L$ENVI  002044 G
L$ETP   002102 G
L$EXP1  002046 G
L$EXP4  002064 G
L$EXP5  002066 G
L$HARD  046052 G
L$HIME  002120 C
L$HPCP  002016 G
L$HPTP  002022 G
L$HW    002124 G
L$ICP   002104 G
L$INIT  021606 G
L$LADP  002026 G
L$LAST  046424 G
L$LOAD  002100 G
L$LUN   002074 G
L$MREV  002050 G
L$NAME  002000 G
L$PRIO  002042 G
L$PROT  021576 G
L$PRT   002112 G
L$REPP  002062 G
L$REV   002010 G
L$RPT   022674 G
L$SOFT  046200 G
L$SPC   002056 G
L$SPCP  002020 G
L$SPTP  002024 G
L$STA   002030 G
L$SW    002134 G
L$TEST  002114 G
L$TIML  002014 G
L$UNIT  002012 G
L10000  002132
L10001  002144
L10002  005226
L10003  011516
L10004  011546
L10005  011564
L10006  011572
L10007  011610
L10010  011626
L10011  011640
L10012  011712
L10013  012062
L10014  012576
L10015  013424
L10016  013446
L10017  016174
L10020  016202
L10021  016210
L10022  016222
L10023  016244
L10024  016272
L10025  016432
L10026  016742
L10030  022316
L10031  022462
L10032  022570
L10033  022644
L10034  022672
L10035  023134
L10036  023724
L10037  023572
L10040  023654
L10041  024676
L10042  024116
L10043  024310
L10044  024510
L10045  026162
L10046  025240
L10047  025560
L10050  030214
L10051  026506
L10052  026772
L10053  027220
L10054  031424
L10055  033500
L10056  031700
L10057  032124
L10060  032312
L10061  036712
L10062  034120
L10063  034512
L10064  035402
L10065  035630
L10066  040430
L10067  037176
L10070  037434
L10071  043610
L10072  041022
L10073  041322
L10074  041630
L10075  044604
L10076  046046
L10077  045032
L10100  046072
L10101  046214
L10102  046430
L10104  046436
MEMADD  013426 G
MENASC  021175
MENERR  021122
MENRES  021224
MESBFA  002716 G
MESBFN  014460
MESHEA  014643
MMVEC = 000250
MPR   = 174406
MSA.FR= 000006
MSA.NO= 000000
MSA.NR= 000004
MSA.VO= 000002
MSGEXP  011630 G
MSGLOO  012536 G
MSGSTA  012022 G
MSGSUB  013414 G
MS.ATT= 000006
MS.EXT= 000200
MS.RSD= 000001
MS.RSF= 000020
MS.RST= 000010
NBA   = 002000
NEWPAS  022020
NODEV   003062 G
NOINIT  004233
NOINTR  004117
NOITS   002136 G
NOMAN   021262
NP.IR = 000200
NP.LOO= 000040
NP.OUT= 000100
NP.WRP= 000020
NSI     004050
NSINIT  004305
NUL     004425
NULCR   004426
NXM   = 004000
NXR     003636
NXRERR  005176 G
NXRX    003675
NXTU    022032
OFL   = 000100
ONEFIL= 000000
O$APTS= 000000
O$AU  = 000001
O$BGNR= 000001
O$BGNS= 000001
O$DU  = 000001
O$ERRT= 000000
O$GNSW= 000001
O$POIN= 000001
O$SETU= 000001
PASRPT  022064
PATCH   046420 G
PATDAT  020756
PC.ERA= 002400
PC.IER= 002000
PC.NOO= 001000
PC.REL= 000000
PC.REW= 000400
PKBCNT= 000006
PKHI  = 000004
PKLOW = 000002
PKTADD  007116
PKTFRM  007060
PKTGET  011550 G
PKTMES  011574 G
PKTNEW  007153
PKTRAM  004643 G
PKTSSR  011520 G
PNT   = 001000 G
PRAMPK  013450
PRBEXP  016164
PRBMSG  016032
PRBREC  016166
PRBTOT  016117
PRBYTE  015616 G
PRI   = 002000 G
PRIADD  007532
PRIA0   007602
PRIBX0  007164 G
PRIEQU  007432
PRIPKT  006712 G
PRIRAM  007440
PRITAD  007646
PRITSS  005264
PRITO   007716
PRIXOR  007314 G
PRI00 = 000000 G
PRI01 = 000040 G
PRI02 = 000100 G
PRI03 = 000140 G
PRI04 = 000200 G
PRI05 = 000240 G
PRI06 = 000300 G
PRI07 = 000340 G
PRMESS  013702
PRMNO   002264 G
PRMSGE  015246 G
PRMSG0  015426
PRMSG1  015473
PRMSG2  015531
PROASC  014540
PR1ASC  014605
PST32W  003104 G
PUNIT   022320
PW.D11= 000021
PW.D13= 000022
PW.D22= 000020
PW.NOP= 000000
PW.NO1= 000023
PW.RDE= 000024
PW.RDR= 000001
PW.RDS= 000005
PW.RFI= 000003
PW.WCT= 000006
PW.WFI= 000004
PW.WFM= 000007
PW.WMI= 000010
PW.WNP= 000011
PW.WTR= 000002
P.ACK = 100000
P.CMD = 000037
P.CONT= 000012
P.CVC = 040000
P.FMT = 000140
P.FORM= 000011
P.GETS= 000017
P.IE  = 000200
P.INIT= 000013
P.MODE= 007400
P.OPP = 020000
P.POSI= 000010
P.READ= 000001
P.SWB = 010000
P.WRIT= 000005
P.WRTC= 000004
P.WRTS= 000006
QVP     002152 G
RAMASC  013616
RAMDAT  002206 G
RAMER   010456 G
RAMERR  016204 G
RAMEXP  016224 G
RAMFHR  014362
RAMFOR  007470
RAMHLD  010640
RAMIOP  010644
RAMPD   010715
```

SYMBOL TABLE

```
RAMR5H  010642          S0.IDB= 000010          TSSX    003716          T$$AU = 010031          T3PACK  025620
RAMSIZ  002246 G        S0.IFB= 000002          TSTBLK  002720 G        T$$AUT= 010033          T3SSR   025655
RAMTAD  016212 G        S0.IFP= 000001          TSTCNT  002162 G        T$$CLE= 010034          T3TSBA  026071
RBPCRA  014755          S0.ILD= 000020          TSTEND  017442          T$$DAT= 010104          T3.1    024734
RCVHIA  002250 G        S0.ION= 000040          TSTFLA  002260 G        T$$DU = 010032          T3.2    025254
RCVLOA  002252 G        S0.IRD= 000100          TSTL00  017200 G        T$$HAR= 010100          T4      026164 G
RDERR   005104          S0.IRW= 000004          TSTPTR  002262 G        T$$HW = 010000          T4BFR   027274
READ  = 000014          S0.ISP= 000200          TSTSET  017232 G        T$$INI= 010030          T4DATA  027260
READY = 000001          S1.ICE= 002000          TST1ID  023704          T$$MSG= 010025          T4INT   027745
RECMSG  002432 G        S1.IEO= 010000          TST10I  044262          T$$PC = 000001          T4LOOP  026220
RECV    002200 G        S1.IFM= 001000          TST11I  045670          T$$PRO= 010027          T4NBA   027406
REGSAV  020662          S1.IHE= 000400          TST2ID  024672          T$$PTA= 010103          T4PACK  027250
REWIND  010254 G        S1.IID= 004000          TST3ID  026143          T$$RPT= 010035          T4REST  030146
RMCHBE= 000167          S1.I1R= 020000          TST4ID  030117          T$$SEG= 010000          T4SP    027270
RMCHEN= 000200          S1.I2R= 040000          TST5ID  031407          T$$SOF= 010101          T4SSR   027656
RMMSGB= 000104          S1.PAR= 100000          TST6ID  033401          T$$SRV= 010026          T4TSBA  030034
RMMSGE= 000117          S2.ATI= 000010          TST7ID  036537          T$$SUB= 010077          T4.1    026220
RMPKTB= 000020          S2.BTI= 000004          TST8ID  040272          T$$SW = 010001          T4.2    026522
RMPKTE= 000027          S2.DIM= 000200          TST9ID  042100          T$$TES= 010076          T4.3    026774
RMR   = 010000          S2.ILW= 000100          TTIBFR= 177562 G        T1      023470 G        T42DAT  027314
RWPACK  010350          S2.INR= 000020          TTICSR= 177560 G        T1LOOP  023524          T42DON= 027330
SC    = 100000          S2.OUT= 000040          TTIVEC= 000060 G        T1.1    023526          T42NBA  027330
SCE   = 020000          S2.UND= 000003          TTOBFR= 177566          T1.2    023606          T42REJ  027461
SCME    004711          TBLEND= 003030 G        TTOCSR= 177564          T10     043612 G        T44REJ  027560
SDELAY  010150          TCOASC  006017          TUV2A   002000 G        T10BFR  044242          T5      030216 G
SEEK  = 000006          TCOCOD  006220          T$ARGC= 000003          T10DAT  044230          T5BFR   030742
SELASC  021170          TEMP1   003064 G        T$CODE= 001130          T10L00  043646          T5DATA  030730
SELDAT= 000004          TEMP2   003066 G        T$ERRN= 002120          T10MEM  044335          T5LOOP  030252
SEL2  = 000002          TERCLS= 000016          T$EXCP= 000000          T10NBA  044374          T5NMSG  031310
SETMAP  020112          TESTNO= 000013          T$FLAG= 000040          T10PAC  044220          T5NVCK  031131
SETU    022116          TEXASC  005756          T$FREE= 046436          T10RES  044536          T5PACK  030720
SFFMSG  011566 G        TFCASC  006060          T$GMAN= 000000          T10SSR  044447          T5SSR   031221
SFHERR  003603          TIMEXP  016246 G        T$HILI= 000776          T11     044606 G        T5VCK   030762
SFIERR  003550          TIMSG0  016274          T$LAST= 000001          T11BFR  045076          T5VCK2  031055
SFIMSG  011506 G        TINERR  011473          T$LOLI= 000000          T11BS0  045070          T6      031426 G
SFPTBL  002134 G        TKB   = 177562          T$LSYM= 010000          T11BS1  045071          T6BFR   032352
SIFLAG  003106 G        TKS   = 177560          T$LTNO= 000013          T11BS2  045072          T6DATA  032340
SIMSG   011440          TMPBFR  002576 G        T$NEST= 000000          T11DAT  045070          T6INT   033227
SKIPT   003332          TNAM    017426          T$NS0 = 000000          T11DTA  045510          T6LOOP  031462
SOFINI  016470 G        TPB   = 177566          T$NS1 = 000005          T11L00  044642          T6NBA   032406
SPACE   007760 G        TPS   = 177564          T$NS2 = 000002          T11NBA  045522          T6NINT  033136
SPM1    046214          TRANST  002134 G        T$NS3 = 000003          T11NIN  045576          T6PACK  032330
SPM4    046260          TSBA  = 177776 G        T$PCNT= 000000          T11PAC  045060          T6REST  033426
SPM6    046310          TSBAH = 177777 G        T$PTAB= 010103          T11PK2  045500          T6SSR   033047
SPM7    046340          TSBAL = 177776 G        T$PTHV= 000001          T11RES  045736          T6TSBA  033316
SR0   = 177572          TSBAM2  024530          T$PTNU= 000001          T11RST  045774          T6.1    031462
SR1   = 177574          TSBAM3  024612          T$SAVL= 177777          T11.1   044642          T6.2    031714
SR2   = 177576          TSDB  = 177776 G        T$SEGL= 177777          T2      023726 G        T6.3    032126
SR3   = 172516          TSDBH = 177777 G        T$SEKO= 010000          T2LOOP  023764          T62DAT  032372
SSR   = 000200          TSDBL = 177776 G        T$SIZE= 000005          T2.1    023744          T62DON= 032406
STATCO  012064          TSFCOD  006560          T$SUBN= 000001          T2.2    024120          T62REJ  032461
SVCGBL= 000000          TSREJ = 000006          T$TAGL= 177777          T2.3    024312          T63REJ  032560
SVCINS= 000001          TSSDEF  006127          T$TAGN= 010105          T3      024700 G        T64REJ  032653
SVCSUB= 000001          TSSR  = 000000 G        T$TEMP= 000014          T3INT   025733          T65REJ  032751
SVCTAG= 000001          TSSRBI  003400 G        T$TEST= 000013          T3LOOP  024734          T7      033502 G
SVCTST= 000001          TSSRF0  005736          T$TSTM= 177777          T3NBA   025630          T7BFR   035662
S$LSYM= 010000          TSSRH = 000001 G        T$TSTS= 000001          T3NINT  026011          T7BUFR  035724
```

SYMBOL TABLE

```
T7DATA  035650          T8SSR   040014          T9.1    040466          XFERAS  016434          X$FALS= 000040
T7DTA   035712          T8TSBA  040222          T9.2    041036          XNXM    017120          X$OFFS= 000400
T7INT   036450          T8.1    036750          T9.3    041336          XORBFO  007246          X$TRUE= 000020
T7LOOP  033536          T8.2    037212          T91LOO  040610          XORFOR  007364          X1.COR= 020000
T7MBF   035744          T82REJ  037622          T92LOO  041154          XST0  = 000006 G        X1.DLT= 100000
T7NBA   036041          T83REJ  037703          T93LOO  041404          XST1  = 000010 G        X1.MBZ= 017375
T7NINT  036357          T84REJ  037733          T94TST  041742          XST2  = 000012 G        X1.RBP= 000400
T7NNBA  036123          T9      040432 G        UAM   = 000200 G        XST3  = 000014 G        X1.SPA= 040000
T7PACK  035640          T9BFR   041712          UNITN   002150 G        XST4  = 000016 G        X1.UNC= 000002
T7PKT   035702          T9BKGN  042367          UNREC = 000006          XSOBOT= 000002          X2.BUF= 000100
T7RST   036566          T9BLK   041744          USI     004021          XSOCON  015022          X2.EXT= 000200
T7RT2   036640          T9CHAR  043546          WAITF   016744 G        XSOEOT= 000001          X2.OPM= 100000
T7SSR   036270          T9CKRA  043326 G        WC.IFA= 000200          XSOIE = 000040          X2.RCE= 040000
T7SSRM  036200          T9CONV  042746          WC.IFE= 000002          XSOILA= 000400          X2.REV= 000077
T7.1    033536          T9CT2   043140          WC.IGO= 000001          XSOILC= 001000          X2.SPA= 035400
T7.2    034122          T9DATA  041700          WC.IRE= 000010          XSOLET= 020000          X2.UNI= 000007
T7.3    034514          T9DPR   042546          WC.IRW= 000004          XSOMOT= 000200          X2.WCF= 002000
T7.4    035404          T9GETS  042126          WC.IOT= 000100          XSONEF= 002000          X3.DCK= 000010
T8      036714 G        T9HIF   041732          WC.I1T= 000040          XSOONL= 000100          X3.MBZ= 000006
T8BFR   037472          T9KT    041740          WC.I5R= 000020          XSOPED= 000010          X3.MDE= 177400
T8BF2   037542          T9LOAD  041734          WF.IED= 000010          XSORLL= 010000          X3.OPI= 000100
T8DATA  037460          T9LOOP  040466          WF.IER= 000004          XSORLS= 040000          X3.REV= 000040
T8DTA   037530          T9MSGB  042271          WF.IHI= 000200          XSOTMK= 100000          X3.RIB= 000001
T8LOOP  036750          T9NINT  042455          WF.IRE= 000040          XSOVCK= 000020          X3.SPA= 000200
T8NBA   037562          T9NXM   042637          WF.IWF= 000020          XSOWLE= 004000          X3.TRF= 000020
T8NINT  040144          T9PACK  041670          WF.IWR= 000100          XSOWLK= 000004          X4.HSP= 100000
T8PACK  037450          T9PAR6  041736          WF.I3R= 000002          XS1CON  015067          X4.MBZ= 017400
T8PK2   037520          T9SETG  043504          WF.I4R= 000001          XS2CON  015134          X4.RCE= 040000
T8REST  040324          T9SWRT  043436          WRTCHR  010152 G        XS3CON  015201          X4.TSM= 020000
T8RT2   040366          T9TBE   042076          WRTERR  005011          XXCOMM  003070 G        X4.WRC= 000377
T8SR2   040070          T9WRTS  042202          WRTMSG  004754          X$ALWA= 000000

. ABS.  046436      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  35592 WORDS  ( 140 PAGES)
DYNAMIC MEMORY:  20060 WORDS  ( 77 PAGES)
ELAPSED TIME:  00:09:31
CZTKEB.BIC,CZTKEB/-SP=SVC/ML,CZTKEB
```